

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL

**SISTEMA DE ANÁLISE QUANTITATIVA DE RISCO POR
ESCORREGAMENTOS RASOS DEFLAGRADOS POR
CHUVAS EM REGIÕES TROPICAIS**

GEORGE FERNANDES AZEVEDO

ORIENTADOR: HERNÁN E. MARTÍNEZ CARVAJAL
CO-ORIENTADOR: NEWTON MOREIRA DE SOUZA

TESE DE DOUTORADO EM GEOTECNIA

PUBLICAÇÃO: G.TD – 107/15

BRASÍLIA / DF: ABRIL / 2015

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL**

**SISTEMA DE ANÁLISE QUANTITATIVA DE RISCO POR
ESCORREGAMENTOS RASOS DEFLAGRADOS POR
CHUVAS EM REGIÕES TROPICAIS**

GEORGE FERNANDES AZEVEDO

**TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA
CIVIL E AMBIENTAL DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.**

APROVADA POR:

**HERNÁN EDUARDO MARTÍNEZ CARVAJAL, DSc (UnB)
(ORIENTADOR)**

**NEWTON MOREIRA DE SOUZA, DSc (UnB)
(CO-ORIENTADOR)**

**ANDRÉ PACHECO DE ASSIS, PhD (UnB)
(EXAMINADOR INTERNO)**

**ENNIO MARQUES PALMEIRA, PhD (UnB)
(EXAMINADOR INTERNO)**

**EDIER VICENTE ARISTIZÁBAL, DSc (ISAGEN-Colômbia)
(EXAMINADOR EXTERNO)**

**EDWIN FABIAN GARCIA ARISTIZÁBAL, PhD (Universidade de Antioquia-
Colômbia)
(EXAMINADOR EXTERNO)**

DATA: BRASÍLIA/DF, 27 de abril de 2015.

FICHA CATALOGRÁFICA

AZEVEDO, GEORGE FERNANDES

Sistema de Análise Quantitativa de Risco por Escorregamentos Rasos Deflagrados por Chuvas em Regiões Tropicais

xxxi, 399 p., 297 mm (ENC/FT/UnB, Doutor, Geotecnia, 2015)

Tese de Doutorado - Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Civil e Ambiental

1. Movimentos de Massa

2. Análise de risco

3. Sistemas de Informação Geográfica

4. Estabilidade de encostas

I. ENC/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

AZEVEDO, G.F. (2015). Sistema de Análise Quantitativa de Risco por Escorregamentos Rasos Deflagrados por Chuvas em Regiões Tropicais. Tese de Doutorado, Publicação G.TD - 107/15, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 254 p.

CESSÃO DE DIREITOS

NOME DO AUTOR: George Fernandes Azevedo

TÍTULO DA TESE DE DOUTORADO: Sistema de Análise Quantitativa de Risco por Escorregamentos Rasos Deflagrados por Chuvas em Regiões Tropicais.

GRAU / ANO: Doutor / 2015.

É concedida à Universidade de Brasília a permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese de doutorado pode ser reproduzida sem a autorização por escrito do autor.

George Fernandes Azevedo
SGAN 912 Mod. C, Bloco F, Apto. 21
Brasília – DF

DEDICATÓRIA

“A verdadeira felicidade está na própria casa, entre as alegrias da família”

Leon Tolstoi

Aos meus pais e à minha esposa, dedico.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela sua bondade e misericórdia. Por me dar alento e esperança nos momentos difíceis e de dificuldade. Por me proporcionar a bênção de realizar mais um sonho com a concretização deste trabalho.

Aos meus pais e minha esposa, pelo apoio incondicional. Sem o apoio de vocês, não poderia ter realizado este trabalho. À minha amada esposa, pela sua compreensão, amor e motivação. Pela sua ajuda e companheirismo durante todo esse caminho, sempre feliz, mesmo diante das dificuldades que passamos, e entendendo a importância deste doutorado para a nossa família. Aos meus amados pais, pela dedicação, amor e esforço, pensando mais em mim do que neles mesmos. Por me ensinarem o valor do trabalho, da honestidade e do estudo. Por me darem a vida e dividirem as suas comigo. Apesar da distância, meus pensamentos sempre estão com vocês.

Aos meus orientadores, pela amizade destes anos de mestrado e doutorado. Aos meu orientador Professor Hernán Martínez Carvajal, pela paciência e por acreditar que eu poderia concretizar esta pesquisa, sempre disposto a ajudar. Ao meu co-orientador Professor Newton Moreira de Souza, pelo seu companheirismo e fundamental auxílio no desenvolvimento do trabalho e por partilhar ideias.

Aos amigos de curso Pedro Henrique e Eduardo Montoya, pelas discussões e trabalhos realizados ao longo do doutorado.

Ao corpo de professores do Programa de Pós-graduação em Geotecnia da Universidade de Brasília.

À CNPq e à CAPES, pelo apoio financeiro ao projeto.

A todos que, de alguma maneira, contribuíram para que este trabalho fosse concluído.

SISTEMA DE ANÁLISE QUANTITATIVA DE RISCO POR ESCORREGAMENTOS RASOS DEFLAGRADOS POR CHUVAS EM REGIÕES TROPICAIS

RESUMO

Em ambientes tropicais montanhosos, a ação potencialmente desastrosa dos movimentos de massa ganha destaque, sendo que o papel das precipitações na definição de eventos associados a deslizamentos, neste contexto, fica mais evidente. O presente trabalho tem por objetivo principal apresentar uma metodologia responsável por quantificar o risco devido a escorregamentos rasos detonados por chuvas e representá-lo de forma espacial, com base na estruturação e modelagem de atributos ambientais considerados relevantes para este fim. Para tanto, desenvolveu-se um sistema computacional composto pela integração de diferentes módulos, responsáveis pela efetivação das etapas que compõem a análise do risco e capaz de se comunicar com uma plataforma pautada em sistemas de informação geográfica, determinando um tratamento mais adequado dos dados com aspectos de natureza espacial. Os passos metodológicos envolvidos na concretização do sistema computacional consistiram no estabelecimento de um arcabouço técnico acerca do risco associado a deslizamentos, com base em uma revisão bibliográfica, possibilitando a seleção dos modelos teóricos de análise que fossem mais adequados, com a sua consequente implementação computacional. Também foram desenvolvidas interfaces gráficas no sentido de facilitar a interação dos usuários com o programa. Como resultados, destacam-se exemplos de aplicações práticas empreendidas com o auxílio do sistema criado, como forma de mostrar a capacidade de cálculo dos módulos disponíveis no programa, do ponto de vista da sua funcionalidade e de diferentes configurações ambientais. Como conclusão, destaca-se que a meta maior, relacionada à concepção e à concretização de uma estrutura capaz de inserir variados elementos focados na avaliação dos efeitos trazidos por um determinado panorama de acidentes em termos de risco, foi alcançada.

QUANTITATIVE RISK ANALYSIS SYSTEM FOR SHALLOW LANDSLIDES TRIGGERED BY RAIN IN TROPICAL REGIONS

ABSTRACT

In mountainous tropical environments, potentially disastrous action of mass movements is highlighted, so that the role of precipitation in defining events associated with landslides, in this context, is apparent. The aim of this study is to present a methodology to quantifying the risk of shallow landslides triggered by rainfall and represent it spatially, based on the structuring and modeling of environmental attributes considered relevant for this purpose. Thus, a computer system, comprising the integration of different modules, was developed. These modules are responsible for conducting the analysis steps of risk and are able to communicate with a geographic information system platform, establishing a more appropriate processing of data with aspects of spatial nature. The methodological steps involved in the implementation of the computer system are related to creating a technical framework on risk of landslides, based on a literature review, enabling the selection of theoretical models of analysis that would be more appropriate, with its consequent computational implementation. Additionally, the graphical interfaces to facilitate user interaction with the program have also been developed. As a result, examples of practical applications have been developed with the aid of the system introduced, in order to display the calculating power of the available program modules, from the point of view of its functionality and different environmental settings. In conclusion, the goal related to the development and creation of a structure able to insert various elements focus on the evaluation of a given accident scenario, in terms of risk, was achieved.

SUMÁRIO

DEDICATÓRIA	iv
AGRADECIMENTOS	v
RESUMO	vi
ABSTRACT	vii
SUMÁRIO	viii
LISTA DE TABELAS	xii
LISTA DE FIGURAS	xv
LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES	xx
1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO	3
1.2 OBJETIVOS	4
1.3 ESCOPO DO TRABALHO	4
2 REVISÃO BIBLIOGRÁFICA	6
2.1 DESLIZAMENTOS DEFLAGRADOS POR EVENTOS DE CHUVAS	6
2.1.1 ASPECTOS GERAIS DE MOVIMENTOS DE MASSA DETONADOS POR CHUVAS	6
2.1.2 MODELOS FÍSICOS UTILIZADOS COM SISTEMAS DE INFORMAÇÃO GEOGRÁFICA PARA ANÁLISE DE DESLIZAMENTOS OCACIONADOS POR CHUVAS	12
2.1.3 DEFINIÇÃO DOS FUNDAMENTOS DOS MODELOS DE ESTABILIDADE DE TALUDES ADOTADOS NA PESQUISA	15
2.2 ABORDAGEM PROBABILÍSTICA USADA EM ESTABILIDADE DE TALUDES	17
2.2.1 INCERTEZAS ASSOCIADAS À ENGENHARIA GEOTÉCNICA	18
2.2.2 UTILIZAÇÃO DE BASES ESTATÍSTICAS APLICADAS À GEOTECNIA	21
2.2.3 MÉTODOS PROBABILÍSTICOS	23
2.2.3.1 MÉTODO DE MONTE CARLO	24
2.2.3.2 MÉTODO FOSM	28
2.2.3.3 MÉTODO DAS ESTIMATIVAS PONTUAIS	32
2.3 RISCO DEVIDO A MOVIMENTOS DE MASSA	38
2.3.1 CONCEITOS FUNDAMENTAIS E AVALIAÇÃO DO RISCO ASSOCIADO A DESLIZAMENTOS	38
	viii

2.3.2	ZONEAMENTO DE DESLIZAMENTOS	42
2.3.2.1	DESCRIÇÃO DAS FORMAS DE ZONEAMENTO	42
2.3.2.2	ESCALAS DE MAPAS PARA ZONEAMENTO DE DESLIZAMENTOS	43
2.3.3	DADOS ESPACIAIS UTILIZADOS NA AVALIAÇÃO DE SUSCEPTIBILIDADE, AMEAÇA E RISCO DECORRENTES DE DESLIZAMENTOS	45
2.3.3.1	INVENTÁRIOS DE DESLIZAMENTOS	46
2.3.3.2	FATORES DEFLAGRADORES	48
2.3.3.3	FATORES AMBIENTAIS	49
2.3.3.4	ELEMENTOS EM RISCO	51
2.4	BASE TEÓRICA DOS MODELOS EMPREGADOS NAS ANÁLISES	53
2.4.1	TEORIAS APLICADAS PARA O CÁLCULO DA AMEAÇA	53
2.4.1.1	CURVAS DE PROBABILIDADE DE RUPTURA EM FUNÇÃO DA PRECIPITAÇÃO ACUMULADA	54
2.4.1.2	MODELO DE BASE FÍSICA PARA PREVISÃO DE DESLIZAMENTOS SHIA_LANDSLIDE	59
2.4.1.3	MODELO SLIDE PARA SISTEMA DE ALERTA DE DESLIZAMENTOS PROVOCADOS POR CHUVAS	72
2.4.1.4	MODELAGEM MATEMÁTICA E NUMÉRICA UTILIZADAS EM ALERTAS DE DESLIZAMENTOS E FLUXO DE DETRITOS	76
2.4.2	TRABALHOS USADOS NA MODELAGEM DE CARACTERÍSTICAS DE INTENSIDADE DA AMEAÇA ASSOCIADA A DESLIZAMENTOS	82
2.4.2.1	SIMULAÇÃO DAS TRAJETÓRIAS DE DESLIZAMENTO PELO USO DE AUTÔMATOS CELULARES	82
2.4.2.2	AVALIAÇÃO DA ESPESSURA DE DETRITOS PELO USO DE ALGORITMOS DE FLUXO PARA MÚLTIPLOS VIZINHOS	87
2.4.2.3	ESTIMATIVA DA VELOCIDADE DO FLUXO DE DESLIZAMENTO PELO USO DE MODELO BASEADO NA TEORIA DO ADENSAMENTO	88
2.4.3	MODELO PARA QUANTIFICAÇÃO DA VULNERABILIDADE	94

3	METODOLOGIA	106
3.1	ANÁLISE QUANTITATIVA DE RISCO DEVIDO A DESLIZAMENTO	107
3.1.1	ESTRUTURA GERAL DO SISTEMA	107
3.1.2	MODOS DE ANÁLISE DE AMEAÇA	116
3.1.2.1	MODELOS DE CURVAS DE PROBABILIDADE DE RUPTURA EM FUNÇÃO DA CHUVA	119
3.1.2.2	MODELO SHIA_LANDSLIDE PROBABILÍSTICO	128
3.1.2.3	MODELO SLIDE PROBABILÍSTICO	136
3.1.2.4	MODELO DETERMINÍSTICO E PROBABILÍSTICO PARA ALERTA DE INICIALIZAÇÃO DE DESLIZAMENTOS E FLUXO DE DETRITOS	143
3.1.3	MODELO DE ANÁLISE DE VULNERABILIDADE	151
3.1.3.1	GERAÇÃO AUTOMÁTICA DAS TRAJETÓRIAS, VELOCIDADE E ÁREA DE DEPOSIÇÃO DE ESCORREGAMENTOS	164
3.1.4	CUSTOS DAS PROPRIEDADES E DEFINIÇÃO DO RISCO	183
3.1.4.1	INCORPORAÇÃO DOS VALORES ECONÔMICOS PARA ESTRUTURAS	185
3.1.4.2	PROCEDIMENTO PARA CÁLCULO DO RISCO	186
3.1.5	VISUALIZAÇÃO E EXPORTAÇÃO DE RESULTADOS	188
3.1.5.1	UTILIZAÇÃO DOS DADOS EXPORTADOS EM AMBIENTE SIG	188
4	RESULTADOS E ANÁLISES	195
4.1	ANÁLISE DA SUSCEPTIBILIDADE A ESCORREGAMENTOS RASOS PELO USO DE MÉTODOS PROBABILÍSTICOS	195
4.2	PROCEDIMENTO PARA AVALIAÇÃO DO RISCO DECORRENTE DE DESLIZAMENTOS APLICADO À REGIÃO DE MEDELLÍN, COLÔMBIA	203
4.3	CENÁRIOS DE AMEAÇA DE DESLIZAMENTOS PELO USO DE CURVAS CONTÍNUAS DE PROBABILIDADE DE RUPTURA EM FUNÇÃO DA CHUVA	214
4.4	DEFINIÇÃO DE AMEAÇA DE DESLIZAMENTOS PELA ASSOCIAÇÃO ENTRE O MODELO SLIDE PROBABILÍSTICO E AUTÔMATOS CELULARES	221
4.5	PROPOSTA PARA DEFINIÇÃO PRÉVIA DO PANORAMA DE SUSCEPTIBILIDADE PELO MÉTODO SHALSTAB COM ENFOQUES DETERMINÍSTICO OU PROBABILÍSTICO	230
5	CONCLUSÕES	239

5.1 MODELOS TEÓRICOS	239
5.2 SISTEMA COMPUTACIONAL	240
5.3 CASOS DE APLICAÇÃO	242
5.4 RECOMENDAÇÕES PARA PESQUISAS FUTURAS	244
REFERÊNCIAS BIBLIOGRÁFICAS	245
APÊNDICE A	263

LISTA DE TABELAS

Tabela 2.1. Resumo da classificação dos movimentos de massa (Cruden & Varnes, 1996).	7
Tabela 2.2. Atividades requeridas para a avaliação dos elementos em risco segundo Fell et al. (2008a).	52
Tabela 2.3. Níveis de alerta para escorregamentos do sistema Alerta Rio (Azevedo, 2011). .	56
Tabela 2.4. Definição do nível baixo de ocorrência de deslizamentos (Azevedo, 2011).	57
Tabela 2.5. Parâmetros de correção e intervalos de valores definidos por Vélez (2001) e Francés et al. (2007) (modificado de Aristizábal, 2014).	71
Tabela 2.6. Parâmetros e símbolos que são usados para a avaliação do modelo (modificado de Liao et al., 2010).	75
Tabela 2.7. Listagem das variáveis classificadas em dinâmicas e estáticas (modificado de Papa et al., 2011).	79
Tabela 2.8. Coeficientes da regressão logística (modificado de Lai & Dragičević, 2011).	86
Tabela 2.9. Indicação de vulnerabilidade em diferentes profundidades de fundação (modificado de Ragozin & Tikhvinsky, 2000).	99
Tabela 2.10. Valores limites exemplificados para deslocamento de fundações, relativos a diferentes estruturas (modificado de Li et al., 2010).	101
Tabela 2.11. Valores propostos do fator de resistência para a tipologia estrutural (modificado de Li et al., 2010).	104
Tabela 2.12. Valores propostos dos fatores de resistência para o estado de manutenção (modificado de Li et al., 2010).	104
Tabela 2.13. Valores propostos dos fatores de resistência para a altura (modificado de Li et al., 2010).	104
Tabela 2.14. Resumo para os métodos de cálculo de vulnerabilidade para estruturas (modificado de Li et al., 2010).	105
Tabela 3.1. Numeração proposta para o fator de resistência da tipologia estrutural.	156
Tabela 3.2. Numeração proposta para o fator de resistência do estado de manutenção.	156
Tabela 3.3. Numeração proposta para o fator de resistência referente à altura.	157
Tabela 3.4. Numeração proposta para os limites de deslocamento das fundações (modificado de Li et al., 2010).	157
Tabela 3.5. Lista de informações correspondente às alternativas de input para vulnerabilidade.	162

Tabela 3.6. Numeração e respectivas características de curvatura que devem estar presentes na grade correspondente.....	180
Tabela 3.7. Dimensões das células de cálculo (Calvello et al., 2013)	189
Tabela 3.8. Síntese das principais características dos modelos utilizados na pesquisa.	190
Tabela 4.1. Parâmetros da análise probabilística.	198
Tabela 4.2. Valores empregados na execução do método das estimativas pontuais.....	200
Tabela 4.3. Definição dos valores médios, coeficientes de variação e desvios padrões para as características do solo.....	204
Tabela 4.4. Valores dos dados de entrada para as variáveis do solo.....	205
Tabela 4.5. Acumulados de chuva mensais para o período considerado.	207
Tabela 4.6. Valores de input para as variáveis dinâmicas.....	207
Tabela 4.7. Critérios de classificação e características atribuídas às construções situadas na região analisada.	209
Tabela 4.8. Deformações impostas às estruturas presentes no interior de deslizamentos.	211
Tabela 4.9. Valores de custo conferidos às edificações (modificado de Restrepo, 2011).	212
Tabela 4.10. Coeficientes que definem as curvas de probabilidade de ruptura em função da chuva acumulada.	215
Tabela 4.11. Médias pluviométricas mensais das estações pluviométricas envolvidas para os anos de 2010 a 2012.....	216
Tabela 4.12. Valores de precipitação acumulada em 24 h para o mês de abril de cada ano utilizados na simulação.	216
Tabela 4.13. Valores de PR máximas e mínimas obtidos nas análises.	219
Tabela 4.14. Média das precipitações máximas diárias por meses.	221
Tabela 4.15. Tipos de solos presentes na área de estudo e suas características incorporadas nas análises.	223
Tabela 4.16. Disposição da porcentagem da área de estudo pelas classes de probabilidade de ruptura em relação às chuvas simuladas.	226
Tabela 4.17. Classes de estabilidade segundo modelo SHALSTAB.	232
Tabela 4.18. Tipos de solos encontrados na área de estudo, classificados conforme Roque & Souza (2008).	234
Tabela 4.19. Classificação da área de estudo entre as categorias de estabilidade previstas pelo SHALSTAB.	235

Tabela 4.20. Distribuição da região avaliada pelas classes de probabilidade para a condição muito instável.	237
--	-----

LISTA DE FIGURAS

Figura 2.1. Diagrama de corpo livre da análise de talude infinito para solo seco e sem coesão (modificado de Jordan, 2011).....	16
Figura 2.2. Método de Monte Carlo para integração (Maia, 2003).	26
Figura 2.3. Esquema para a aproximação da variável de saída probabilística conseguida pelo método das estimativas pontuais, considerando-se duas variáveis de entrada randômicas (modificado de Valley & Duff, 2011).	32
Figura 2.4. Representação dos pontos e pesos para duas variáveis correlacionadas ou não correlacionadas (Christian & Baecher (1999).	35
Figura 2.5. Combinação (2^n) dos pontos particulares $X_{i+} = \mu_i + \sigma_i$ e $X_{i-} = \mu_i - \sigma_i$ (Maia, 2003).	37
Figura 2.6. Processo de análise de risco devido a deslizamentos (modificado de Fell et al., 2005).	40
Figura 2.7. Ciclo de análise de decisão em função do risco (modificado de Einstein, 2003)..	41
Figura 2.8. Representação esquemática do processo de avaliação de risco devido a deslizamentos. (modificado de Van Westen et al., 2008).	45
Figura 2.9. Área piloto de estudo utilizada na pesquisa de Azevedo (2011).	55
Figura 2.10. Áreas de influência das estações pluviométricas para a área piloto de estudo (Azevedo, 2011).	56
Figura 2.11. Curvas de probabilidade de ruptura contra precipitação acumulada de 1 hora (Azevedo, 2011).	58
Figura 2.12. Curvas de probabilidade de ruptura contra precipitação acumulada de 24 horas (Azevedo, 2011).	59
Figura 2.13. Modelo hidrológico conceitual com seus respectivos tanques (modificado de Aristizábal, 2014).	62
Figura 2.14. Conexão entre tanques do módulo hidrológico (modificado de Aristizábal, 2014).	67
Figura 2.15. Representação da susceptibilidade à ocorrência de deslizamentos em função da inclinação da encosta e da espessura do solo (modificado de D’Odorico & Fagherazzi, 2003).	70
Figura 2.16. Módulo geotécnico do modelo SHIA_Landslide (modificado de Aristizábal, 2014).	71

Figura 2.17. Esquema representativo para o modelo SLIDE de talude infinito (modificado de Liao et al., 2010).	73
Figura 2.18. Interação entre os componentes do modelo (modificado de Lai & Dragičević, 2011).	83
Figura 2.19. Vizinhança de Moore utilizada nos autômatos celulares por Lai & Dragičević (2011).	84
Figura 2.20. Vizinhança de Moore modificada para uma exposição de vertentes com direção sudoeste (modificado de Lai & Dragičević, 2011).	84
Figura 2.21. Fluxograma da modelagem do processo com base nas características topográficas derivadas do MNT e se a curvatura é favorável ao deslizamento (modificado de Lai & Dragičević, 2011).	87
Figura 2.22. Direções múltiplas de fluxo designadas para um pixel central com uma vizinhança 3x3 (modificado de Gruber & Peckham, 2009).	87
Figura 2.23. Configurações para o excesso de poropressão estudados (modificado de Qiao & Clayton, 2013).	89
Figura 2.24. (a) Seção esquemática para o fluxo. (b) Configuração assumida para o excesso inicial de poropressão, derivado de Hutchinson (1986) (modificado de Qiao & Clayton, 2013).	90
Figura 2.25. Esquema que representa o fluxo em uma seção do talude com diferentes ângulos de declividade (modificado de Qiao & Clayton, 2013).	94
Figura 2.26. Variação teórica de V em relação ao quociente I/R (modificado de Li et al., 2010).	95
Figura 2.27. Tendência teórica de variação das curvas de vulnerabilidade em função da intensidade da ameaça para distintos valores de R (modificado de Li et al., 2010).	96
Figura 2.28. Modelo de intensidade dinâmica para estruturas (modificado de Li et al., 2010).	98
Figura 3.1. Etapas propostas para a metodologia.	106
Figura 3.2. Equação e diagrama com a abordagem do modelo de risco utilizado.	108
Figura 3.3. Organização da entrada de dados para o cálculo do risco decorrente de deslizamentos.	109
Figura 3.4. Tela do menu principal da versão inicial do sistema.	113
Figura 3.5. Interface gráfica inicial com as principais opções do sistema.	115
Figura 3.6. Ícones usados para identificar as análises.	116

Figura 3.7. Interface gráfica contendo as opções de ameaça.	118
Figura 3.8. Mecanismo geral para cálculo da ameaça.....	119
Figura 3.9. Visão da interface gráfica com opções que usam curvas de probabilidade de ruptura em função da chuva e importação direta da grade.....	121
Figura 3.10. Inputs para a opção de curvas discretas.	123
Figura 3.11. Inputs para análise utilizando curvas contínuas.....	123
Figura 3.12. Dados necessários para inclusão da precipitação nas análises.	124
Figura 3.13. Exemplo de arquivo empregado na opção que usa níveis de probabilidade de ruptura.	125
Figura 3.14. Indicação da barra de progresso para o processo de ameaça.	127
Figura 3.15. Interface gráfica do usuário referente ao modelo SHIA_Landslide.	129
Figura 3.16. Painel específico para incorporação de dados de chuva.	130
Figura 3.17. Painel retratando as opções de análise da PR.	132
Figura 3.18. Obtenção da PR a partir da função de distribuição de probabilidade.....	135
Figura 3.19. Implementação da obtenção da probabilidade de ruptura.	136
Figura 3.20. Recursos vinculados ao modelo SLIDE agrupadas na interface correspondente.	138
Figura 3.21. Esquemática da implementação do método de Monte Carlo usada para o cálculo do valor médio e desvio padrão de cada célula de trabalho.	143
Figura 3.22. Elemento gráfico voltado para o uso da simulação de alerta de inicialização de fluxos de detritos.	144
Figura 3.23. Forma de calibração do método de Papa et al. (2011).	149
Figura 3.24. Geração dos planos de probabilidade de ruptura para as opções determinísticas.	150
Figura 3.25. Interface gráfica para execução das análises de vulnerabilidade.....	152
Figura 3.26. Estrutura de input e processo de substituição de dados nas opções que utilizam grades numéricas codificadas.....	155
Figura 3.27. Exemplo para processo de iterações realizadas pelo autômato celular na definição da trajetória dos deslizamentos.	166
Figura 3.28. Disposição das células contíguas ao elemento avaliado onde são aplicados os critérios pra mudança de estado do autômato celular.	167
Figura 3.29. Esquema utilizado para transferência de material entre duas células adjacentes.	169

Figura 3.30. Esquema utilizado para o cálculo da velocidade média em cada célula de escorregamento.....	171
Figura 3.31. Delimitação do intervalo com a raiz da função espaço percorrido $x(t)$, proposta pelo algoritmo para as situações em que $x(a) > 0$ e $x'(t) < 0$ (I) e $x(a) > 0$ e $x'(t) > 0$ (II). ...	173
Figura 3.32. Forma para a definição do intervalo com a raiz da função $x(t)$ proposta pelo algoritmo para as situações em que $x(a) < 0$ e $x'(t) < 0$ (III) e $x(a) < 0$ e $x'(t) > 0$ (IV).....	174
Figura 3.33. Interface gráfica para simulação de deslizamentos segundo autômatos celulares.	177
Figura 3.34. Informações requeridas sob a forma de grades da área de estudo.	179
Figura 3.35. Definição padrão das variáveis do modelo relacionadas à malha, restrições e condições iniciais do autômato celular.....	181
Figura 3.36. Interface para visualização dos resultados para as simulações dos autômatos celulares.....	182
Figura 3.37. Interface gráfica responsável pela inclusão dos custos das estruturas e pela determinação da grade de risco.	184
Figura 3.38. Elementos gráficos voltados para a análise de risco.....	187
Figura 3.39. Interface gráfica correspondente à exportação e visualização de resultados obtidos no sistema.	188
Figura 4.1. Mapa de declividades em graus para a Vila Varjão.....	198
Figura 4.2. Carta de probabilidade de ruptura determinada pelo método FOSM.	201
Figura 4.3. Carta de probabilidade de ruptura determinada pelo método das estimativas pontuais.	202
Figura 4.4. Configuração para o evento de precipitação aplicado à análise de ameaça.....	208
Figura 4.5. Carta de ameaça com a identificação dos deslizamentos utilizados nas análises.	210
Figura 4.6. Representação das estruturas, das velocidades e áreas de acúmulo de detritos para os deslizamentos pertencentes à área piloto.	211
Figura 4.7. Carta de vulnerabilidade para as estruturas na área piloto.....	212
Figura 4.8. Carta final de risco para a área piloto.	213
Figura 4.9. Destaque da área de estudo utilizada na pesquisa.....	214
Figura 4.10. Curvas de probabilidade de ruptura em função da precipitação acumulada em 24 horas empregadas na pesquisa (Azevedo, 2011).....	215
Figura 4.11. Cenários de precipitação acumulada em 24 horas, que foram simulados na pesquisa para os meses de abril.....	217

Figura 4.12. Carta de ameaça para a configuração de chuva referente ao mês de abril, para o ano de 2010.	218
Figura 4.13. Carta de ameaça para a precipitação relativa ao mês de abril, para o ano de 2011.	218
Figura 4.14. Carta de ameaça para a configuração de chuva referente ao mês de abril, para o ano de 2012.	219
Figura 4.15. Carta de solos para a área analisada conforme Reatto et al. (2004).	223
Figura 4.16. Carta de probabilidade de ruptura para a chuva diária do mês de novembro.	224
Figura 4.17. Carta de probabilidade de ruptura para a chuva diária do mês de dezembro.	225
Figura 4.18. Carta de probabilidade de ruptura para a chuva diária do mês de janeiro.	225
Figura 4.19. Carta com as fontes simuladas dos deslizamentos.....	227
Figura 4.20. Carta com a simulação para a iteração 25.....	227
Figura 4.21. Carta com a simulação para a iteração 50.....	228
Figura 4.22. Carta com a simulação para a iteração 75.....	228
Figura 4.23. Carta com a simulação para a iteração 100.....	229
Figura 4.24. Carta de susceptibilidade à ocorrência de deslizamentos pelo uso do modelo SHALSTAB.	234
Figura 4.25. Carta de probabilidade de classificação de áreas muito instáveis, obtida pela aplicação do método de Rosenblueth (1975).	236

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

A: Área da seção transversal onde ocorre o fluxo

A: Parâmetro dependente do tipo de solo e do pico da tensão cisalhante na ruptura

A: Bacia drenada

a: coeficiente da função gaussiana para a probabilidade de ruptura

a: Limite inferior do intervalo real utilizado no Teorema de Bolzano

a: Área de contribuição

a(t): Aceleração do elemento em fluxo

a₁: Termos que compõem as funções relacionadas ao elemento de fluxo

a₂: Termos que compõem as funções relacionadas ao elemento de fluxo

a₃: Termos que compõem as funções relacionadas ao elemento de fluxo

a₄: Termos que compõem as funções relacionadas ao elemento de fluxo

asc: Extensão para arquivo tipo ASCII

ASCII: American Standard Code for Information Interchange

ASTER: Advanced Spaceborne Thermal Emission and Reflection Radiometer

ANA: Agência Nacional de Águas

b: Limite superior do intervalo real utilizado no Teorema de Bolzano

b: coeficiente da função gaussiana para a probabilidade de ruptura

b: Parâmetro para o cálculo da velocidade do fluxo subsuperficial que depende do tipo do solo

b: Largura do elemento da encosta ao longo da direção tangente ao contorno topográfico local

b: Comprimento do elemento principal da lâmina de detritos

b: Largura da fatia de solo

BC: Campo para tipo estrutural referente a paredes de tijolo e concreto

B₀: Intercepto da regressão logística

B_n: Coeficiente de regressão

C: Velocidade média da massa deslizando

C: Consequências

c: Coeficiente da função gaussiana para a probabilidade de ruptura

c: Coesão do solo

C': Coesão total do solo

c': Coesão efetiva do solo

cell: Célula central analisada

CV: Coeficiente de variação
C_v: Coeficiente de adensamento
C_x MAX: Parâmetro de correção máximo a ser adotado
C_x MIN: Parâmetro de correção mínimo a ser adotado
C₁: Parâmetro de correção para o armazenamento estático máximo
C₂: Parâmetro de correção para a evapotranspiração
C₃: Parâmetro de correção para a capacidade de infiltração
C₄: Parâmetro de correção para a capacidade de percolação
C₅: Parâmetro de correção para a capacidade de fluxo subterrâneo
C₆: Parâmetro de correção para a velocidade de fluxo superficial
C₇: Parâmetro de correção para a velocidade subsuperficial
C₈: Parâmetro de correção para a velocidade do fluxo de base
C₉: Parâmetro de correção para a velocidade de canal
C₁₀: Parâmetro de correção para a coesão efetiva do solo
C₁₁: Parâmetro de correção para o ângulo de atrito
C₁₂: Parâmetro de correção para a espessura do solo
C₁₃: Parâmetro de correção para o armazenamento gravitacional
C₁₄: Parâmetro de correção para o fator de segurança
c_φ: Coesão aparente
d: Desvio padrão relativo ao nível de confiança usado no Método de Monte Carlo
d: Profundidade do lençol freático medida na direção vertical
d: Fração da massa transferida a uma determinada célula
dt: Resolução temporal
dv(t)/dt: Derivada da velocidade do elemento em relação ao tempo
dx: Tamanho da célula
D_{dpt}: Profundidade de detritos
Dez.: Dezembro
D_{fod}: Profundidade da fundação da estrutura
D_S: Índice de deslocamento da fundação ou da estrutura
D_{S,T}: Valor limite do deslocamento da fundação ou da estrutura
D₀: Difusividade máxima característica
D₁: Volume que entra no tanque T1
D₂: Volume que entra no tanque T2

D_3 : Volume que entra no tanque T3

D_4 : Volume que entra no tanque T4

D_5 : Volume que entra no tanque T5

e: número neperiano

Embrapa: Empresa Brasileira de Pesquisa Agropecuária

erfc: Função de erro complementar

Evp: Evapotranspiração potencial

exp: Exponencial

E_1 : Saída de água por evapotranspiração

e_1 : Parâmetro relacionado ao tipo de superfície

E_2 : Escoamento superficial

E_3 : Escoamento subsuperficial

E_4 : Fluxo de base

E_5 : Fluxo de canal

$E[F]$: Valor médio ou esperado da função F

$E[FS]$: Valor médio ou esperado do fator de segurança

FOSM: First Order Second Moment

E: Elementos em risco

$E(x_i)$: Valor médio da variável x_i

FS: Fator de segurança ou função associada ao fator de segurança

FS_i : Fator de segurança obtido pela substituição, alternadamente, dos valores das variáveis independentes

FS_{++} : Fator de segurança obtido pela combinação de variáveis probabilísticas

FS_{+-} : Fator de segurança obtido pela combinação de variáveis probabilísticas

FS_{-+} : Fator de segurança obtido pela combinação de variáveis probabilísticas

FS_{--} : Fator de segurança obtido pela combinação de variáveis probabilísticas

FS_{crit} : Valor do fator de segurança crítico relacionado a estabilidade do talude

$f(x)$: Função contínua

$f^{(n)}(\bar{x})$: n-ésima derivada da função $f(x)$ sobre o ponto $x = \bar{x}$

$F(x,y)$: Função de duas variáveis

$F(\bar{x},\bar{y})$: Função de duas variáveis avaliada nos pontos \bar{x} e \bar{y}

$F(\bar{x}_1,\bar{x}_2,\dots,\bar{x}_N)$: Função F avaliada nos pontos médios das N variáveis

$FS(\bar{x}_1,\bar{x}_2,\dots,\bar{x}_N)$: Função do fator de segurança avaliada nos pontos médios das N variáveis

G: Campo para o estado de manutenção denominado bom

G_s : Gravidade específica do solo

GUI: Graphical User Interface

GUIDE: GUI Development Environment

$G(x)$: Função de desempenho que estabelece a variável dependente

$G(x)_i$: Cada um dos valores obtidos para a variável dependente pela aplicação de $G(x)$

$G(\bar{X})$: Média de uma amostra obtida pela aplicação da função $G(X)$

g: Aceleração da gravidade

H: Espessura do solo

H: Risco natural

H: Campo para a altura de edifícios altos

h: hora

h: Diferença entre a raiz real e sua estimativa

h: Espessura total da camada de solo

h: Altura da fatia de solo

\bar{h} : Altura do elemento principal da lâmina de detritos

H_{central} : Altura de solo presente em um pixel com deslizamento

H_g : Altura da construção acima do solo

H_{NBi} : Espessura de detritos na célula pertencente a vizinhança para qual se calcula a transferência de massa

h_s : Comprimento do caminho de percolação da água

I: Valor da integral

I: Intensidade da ameaça

i: Índice para variáveis dinâmicas que varia de 1 a 3

i: Número de variáveis consideradas randômicas

i: Referência ao intervalo de chuva

IAEG: International Association for Engineering Geology and the Environment

I_{dfm} : Fator de deformação

I_{dpt} : Fator de profundidade de detritos

I_{dym} : Fator de intensidade dinâmica

I_{gem} : Fator de intensidade geométrica

ISSMGE: International Society of Soil Mechanics and Geotechnical Engineering

I_t : Intensidade da chuva

I_z : Razão de infiltração na superfície do solo

$(I_z)_{\text{estacionário}}$: Taxa de infiltração da chuva antecedente na superfície do terreno na direção normal do talude

j : Índice de coluna

Jan.: Janeiro

k : Índice de linha

kg: Quilograma

km²: Quilômetro quadrado

kN: Kilonewton

kPa: Kilopascal

K_p : Permeabilidade do saprólito

K_{pp} : Permeabilidade da rocha

K_s : Permeabilidade do solo residual

K_t : Capacidade global de drenagem

K_x : Condutividade hidráulica na direção paralela ao talude

K_z : Condutividade hidráulica em relação à direção normal da encosta

L : Comprimento finito do talude

L : Campo para a altura de edifícios baixos

L : Comprimento de contorno para drenagem

L : Dimensão do pixel

LANDSAT: Land Remote Sensing Satellite

l : Comprimento de transição da lâmina de detritos

LS: Campo para tipo estrutural referente à estrutura muito leve

LT: Campo para tipo estrutural referente à estrutura leve

L_m : Comprimento do segmento m que compõe a divisão do talude

M : Campo para o estado de manutenção denominado médio

M : Campo para a altura de edifícios médios

m : Metro

m : Espessura adimensional da porção saturada do solo

m : Massa

m : Número de termos empregados na composição das séries das funções para o elemento principal de fluxo

m : Número de variáveis envolvidas no método de Monte Carlo

m: expoente

m_i : Elemento que representa as variáveis de entrada dinâmicas

m_t : Espessura adimensional da porção saturada do solo no passo de tempo atual

m_{t+1} : Espessura adimensional da porção saturada do solo atualizada

mm: milímetros

m_1 : Valor inicial da espessura adimensional da porção saturada do solo

Matlab: Matrix Laboratory

Max: Operador que retorna o maior dos valores apresentados como argumentos

MDF: Multiple Flow Direction

Min: Operador que retorna o menor dos valores apresentados como argumentos

MNT: Modelo numérico de terreno

mod: Operador que retorna o resto da divisão entre dois números

MS: Campo para tipo estrutural referente à estrutura mista

N: Newton

N: Total de simulações executadas na bacia

N: Força normal atuando na fatia de solo

n: Coeficiente de Manning

n: Porosidade

n: Quantidade de elementos computacionais da bacia empregados na simulação

n: Índice para os termos do somatório da série que define a poropressão

N_{B_i} : Célula pertencente a vizinhança para qual se calcula a transferência de massa

N_{B_j} : Células que compõem a vizinhança

N_H : Dardos que atingiram a região hachurada

N_M : Dardos que não atingiram a região hachurada

N_{mc} : Número de iterações para o método de Monte Carlo

Nov.: Novembro

n_s : Quantidade de fatores de resistência

n_w : Função integrante que compõe o fator de segurança

NASA: National Aeronautics and Space Administration

N_T : Número total de análises

O_t : Saída de água de uma porção finita do talude

P: Campo para o estado de manutenção denominado pobre

P: Peso do elemento do fluxo

P: Peso relativo do ponto μ_x para o método das estimativas pontuais

P_+ : Peso relativo do ponto x_+ para o método das estimativas pontuais

P_- : Peso relativo do ponto x_- para o método das estimativas pontuais

P[T]: Evento de ameaça

P[C|T]: Probabilidade condicional (dada à ocorrência de uma falha T, haja a definição de danos C)

p: Medida de precipitação acumulada

p: Probabilidade da variável dependente

Pr: Ameaça

PR: Probabilidade de ruptura

q: Quantidade de chuva

r: Raiz da função $f(x)$

R: Resistência do elemento em risco

R: Risco

RC: Campo para tipo estrutural referente a concreto armado

RiskLab: Risk-Landslide-Laboratory

RF: Campo para tipo estrutural reforçado

$R(T^*)$: Termo utilizado no cálculo da carga hidráulica total

R_S : Risco específico

R_t : Risco total

R_{str} : Resistência de estruturas

r_0 : Termo usado para descrever a razão $u_b(0)/(u_{b0})_{max}$ ou $u_t(0)/(u_{t0})_{max}$

R_N : Resto associado a série de Taylor

R_1 : Chuva

R_2 : Excesso de água enviado ao tanque 2

R_3 : Volume decorrente da infiltração

R_4 : Volume decorrente da percolação

R_5 : Fluxo de saída de água subterrânea

S: Campo para a altura de estruturas com andar único

S: Expressão envolvida na série infinita para as funções do elemento em fluxo

S: Força resistente desenvolvida ao longo da superfície de ruptura

s: Segundos

s: Porção saturada da altura do elemento principal da lâmina de detritos

S_m : Sequência numérica
 S_r : Grau de saturação do solo
 S_1 : Volume de água atual no tanque 1
 S^*_1 : Volume de água no tanque T1 ao fim do passo de tempo anterior
 S_{1max} : Capacidade máxima do tanque T1
 S_2 : Volume de água atual no tanque 2
 S^*_2 : Volume de água no tanque T2 ao fim do passo de tempo anterior
 S_3 : Volume de água atual no tanque 3
 S^*_3 : Volume de água no tanque T3 ao fim do passo de tempo anterior
 S_{3max} : Capacidade máxima do tanque 3
 S_4 : Volume de água atual no tanque 4
 S^*_4 : Volume de água no tanque T4 ao fim do passo de tempo anterior
SARD: Sistema de análise de risco devido a deslizamentos
SHALSTAB: Shallow Landsliding Stability Model
SHIA: Simulación Hidrológica Abierta
SIATA: Sistema de Alerta Temprana de Medellín
SICAD: Sistema Cartográfico do Distrito Federal
SIG: Sistema de informação geográfica
SIGMA: Sistema de Información Geográfica para Medellín y sus Alrededores
TMPA: Multi-satellite Precipitation Analysis
SLIDE: Slope - Infiltration - Distributed Equilibrium
TRIGRS: Transient Rainfall Infiltration and Grid-Based Regional Slope-Stability
TRMM: Tropical Rainfall Measuring Mission
T: Duração do evento de chuva
T: Transmissividade do solo
T: Força desestabilizadora tangencial atuante na fatia de solo
 T^* : Termo utilizado no cálculo da carga hidráulica total
t: Tempo
tf: Número de passos impostos ao autômato celular
 t_0 : Estimativa inicial para a raiz da função $x(t)$
txt: Extensão para arquivo texto
Terracap: Companhia Imobiliária de Brasília
T1: Tanque de armazenamento estático

T2: Tanque de armazenamento superficial

T3: Tanque de armazenamento gravitacional

T4: Tanque que representa o aquífero

T5: Tanque que representa o canal

u: Poropressão

u_a : Poropressão na base da camada impermeável

u_b : Poropressão no topo da camada saturada

$u_b(0)$: Poropressão na base da camada impermeável no início do fluxo

$u_b(t)$: Poropressão basal em um dado instante

$(u_{b0})_{max}$: Excesso inicial de poropressão máximo na base da camada impermeável no início do fluxo

u(C): Custo provocado pelas consequências

US\$: Dólar americano

$(u_{t0})_{max}$: Excesso inicial de poropressão máximo no topo da camada saturada no início do fluxo

$u_t(0)$: Poropressão no topo da camada saturada no início do fluxo

u_0 : Poropressão inicial máxima

UTM: Universal Transversa de Mercator

V: Vulnerabilidade

V[F]: Variância da função F

V[FS]: Variância do fator de segurança

V[G(X)]: Variância de uma amostra obtida pela aplicação da função G(X)

v: Parâmetro que controla a dispersão do fluxo

VG: Campo para o estado de manutenção denominado muito bom

VP: Campo para o estado de manutenção denominado muito pobre

v(t): Velocidade do fluxo em função do tempo

V[x_i]: Variância de cada variável x_i

$v_{inicial}$: velocidade para o escoamento superficial inicialmente adotada

v_{cal} : velocidade calculada para o escoamento superficial

v_{cell} : Velocidade final do escorregamento em uma dada célula

v_{med} : velocidade convergente para o escoamento superficial

v_0 : Velocidade inicial atribuída ao elemento principal do fluxo no início do movimento

V1: Célula vizinha ao pixel central estudado

V2: Célula vizinha ao pixel central estudado
 V_2 : Velocidade do fluxo superficial
 v_2 : Velocidade na superfície da encosta
V3: Célula vizinha ao pixel central estudado
 V_3 : Velocidade do fluxo superficial
V4: Célula vizinha ao pixel central estudado
 V_4 : Velocidade do fluxo de base
V5: Célula vizinha ao pixel central estudado
 V_5 : Velocidade do canal
V6: Célula vizinha ao pixel central estudado
V7: Célula vizinha ao pixel central estudado
V8: Célula vizinha ao pixel central estudado
W: Peso do elemento de solo
 W_{fc} : Capacidade de campo
 W_{pwp} : Ponto de murcha do solo
 W_s : Saturação
 x_0 : Aproximação para a raiz da função $f(x)$
 x_n : n-ésima aproximação da raiz da função $f(x)$
 X_n : Variável independente na variação de Y
X: Variável independente
 x_i : Parâmetro estocástico
 \bar{x}_i : Parâmetro estocástico médio
 x_+ : Valor da variável independente acima da média
 x_- : Valor da variável independente abaixo da média
 $x(t)$: Distância percorrida pelo fluxo em função do tempo
 $x'(t)$: Derivada da função relativa à distância percorrida
Y: Variável dependente, variável resposta ou função que a define
 y_+ : Valor da função de desempenho avaliada em um ponto maior que a média μ_x
 y_- : Valor da função de desempenho avaliada em um ponto menor que a média μ_x
 y_μ : Valor da função de desempenho avaliada no ponto médio μ_x
Z: Espessura do solo medida verticalmente
Z: Variável normal padronizada

Z_{crit} : Valor crítico do fator de segurança para a estabilidade em termos da variável normal padronizada

z : Profundidade

z : Espessura do solo

z_i : Número aleatório

Z_{max} : Espessura do solo que torna nula a profundidade saturada necessária para deflagrar um deslizamento

Z_{min} : Profundidade imune à instabilidade

Z_r : Profundidade das raízes

Z_s : Espessura do solo

Z_T : Profundidade da camada impermeável

Z_w : Posição do nível freático medida a partir da superfície de deslizamento

Z_{wcrit} : Valor crítico da camada saturada capaz de provocar o deslizamento

Z_3 : Quantidade de água excedente do tanque 3

α : Coeficiente de descarga do reservatório linear

α : Parâmetro que retrata a tendência não linear da curva de coesão

α : Inclinação do talude

α_m : Inclinação do segmento m que compõe a divisão do talude

α_0 : Inclinação do talude na condição inicial do fluxo

β : Inclinação da encosta

β_i : Índice de confiabilidade do fator de segurança

β_0 : Máximo ângulo de inclinação em que o talude é sempre estável

Γ : Função integrante que compõe o fator de segurança

γ : Peso específico do solo

γ_s : Peso específico do solo

γ_{sat} : Peso específico saturado do solo

γ_w : Peso específico da água

Δm_i : Variação da espessura adimensional da porção saturada do solo no passo atual

Δs : Comprimento unitário do talude

Δt : Duração da chuva

δa : incremento imposto ao limite inferior do intervalo $[a,b]$

δb : incremento imposto ao limite superior do intervalo $[a,b]$

ϵ : Nível de confiança desejado no método de Monte Carlo

θ : Declividade da encosta no modelo SHALSTAB
 λ : Coeficiente de intensidade relacionado ao tipo de solo
 μ_{FS} : Média do fator de segurança
 μ_x : Média da distribuição para uma variável independente
 μ_{x_i} : Média da variável x_i
 ξ : Parâmetro associado ao tipo superfície
 ξ_i : Fator de resistência
 ξ_{sfd} : Fator de resistência para a profundidade da fundação
 ξ_{sht} : Fator de resistência para a altura da construção
 ξ_{smn} : Fator de resistência para o estado de manutenção
 ξ_{sty} : Fator de resistência para a tipologia estrutural
 ρ : Coeficiente de correlação entre variáveis
 ρ_s : Densidade do solo
 ρ_w : Densidade da água
 σ_{FS} : Desvio padrão para fator de segurança
 σ_x : Desvio padrão para a variável x
 σ_{x_i} : Desvio padrão para a variável x_i
 τ : Resistência do solo
 ν_x : Coeficiente de assimetria
 ϕ : Ângulo de atrito efetivo do solo
 ϕ' : Ângulo de atrito efetivo do solo
 φ : Ângulo de atrito interno
 $\psi(Z, t)$: Carga hidráulica total
 $\psi(Z, 0)$: Carga hidráulica no início do evento pluviométrico
 Ω : Função integrante que compõe o fator de segurança
 $^\circ$: Graus
 ∂F_i : Variação processada na função F , a qual sucede quando se aplica ∂x_i
 ∂x_i : Taxa de variação da variável x_i
 ∞ : Infinito

1 INTRODUÇÃO

Os desastres naturais referem-se ao conjunto de eventos danosos que se encontram completamente fora do controle das pessoas, e que, muitas vezes, são agravados indiretamente pela intervenção humana (Alimohammadlou et al., 2013). Casos dessa natureza vêm apresentando um crescimento considerável com relação à quantidade de ocorrências registradas em diversas partes do mundo nas últimas décadas (Coutinho, 2010). Conforme Souza et al. (2009), a recorrência e os impactos associados a estes processos mostram-se cada vez mais intensos, proporcionando implicações significativas que se refletem sob a forma de perdas e danos de natureza social, econômica e ambiental. O desenvolvimento destas expressivas consequências, segundo estudos, pode estar diretamente relacionado às mudanças climáticas globais (Marcelino, 2007).

Alimohammadlou et al. (2013) afirmam que algumas ameaças são conhecidas por terem maior recorrência, como secas, tempestades, inundações, terremotos, vulcões e temperaturas extremas. Segundo García (2004), apesar das inundações e terremotos se apresentarem como os desastres naturais mais catastróficos, pode-se atribuir aos movimentos de massa, respeitáveis danos socioeconômicos. Esses danos são muito mais proeminentes do que geralmente se acredita, configurando-se como uma das causas mais frequentes de perdas humanas e econômicas ao redor do mundo, tendo, muitas vezes, seu potencial de dano subestimado (Sidle & Ochiai, 2006; Thiebes et al., 2014). Conforme Herath & Wang (2009), os deslizamentos configuram-se como a sétima maior causa de morte entre as catástrofes naturais, contribuindo para 17% de casos de mortalidade.

Yalcin (2011) verificou que as mortes e os prejuízos econômicos trazidos pelos casos de deslizamento são maiores do que geralmente reconhecidos, chegando a causar, em alguns países, como a Turquia, uma perda anual atrelada à propriedade maior que qualquer outro desastre natural. Conforme Turner (1996), o dano econômico nos Estados Unidos varia de 1 a 2 bilhões de dólares, com a associação de 25 a 30 mortes. Na China, o valor direto anual de perdas econômicas é estimado em, aproximadamente, 1,3 bilhões de dólares, com cerca de 900 perdas de vidas (Yin, 2009).

Os movimentos de massa configuram-se como eventos de caráter potencialmente desastroso, principalmente em zonas tropicais montanhosas, uma vez que a susceptibilidade do terreno em desencadear estas formas de processo aumenta em função de suas características fisiológicas e das atividades antrópicas (Schuster, 1996). Em diversos países,

deslizamentos rápidos e de grandes dimensões ameaçam regiões densamente povoadas com infraestruturas modernas. Ressalta-se que a crescente e indiscriminada ocupação urbana que se processa em áreas desfavoráveis, sem o apropriado planejamento do uso do solo e com a ausência da incorporação de técnicas adequadas de estabilização, vem acarretando o desenvolvimento de acidentes vinculados a processos desta natureza que, por diversas vezes, tomam proporções de desastres (Tominaga, 2007).

Assim sendo, torna-se necessária a inclusão de considerações sobre esse tipo de acidente natural em abordagens sobre o uso do solo e no planejamento de ações de emergência para a segurança pública, bem como para a elaboração e execução de projetos seguros de engenharia (Fell et al., 2008a). Conforme Motamedi & Liang (2014), a avaliação da ameaça de deslizamentos tem desempenhado, nos últimos anos, um papel importante no desenvolvimento de normas de utilização do solo com o objetivo de minimizar a perda de vidas e os danos às propriedades. Devido às consequências e aos impactos trazidos pelos escorregamentos de terra e, na tentativa de reduzir estas perdas, instituições governamentais e de pesquisa vêm buscando avaliar a susceptibilidade, a ameaça e o risco atrelado a estes eventos, conduzindo a mapas destes aspectos, no sentido de fornecer informação espacial e para auxiliar no ordenamento territorial, contribuindo para a prevenção de danos e uma melhor preparação contra desastres (Felicísimo et al., 2013; Thiebes et al., 2014).

Em um primeiro momento, seria possível mitigar o risco por intermédio de obras passivas, capazes de interceptar a massa de solo mobilizada, mas estas medidas são restritas em virtude do seu alto custo e da falta de grandes áreas disponíveis para comportar este tipo de solução, sendo que, frequentemente, o impacto ambiental é muito elevado (Olivares et al., 2014). A desocupação das áreas ameaçadas mostra-se como uma alternativa, sendo viável apenas no caso em que a quantidade de pessoas em risco é limitada. No entanto, essa medida extrema geralmente encontra resistência por parte dos moradores que não estão dispostos a abandonar suas terras e propriedades (Olivares et al., 2014).

Por fim, resta a adoção de um sistema de alerta de deslizamentos, sendo que, para isso, o ponto mais importante a ser considerado neste caso, corresponde à adoção de procedimentos de previsão totalmente confiáveis relativos à inicialização da ruptura dos taludes, abordando aspectos temporais e espaciais (Mathew et al., 2014; Olivares et al., 2014).

1.1 MOTIVAÇÃO

Chacón et al. (2006) destacam que a avaliação das ameaças trazidas pelos movimentos de massa e a capacidade de prever estes tipos de eventos têm se mostrado como um assunto de grande interesse para a comunidade científica.

Para Cascini (2005), nas últimas décadas, a literatura técnica mostrou um desenvolvimento extensivo no que se refere ao zoneamento relativo à ameaça e susceptibilidade de deslizamentos, sendo que, atualmente, estes se configuram como ferramentas poderosas para evitar o desenvolvimento de áreas ameaçadas, além de se apresentarem como a forma mais eficiente e econômica para a redução de danos futuros e para a perda de vidas.

Entretanto, segundo Cascini et al. (2005), o zoneamento da ameaça e do risco não pode ser considerado como um tema simples, em virtude dos variados fatores envolvidos, dentre os quais pode-se destacar a falta de entendimento e aceitação dos conceitos de risco e ameaça, tanto por parte dos políticos como pela população. Cascini (2008) enfatiza, ainda, que a ausência de procedimentos padronizados restringe a utilização de mapas com estes temas, de maneira que a aplicação de diferentes terminologias e métodos gera, com frequência, confusão, dificultando, assim, a comparação entre contextos geoambientais diferentes.

Uzielli et al. (2008) estabelecem que a análise de risco referente a deslizamentos é inerentemente complexa, sendo que as dificuldades de obtenção de resultados confiáveis em relação a este tipo de ameaça natural, quando comparados com os demais, encontram-se destacadas na literatura técnica. Uzielli et al. (2008) destacam, ainda, que estes problemas derivam da complexidade envolvida na modelagem da ameaça de deslizamentos, da intensidade e da vulnerabilidade ligada à formação do meio físico envolvido.

Assim sendo, baseando-se em diferentes trabalhos selecionados no meio técnico, a presente pesquisa propôs-se a estruturar uma ferramenta computacional capaz de fornecer e integrar as diversas componentes constituintes do arcabouço teórico que fazem parte da abordagem de risco dos deslizamentos, permitindo, ao final, o cálculo e a mensuração da estimativa desta grandeza.

1.2 OBJETIVOS

O objetivo desta tese é desenvolver uma ferramenta voltada para a obtenção do risco, de forma quantitativa, devido a escorregamentos rasos em solo deflagrados por chuvas, o qual possa ser representado de maneira espacial a partir da estruturação e modelagem de atributos ambientais considerados relevantes para este fim, visando ao direcionamento de ações propostas para o planejamento territorial e aplicações centradas no desenvolvimento de sistemas de alertas para a ocorrência destas categorias de movimentos de massa.

Os objetivos específicos são:

- Criação de um sistema computacional composto pela integração dos diferentes módulos responsáveis pela avaliação das etapas concernentes à análise do risco, de modo que o software produzido apresente uma variedade quanto às opções de modelos empregados para os cálculos, assim como uma flexibilidade em relação à sua entrada de dados;
- Estabelecimento de um vínculo entre a ferramenta computacional criada e uma plataforma pautada em sistemas de informação geográfica, utilizando-se esta última no que diz respeito ao pré-processamento e organização dos insumos necessários pelo sistema e na etapa de pós-processamento dos dados gerados nas análises;
- Aplicação do sistema proposto a variados casos de estudo, compostos por distintas localidades, com ênfase nas características tanto de caráter de preditivo, representado pelos sistemas de alerta, quanto naquelas voltadas para avaliação e desenvolvimento de uma estruturação adequada de ocupação.

1.3 ESCOPO DO TRABALHO

O presente trabalho encontra-se dividido em cinco capítulos, acrescentando-se ainda as referências bibliográficas e o apêndice com a listagem da programação. O primeiro capítulo contextualiza o tema a ser versado na pesquisa, mostrando sua relevância e motivação, bem como na exposição dos objetivos a serem alcançados.

A revisão bibliográfica, a qual se encontra explicitada no segundo capítulo, corresponde ao estabelecimento dos aspectos de maior destaque sobre a ocorrência de movimentos de massa presentes na literatura especializada, com enfoque nos eventos determinados pela ação de precipitações. Abordam-se, também, questões sobre a avaliação do risco dos deslizamentos, assim como a natureza das incertezas presentes no contexto da

engenharia geotécnica e os fundamentos atrelados aos métodos probabilísticos. A última parte do capítulo corresponde à apresentação dos modelos teóricos aplicados à análise de estabilidade de taludes e à vulnerabilidade, os quais serviram de base para a estruturação do sistema de análise de risco.

O terceiro capítulo explica todo arcabouço metodológico desenvolvido no presente trabalho, no sentido de gerar o resultado final em termos de risco devido a deslizamentos, dando-se ênfase aos aspectos relevantes da implementação computacional dos modelos aplicados em cada etapa do processo.

Os resultados e as análises provenientes da execução de exemplos realizados no sentido de averiguar a aplicação e as potencialidades do sistema desenvolvido, estão mostrados no quarto capítulo. Por fim, o quinto capítulo traz as conclusões geradas nesta tese, bem como as recomendações para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 DESLIZAMENTOS DEFLAGRADOS POR EVENTOS DE CHUVAS

Esta seção busca expor as principais características relativas às ocorrências de movimentos de massa, fornecendo, em um primeiro instante, uma visão geral destes processos e, posteriormente, concentrando-se naqueles definidos especificamente pela ação da precipitação, destacando-se, ainda, o papel deste fator na deflagração dessa categoria de evento. De forma adicional, discorre-se sobre as práticas geralmente usadas na análise espacial de deslizamentos detonados por chuvas, com auxílio de sistemas de informação geográfica, onde são apresentados os principais pontos relativos à sua aplicação, além da definição da visão geral do modelo de engenharia adotado como base para o desenvolvimento da abordagem probabilística.

2.1.1 ASPECTOS GERAIS DE MOVIMENTOS DE MASSA DETONADOS POR CHUVAS

Componentes dos processos superficiais que modelam o relevo, os movimentos de massa têm sua origem relacionada a uma variedade extensa de atividades geológicas, hidrometeorológicas, químicas e mecânicas que têm lugar na crosta terrestre (Entralgo, 2013). As encostas, visando a um estado central compatível com o equilíbrio em relação aos processos nelas atuantes, no sentido de obter um arranjo mais eficiente possível, apresentam uma tendência frequente a mudanças (Bloom, 1988). Neste sentido, enquanto se considera que os movimentos de natureza tectônica tendem a originar montanhas, as ações do intemperismo, os sismos, as chuvas e demais eventos (como a própria atuação antrópica), aplicadas às montanhas, proporcionam sua desestabilização e a conseqüente alteração do relevo, até então existente, para uma condição mais plana.

Para Skempton & Hutchinson (1969), a evolução das ocorrências de movimentos de massa compreende as deformações de pré-rupturas, a ruptura em si e os deslocamentos que se processam em um estágio após a falha, sendo que o termo ruptura pode ser entendido, conforme Leroueil et al. (1996), como o único episódio de movimento mais significativo na conhecida ou prevista história do deslizamento, envolvendo, em geral, a primeira formação de uma superfície de ruptura completamente desenvolvida.

García (2004) destaca que uma diretriz básica na realização de uma análise que envolva o risco e as conseqüências do deslizamento está vinculada à identificação dos tipos de

movimentos de massa, uma vez que estes episódios podem diferir em termos de forma, dimensão, velocidade, materiais envolvidos, estado de atividade, entre outras características, atentando, ainda, que este reconhecimento pode ser realizado por intermédio de técnicas de sensoriamento remoto ou estudos de campo.

A classificação dos movimentos de massa proposta por Varnes (1978), posteriormente modificada por Cruden & Varnes (1996) e Hutchinson (1988), é amplamente aceita, sendo que a utilidade deste sistema é atestada diante da popularidade por ele alcançada, tanto na América do Norte, em seu formato original, quanto em todas as outras localidades, em suas variações (Hungar et al., 2014). Coronado (2006) atenta que os critérios básicos considerados nesta classificação referem-se à natureza do material e ao tipo do movimento. Na Tab. 2.1 encontram-se representadas as categorias de movimentos de massa segundo esta classificação.

Tabela 2.1. Resumo da classificação dos movimentos de massa (Cruden & Varnes, 1996).

Tipo de Movimento	Tipo de Material		
	Rocha	Solos	
		Predominantemente grosso	Predominantemente fino
Quedas	Queda de blocos	Queda de detritos	Queda de solo
Tombamentos	De rocha	De detritos	De solo
Deslizamentos (Escorregamentos)	Rotacionais de rocha	Rotacionais de rocha e solo	Rotacionais de solo
	Translacionais de rocha	Translacionais de rocha e solo	Translacionais de solo
Expansões laterais	De rocha	De detritos	De solo
Fluxos (Escoamentos)	Rastejo de rochas (creep)	Rastejos e corridas	
Complexos	Combinação de dois ou mais tipos de movimentos		

Olivares et al. (2014) destacam que, em muitos países, movimentos de massa rápidos e extensos ameaçam regiões densamente ocupadas, nas quais se encontram instaladas modernas infraestruturas. Conforme ainda Olivares et al. (2014), o risco é extremamente alto em áreas de encostas íngremes constituídas por material granular não saturado, em função do curto

tempo de resposta disponível após o evento ser detonado, com a conseqüente impossibilidade de proteção de pessoas e propriedades, bem como a vulnerabilidade das estruturas expostas ao impacto de movimentos de massa providos de uma alta energia cinética.

Segundo Thanh & De Smedt (2013), a base de muitos métodos voltados para a avaliação de estabilidade de taludes está vinculada à identificação dos fatores que causam os movimentos de massa. Em conformidade com esta afirmação, Silva (2006) afirma que o entendimento dos fatores deflagradores destes eventos torna-se essencial para o estabelecimento de decisões corretas no sentido de mitigar as suas conseqüências.

Pode-se dizer que os movimentos de massa resultam da paulatina mudança imposta por processos naturais, como o intemperismo, ou por atividades antrópicas às propriedades mecânicas dos materiais que compõe um talude (Soeteres & Van Westen, 1996). Por outro lado, segundo Wang & Sassa (2006), uma encosta pode ser solicitada como resultado de uma resposta quase imediata a estímulos externos, os quais podem se configurar como um único fator, tal como uma chuva intensa, um evento sísmico ou mesmo marés atmosféricas, que ocasionam, desta forma, uma redução da resistência ao cisalhamento ou um aumento dos esforços atuantes.

Afirma-se que uma cadeia de eventos, possuindo diversas vezes um caráter cíclico, originando-se com o surgimento da própria rocha, juntamente com a sua história geológica e geomorfológica seguinte, é que controla a ocorrência das instabilidades de encostas e taludes. Desta forma, conforme Broms (1975), pode-se estabelecer que condições geológicas, hidrológicas e fatores como o clima e a topografia afetam a estabilidade do talude, corroborando para o início de uma movimentação. Fernandes & Amaral (1996) atentam que vários tipos de feições geológicas, como fraturas, falhas, foliações e descontinuidades, podem atuar como condicionantes da ocorrência de movimentos de massa. Estes fatores são responsáveis pela distribuição e estabelecimento dos acidentes, em se tratando do ponto de vista temporal e espacial. Além disso, em um talude, a principal força de natureza instável que exerce influência é a gravidade, a qual usualmente se desenvolve pela combinação entre o peso próprio do talude e o peso relativo de um carregamento aplicado sobre o mesmo (Krynine & Judd, 1957).

Não obstante, determinar um grupo de condicionantes que atuam diretamente e de maneira imediata nesses acontecimentos é quase sempre possível (Augusto Filho & Virgili, 1998). Assim, os movimentos de massa têm sua distribuição e ocorrência atreladas a fatores

que podem ser agrupados em duas categorias, podendo estas serem denominadas variáveis condicionantes ou variáveis dinâmicas (detonantes).

As variáveis dinâmicas, as quais podem ser exemplificadas pelo grau de saturação, atuam no controle dos fatores desencadeantes de uma encosta susceptível a movimentos de massa, sendo que estas variáveis são controladas pelo clima, pela atividade humana e por processos hidrológicos que caracterizam, também, os padrões temporais dos movimentos de massa (Crosta & Frattini, 2003). Já quando se faz referência às propriedades mecânicas do solo e à topografia, que contribuem para a susceptibilidade da encosta e definem a distribuição espacial dos movimentos de massa, levam-se em conta as variáveis condicionantes.

Montoya (2013) afirma que os elementos deflagrantes que mais se destacam correspondem às atividades vulcânicas e antrópicas, assim como sismos e precipitações, sendo que este último fator apresenta-se como o principal responsável pela determinação de episódios de movimentos de massa em diversos países, como no caso da Colômbia e do Brasil.

Os eventos chuvosos, em geral, definem movimentos de massa denominados como escorregamentos rasos, os quais apresentam uma superfície de deslizamento planar. Pode-se considerar um deslizamento como um sistema físico que se desenvolve por meio de vários estágios no decorrer do tempo (Terzaghi, 1950; Leroueil et al., 1996). Este trabalho foca-se exatamente no estudo destes escorregamentos rasos detonados por chuvas.

Em todo o mundo, as rupturas planares de talude induzidas por chuvas mostram-se como um perigo natural que ameaça a segurança pública (Zhang et al., 2014), sendo que, no sentido de prever estes acidentes, grandes esforços têm sido realizados, a exemplo dos trabalhos de Iverson (2000), Alonso et al. (2003), Rahardjo et al. (2005), Li et al. (2009) e Zhang et al. (2011). Podem-se acrescentar pesquisas focadas no estabelecimento de modelagens responsáveis por analisar, de maneira acoplada, os movimentos de massa (especificamente escorregamentos rasos) e a precipitação (Rahardjo et al., 2001; Gitirana, 2005; Karam, 2005; Godt et al., 2008; Salciarini et al., 2008; Cascini et al., 2010; Conte & Troncone, 2012).

Chuvas intensas proporcionam o incremento rápido da pressão nos poros ou promovem a perda da componente relativa à coesão aparente, o que determina a geração de movimentos deste tipo (Terlien, 1998; Crosta, 1998; Wang & Sassa, 2003; Crosta & Frattini, 2003). A evolução transcorre de forma que o material deslocado pela rápida redução na

resistência ao cisalhamento, em condições não drenadas e processos de liquefação (Anderson & Sitar, 1995), torne-se um fluxo que se espalha descendentemente com o aumento do volume inicial do material deslocado pelo transporte de sedimentos retirados do canal (Wieczorek et al., 2000; Wang & Sassa, 2003).

A infiltração das chuvas promove a redução da sucção, o que determina a diminuição da resistência ao cisalhamento do solo e explica, para Li et al. (2005), o processo no caso dos solos não saturados. Conforme Ribeiro Junior (2011), quando se trata de solos não saturados, a estabilidade das encostas é fortemente influenciada pela sucção. A sucção e a coesão aparente são muito sensíveis à variação de água no solo, de forma que a parcela de sucção pode ser anulada facilmente com o aumento da umidade, tornando-se, assim, um agente deflagrador de escorregamentos. A parcela de sucção desaparece quando o solo encontra-se totalmente saturado, de forma que o lençol freático gera uma poropressão superior a zero, o que favorece o acréscimo de instabilidade do talude, uma vez que, estando positiva, esta poropressão promove a redução da resistência ao cisalhamento. Em se tratando das formas de ruptura determinadas pela infiltração, há a possibilidade de acontecer por dois mecanismos diferentes. O acréscimo de pressão positiva nos poros, ocasionado pela liquefação do material, apresenta-se como a primeira forma de ruptura, enquanto que a segunda forma processa-se em virtude da redução da sucção, quando o material encontra-se ainda em um estado não saturado, em condições de poropressões negativas, sendo que a massa se comporta como um corpo rígido (Collins & Znidarcic, 2004).

Um limiar que relaciona intensidade e duração identifica as condições mínimas de chuva que provocam uma poropressão contrária à segurança em relação a uma determinada profundidade no talude (Godt et al., 2006). Para Matlan (2009), quando a duração da precipitação torna-se capaz para infiltrar no plano de escorregamento potencial considerado e, conseqüentemente, causar a redução da sucção, o mecanismo de ruptura do talude é induzido.

Assim, quanto à estabilidade de taludes, pode-se afirmar, segundo Gostelow (1991) e Iiritano et al. (1998), que o papel da precipitação pode atuar de duas formas distintas, sendo que os episódios muito intensos de chuvas proporcionam uma redução da resistência ao cisalhamento acompanhada da diminuição da coesão e de deslocamentos na massa superficial. Por outro lado, o aumento em longo prazo da pressão nos poros em uma região de ruptura potencial pode ser estabelecido por ação de chuvas, o que proporcionaria movimentos de massa de caráter mais profundo e por meio de superfícies potenciais de ruptura já desenvolvidas. Para Rahardjo et al. (2007), a magnitude na redução do fator de segurança é

diretamente proporcional à grandeza e à intensidade da precipitação, de forma que estas características, juntamente com as propriedades do solo, configuram-se como os fatores principais que controlam a instabilidade de encostas, tendo-se que, quanto mais intenso o evento pluviométrico, mais rápido será o decréscimo do fator de segurança.

Toll et al. (2001) verificaram que a chuva antecedente tem um papel importante em relação às respostas das medidas de poropressão durante um evento chuvoso, assim como na inicialização dos deslizamentos (Wieczorek, 1987), de forma que, quando seus valores iniciais encontram-se elevados, uma forte precipitação tem a capacidade de imprimir apenas uma pequena mudança na medida da poropressão. Por outro lado, quando as medidas iniciais de poropressão encontram-se baixas, um pequeno evento chuvoso é o bastante para proporcionar uma grande mudança em seus valores.

Segundo Aristizábal et al. (2010), há estudos que expõem a importância das precipitações de longas durações ou das chuvas antecedentes na determinação da ruptura de taludes que possuem baixa permeabilidade, uma vez que necessita-se de um tempo maior para que a água proveniente da precipitação infiltre em encostas que apresentam estas características, fazendo com que a restauração do fator de segurança perante o evento chuvoso seja muito lenta. Ao averiguarem a estabilidade das vertentes, levando-se em consideração a chuva antecedente, Rahardjo et al. (2001) identificaram que, tanto a chuva diária quanto a acumulada, possuem atuação na deflagração de deslizamentos, sendo que ambos os fatores devem ser analisados em conjunto, já que a permeabilidade do solo aumenta com o incremento da chuva antecedente, contribuindo, assim, para a chance de que posteriores precipitações promovam escorregamentos rasos.

No entanto, Giannecchini et al. (2012) atentam para o fato de que a influência imposta pela chuva antecedente é de difícil quantificação e depende, conforme Aleotti (2004), de variados fatores como condições climáticas locais, inclinação das vertentes e heterogeneidade dos solos, quanto às suas propriedades físico-mecânicas e a permeabilidade. Em ambientes tropicais, Terlien (1998) destaca que a profundidade da superfície de ruptura é influenciada pelo número de dias de chuva antecedente, de forma que escorregamentos rasos são determinados pela chuva diária e intensa, enquanto a chuva acumulada de vários dias associa-se a eventos de caráter relativamente profundo.

2.1.2 MODELOS FÍSICOS UTILIZADOS COM SISTEMAS DE INFORMAÇÃO GEOGRÁFICA PARA ANÁLISE DE DESLIZAMENTOS OCACIONADOS POR CHUVAS

Em se tratando dos métodos de análise de deslizamentos, segundo Silveira & Ramos (2007), em paralelo ao progresso da computação, a geotecnologia vem propiciando o desenvolvimento de inúmeras ferramentas que permitem a gerência dos mais variados aspectos do ambiente geográfico como recursos naturais, meio-ambiente, infraestrutura e urbanização, de forma que estes instrumentos proporcionam respostas aos questionamentos impostos em um tempo menor do que aquele requerido antes do seu surgimento.

Ramos et al. (2007) destacam que os sistemas de informação geográfica (SIG) se sobressaem no conjunto destas geotecnologias pelo fato de possibilitar a integração de um volume substancial de informações geográficas e alfanuméricas correlacionadas, ampliando, assim, as possibilidades de análises, juntamente com a otimização do tempo de recuperação e tratamento dos dados inter-relacionados. Avanços alcançados na tecnologia dos sistemas de informação geográfica e em ferramentas matemáticas e estatísticas voltadas para a modelagem e simulação têm permitido a crescente aplicação de técnicas de caráter quantitativo em vários campos das ciências da Terra (Carrara & Pike, 2008).

Os métodos de inferência espacial em conjunto com as técnicas de SIG assumem um papel de destaque, em se tratando da identificação de áreas susceptíveis aos movimentos de massa, permitindo a elaboração de prognósticos mais completos e confiáveis da área de interesse (Bispo et al., 2009). Da mesma forma, Souza (2004) ressalta que, na identificação das áreas mais afetadas pelos acidentes geotécnicos e, conseqüentemente, daquelas que possuem fatores geológicos e morfológicos associados à instabilidade de talude, as ferramentas de geoprocessamento apresentam grande utilidade.

Por meio da intersecção entre a distribuição espacial dos movimentos de massa conhecidos e os padrões espaciais dos elementos condicionantes considerados relevantes para o entendimento da instabilidade, pode-se estabelecer a distribuição espacial da susceptibilidade à ocorrência de deslizamentos (Carrara et al., 1998).

Conforme Reis et al. (2003), a elaboração de modelos de avaliação de susceptibilidade cada vez mais complexos fundamentados em métodos quantitativos foi permitida pelo uso de sistemas de informação geográfica, uma vez que propicia o cruzamento de um grande número

de temas cartográficos e a análise estatística da relação entre os deslizamentos identificados e as variáveis independentes espacializadas.

Desta forma, a fusão entre um bom modelo de avaliação de susceptibilidade aos movimentos de massa e os sistemas de informação geográfica geram um instrumento poderoso para os profissionais das áreas envolvidas na mitigação dos riscos e no controle do crescimento das cidades (García, 2004).

Segundo Guimarães et al. (2009), a utilização da modelação matemática, na tentativa de preconizar eventos naturais, tem aumentado consideravelmente, permitindo que estes modelos gerados estejam pautados em informações físicas concernentes à dinâmica dos fenômenos da natureza. Esta premissa possibilita ainda, segundo Guimarães et al. (2009), um controle mais eficaz sobre estes episódios, com a conseqüente previsão das características temporais e espaciais de suas ocorrências.

Os modelos constituem-se, desta forma, como uma opção para a presciência da ocorrência de movimentos de massa, sendo capazes de representar, simular, prever a susceptibilidade e até o risco provocado por esses acontecimentos na população (Zaidan & Fernandes, 2009). Endossando o exposto, existem inúmeras metodologias desenvolvidas incorporando modelos com o objetivo de prever escorregamentos (Dias, 2006), de forma que o efetivo uso destes modelos matemáticos decorre do entendimento das complexas interações entre os mecanismos detonantes e os fatores condicionantes (Fernandes et al., 2004).

Conforme Dias (2006), é possível categorizar estes modelos preditivos em duas vertentes. A primeira delas está relacionada aos modelos empíricos, sendo que uma parte deste grupo é fundamentada em inventários de deslizamentos, combinando variados planos de informação e se valendo de análises estatísticas. A outra parcela se baseia em leis físicas e, segundo Fernandes & Amaral (1996), esta permite o teste, em escalas que variam do nível de encostas a bacias de drenagem, de hipóteses bem específicas.

Segundo Borga et al. (2002), os modelos geotécnicos que são aplicados aos casos de movimentos de massa deflagrados por precipitação consideram, na sua formulação, inclinação constante e comprimento infinito, supondo que a superfície de ruptura esteja paralela à superfície do terreno e que o comprimento desta superfície de ruptura seja muito maior que a espessura de camada deslocada.

No intuito de prever a susceptibilidade a movimentos de massa, a maior parte dos métodos analíticos utiliza a integração entre modelos hidrológicos, capazes de realizar a previsão do fluxo subterrâneo e do superficial, e modelos de estabilidade para taludes infinitos

(Silva, 2006). Aristizábal et al. (2010) atentam que, embora os modelos em sua grande maioria empreguem propriedades dos solos na obtenção do fator de segurança vinculado na análise de estabilidade de taludes, eles se diferenciam no método pelo qual a poropressão é calculada.

Como exemplo, pode-se citar o modelo TRIGRS (Transient Rainfall Infiltration and Grid-Based Regional Slope-Stability), que leva em consideração a ação transiente das precipitações na determinação de escorregamentos rasos, de forma a combinar uma solução analítica para o fluxo subterrâneo unidimensional com o cálculo da estabilidade em talude infinito (Iverson, 2000; Baum et al., 2002).

Outro modelo utilizado na predição de escorregamentos translacionais rasos denomina-se SHALSTAB (Shallow Landsliding Stability Model), desenvolvido por Montgomery & Dietrich (1994), o qual, conforme Aristizábal et al. (2010), configura-se como um dos modelos físicos mais reconhecidos, privilegiando a topografia como elemento crucial na determinação de deslizamentos (Ramos et al., 2002). Guimarães et al. (2009) ressaltam que este método integra modelos de estabilidade de taludes e hidrológico, no sentido de determinar a precipitação capaz de alterar o estado de equilíbrio presente na encosta. Outros modelos de base física, como LISA (Hammond et al., 1992), SINMAP (Pack et al., 1998) e CHASM (Anderson & Lloyd, 1991), também podem ser citados como opções utilizadas na análise e previsão de deslizamentos, no sentido de evitar as consequências potencialmente danosas associadas às rupturas de taludes.

Nesse contexto, ressalta-se, ainda, a existência de variadas formas quanto ao tratamento da ameaça de deslizamentos em escala regional. Entre estas abordagens, destaca-se a utilização de limiares críticos na previsão de movimentos de massa com uso de métodos empíricos ou estatísticos, os quais podem ser incorporados de maneira eficaz a sistemas de alerta e cuja importância é evidenciada pelas inúmeras pesquisas presentes na área, podendo-se citar aquelas desenvolvidas por Tatizana et al. (1987), Caine (1980) e Guzzetti et al. (2008).

Outros autores, como Botero (2013), García (2004) e Souza (2004), vêm aplicando, com sucesso, redes neurais artificiais devidamente treinadas no tratamento de informações vinculadas ao problema da ocorrência de deslizamentos.

De outra maneira, a investigação do papel das precipitações no processo de indução de escorregamentos pela modelagem hidrogeológica mostra-se como uma ferramenta robusta,

principalmente com a sua integração aos sistemas de informação geográfica, conforme exemplificado nos trabalhos de Tan et al. (2008) e Yuan et al. (2005).

2.1.3 DEFINIÇÃO DOS FUNDAMENTOS DOS MODELOS DE ESTABILIDADE DE TALUDES ADOTADOS NA PESQUISA

Nesta seção, busca-se explicitar e fundamentar o modelo de engenharia adotado nas análises de estabilidade de taludes, o qual serviu de base para a aplicação das abordagens probabilísticas.

Silva (2006) estabelece que, em termos teóricos, três diferentes campos de força atuam em um talude (aquelas decorrentes do peso dos materiais e da percolação da água, contrárias à estabilidade, e as advindas da resistência ao cisalhamento, favoráveis à estabilidade), sendo que, na avaliação da estabilidade das encostas, considera-se o equilíbrio destas forças. De forma mecânica, a determinação de um escorregamento de massa ocorre quando as tensões solicitantes ultrapassam a resistência ao cisalhamento oferecida pelo solo depositado (Silva, 2005).

Conforme Pereira (2013), a análise de estabilidade de taludes se baseia em procedimentos voltados para a definição de um índice ou indicador que fornece a quantificação da proximidade de ruptura de uma determinada encosta, em função de um conjunto de condicionantes. Segundo Silva (2005), esse fator é obtido, matematicamente, ao se processar o balanço entre forças resistivas e cisalhantes, o qual pode ser traduzido como a razão entre a resistência cisalhante média e a tensão cisalhante ao longo da superfície crítica de ruptura, sendo conhecido como fator de segurança.

Deslizamentos translacionais ocorrem nos locais em que a presença de uma camada adjacente com resistência significativamente diferente influencia a forma da superfície de ruptura, determinando que muitas destas rupturas se processem pelo estrato com menor resistência ao cisalhamento (Craig, 2004). Nesses casos, a ruptura tende a ser plana e paralela ao talude, quando a camada adjacente situa-se a uma profundidade relativamente rasa abaixo da superfície do talude. Sharma (1996) afirma que rupturas de encostas com essas características podem ser analisadas como uma ruptura de talude infinito, podendo-se aplicar, nessas situações, segundo Jordan (2011), métodos de equilíbrio limite.

De forma a exemplificar a utilização desta abordagem, destaca-se a forma mais simples de equação para o modelo de talude infinito, correspondente ao caso de solos secos e sem coesão, explicitada por Jordan (2011). Na Fig. 2.1 encontra-se representado o diagrama

de corpo livre utilizado para se determinar as forças desestabilizadoras e as resistentes que atuam na encosta.

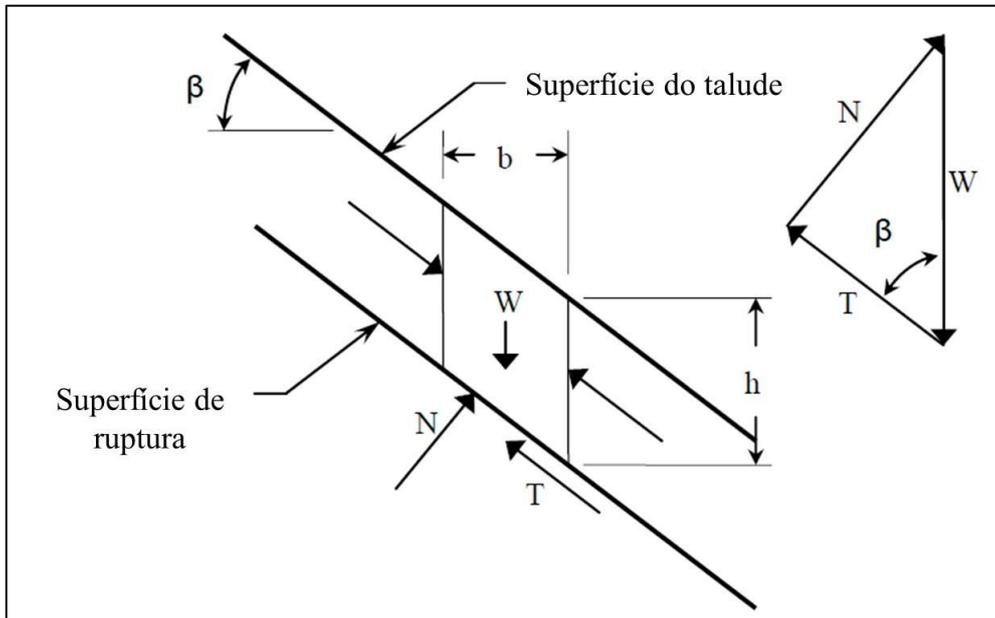


Figura 2.1. Diagrama de corpo livre da análise de talude infinito para solo seco e sem coesão (modificado de Jordan, 2011).

O peso de cada fatia, W , pode ser determinado a partir do peso específico do solo (γ), da largura e da altura da fatia, representadas por b e h , respectivamente, conforme equação abaixo:

$$W = \gamma.b.h.(1) \quad (2.1)$$

onde 1 refere-se ao comprimento da fatia correspondente à sua terceira dimensão. A força normal N e a força desestabilizadora T podem ser definidas conforme as seguintes expressões:

$$N = W.\cos(\beta) \quad (2.2)$$

$$T = W.\sin(\beta) \quad (2.3)$$

onde β é a inclinação da encosta. Pelo critério de ruptura de Mohr-Coulomb, a força resistente S desenvolvida ao longo da superfície de ruptura é dependente do ângulo de atrito interno do solo ϕ , podendo-se escrever a equação abaixo:

$$S = N \cdot \tan(\beta) \quad (2.4)$$

Pelo uso do princípio de equilíbrio limite, o fator de segurança, denominado FS, para um simples talude seco e sem coesão, pode ser escrito pelas equações a seguir:

$$FS = \frac{N \cdot \tan(\phi)}{W \cdot \sin(\beta)} \quad (2.5)$$

$$FS = \frac{\tan(\phi)}{\tan(\beta)} \quad (2.6)$$

Podem-se aplicar os mesmos conceitos de equilíbrio limite para outros casos de talude infinito incorporando, além da parcela promovida pelo atrito, a resistência fornecida pela coesão, e em condições em que o solo esteja saturado ou úmido, podendo-se considerar, também, a percolação.

A condição limite de estabilidade é representada por um fator de segurança igual a um (Miranda, 2005). Conforme Maia (2003), sabe-se que, na engenharia civil, projetos que apresentam fatores de segurança inferiores à unidade estão potencialmente sujeitos à ruptura. Desta forma, assume este valor crítico de estabilidade como referência para o desenvolvimento da abordagem probabilística e para a aplicação dos métodos probabilísticos associados.

2.2 ABORDAGEM PROBABILÍSTICA USADA EM ESTABILIDADE DE TALUDES

O estudo dos processos relacionados a eventos de escorregamentos passa, fundamentalmente, pela análise de estabilidade de taludes, a qual é realizada tradicionalmente por métodos determinísticos (Flores, 2008), sendo estes influenciados pelo julgamento do engenheiro e resultando, ao final, em uma única informação, que corresponde ao fator de segurança. O atual interesse no estudo de deslizamentos acarretou uma revigorada avaliação das equações, baseada em taludes infinitos como um modelo simples, capaz de ser utilizado para ponderar o valor deste fator de segurança em taludes longos (Griffiths et al., 2011).

Entretanto, em decorrência de inúmeros parâmetros do solo que possuem um alto grau de incerteza, o fator de segurança determinado a partir de métodos determinísticos não leva em consideração a aleatoriedade e a variabilidade das variáveis envolvidas no projeto, de maneira que o seu valor não demonstra exatamente a verdadeira segurança da encosta, na maioria das vezes (Ge et al., 2011). Montoya & Assis (2011b) ressaltam que nos últimos

anos, houve um grande interesse em incorporar metodologias que permitam estimar de maneira racional a incerteza associada a estudos geotécnicos, como é o caso dos métodos probabilísticos. Ríos & Carvajal (2011) atentam que a abordagem probabilística configura-se como um enfoque comum para estas categorias de problemas, uma vez que existe uma variabilidade espaço-temporal intrínseca aos parâmetros que compõe os modelos de equilíbrio limite. Nas próximas subseções, busca-se explicar sobre as incertezas naturalmente presentes em abordagens relacionadas à engenharia geotécnica, direcionando-se, especificamente, para a prática de estabilidade de encostas, bem como sobre a utilização de fundamentos estatísticos e probabilísticos focados no tratamento da variabilidade das características envolvidas nas análises de ocorrência de eventos de deslizamentos.

2.2.1 INCERTEZAS ASSOCIADAS À ENGENHARIA GEOTÉCNICA

Segundo Restrepo (2011), a caracterização das incertezas e dos riscos intrínsecos às obras geotécnicas encontra-se vinculada a eventos danosos de caráter geológico e/ou geotécnico. Para Rilho (2011), a presença de incertezas torna-se inevitável quando se trabalha com problemas reais.

Costa (2005) ressalta que este aspecto encontra-se inserido em todas as fases de uma obra geotécnica, partindo desde a caracterização do local, continuando presente pelas fases de análise e projeto e se estendendo até a tomada de decisões e construção, afirmando, ainda, que este fato é indissociável à profissão. Deste modo, Ang & Tang (2007) destacam a importância de distinguir a maioria das fontes de incertezas existentes em projetos de engenharia por parte dos profissionais envolvidos.

Tratando-se de uma primeira abordagem, as fontes de incertezas geotécnicas podem ser classificadas em incertezas de dados ou de modelos (Einstein, 2003). No processo de caracterização de estruturas de solo, Vanmarcke (1977) indica o emprego de três categorias voltadas para a caracterização das origens de incerteza:

- Heterogeneidade natural dos solos *in situ*;
- Limitação das informações passíveis de serem usadas na caracterização dos solos;
- Erros processados nas medidas das propriedades do solo.

Conforme Suchomel & Mašín (2009), observa-se que há uma marcante dispersão nas propriedades mecânicas do solo, as quais são alcançadas por intermédio de investigações geotécnicas detalhadas, sendo que tal resultado provém tanto da variabilidade espacial, intrínseca a estas características, quanto de erros relativos à medição. Suchomel & Mašín

(2009) atentam, também, que uma parcela adicional de incerteza é incorporada, já que apenas uma quantidade restrita de medições encontra-se geralmente disponível, derivando de uma calibração subjetiva referente a modelos constitutivos simples, os quais são frequentemente adotados em análises geotécnicas. Comumente, neste campo da engenharia, vale-se de conceitos deterministas na consideração destas incertezas, de forma que tal enfoque, entretanto, não promove o desenvolvimento do entendimento mais claro do peso atrelado às incertezas abrangidas nas simulações (Singh & Chung, 1991; Phoon & Kulhawy, 1999; Elkateb et al., 2003).

Na análise geotécnica, verifica-se que inúmeras fontes de incertezas podem ser reconhecidas, as quais variam desde os parâmetros do material envolvidos na amostragem até as técnicas empregadas nos ensaios (Griffiths et al., 2002). Nesse sentido, sobressaem-se, conforme Wu et al. (1996), como fontes de incertezas mais comuns em engenharia geotécnica, as cargas futuras e condições ambientais, juntamente com as condições de campo.

Conforme Rilho (2011), ficam evidentes as inúmeras possibilidades de ocorrências de incertezas capazes de atuar em um projeto geotécnico. Lacasse & Nadim (1996) distinguiram-nas em incertezas inerentes (ou naturais) e aquelas decorrentes da carência de conhecimento. Já para Baecher & Christian (2003), variabilidade natural, incertezas nos modelos de decisão e no conhecimento são consideradas como as três principais categorias de incertezas presentes na engenharia geotécnica.

A aleatoriedade proveniente dos eventos naturais encontra-se atrelada a variabilidade natural, apresentando-se como variabilidade no espaço, para ocorrências processadas em distintos locais em um mesmo instante (caracterizando, portanto, a variação espacial) ou aquela processada no tempo, no caso de fenômenos desenvolvidos em um mesmo local, chamando-se de variação temporal.

Para analisar a variabilidade natural, pode-se fazer uso de modelos matemáticos simplificados ou modelos físicos capazes de fornecer somente uma aproximação do evento natural que está sendo estudado.

Quando não há informação sobre os processos e eventos, ou a modelagem do fenômeno é restringida pelo não entendimento das leis físicas que o governam, ou ainda, pela carência de um conjunto pertinente de dados, está-se diante das incertezas no conhecimento, as quais também podem ser denominadas de epistêmicas, subjetivas ou internas. A incerteza epistêmica, a qual decorre exclusivamente do fato da informação encontrar-se incompleta, corresponde à causa principal de incerteza na maior parcela dos problemas práticos.

Hachich (1998) classificou as incertezas em intrínsecas, de parâmetros e de modelo. De maneira semelhante, Morgenstern (1995) também se valeu de três grupos para catalogar as incertezas geotécnicas, estabelecidos em incertezas nos parâmetros, nos modelos e humana.

Levando-se em consideração a última classificação acima mencionada, pode-se referir à incerteza de parâmetros, também, como sendo estatística, a qual se relaciona à ausência de acurácia nos valores dos parâmetros para o projeto (Rilho, 2011). Já a incerteza de parâmetros advém, conforme Baecher (1987), da dispersão dos dados e do erro sistemático. Este corresponde ao desvio da tendência medida em relação à tendência verdadeira, a qual não é conhecida, enquanto aquela está relacionada à dispersão em torno da média.

A categorização da dispersão de dados é realizada em variabilidade espacial e erros aleatórios. As variações essenciais que as propriedades do solo possuem em diferentes locais, ainda que sejam referentes a uma área com características homogêneas, relacionam-se à variabilidade espacial. Destacam-se como razões para a definição da variabilidade espacial dos solos, conforme Lacasse & Nadim (1996), as variações sobrevindas da sua história de tensões, composição mineralógica, processos de natureza física e mecânica desenvolvidos na sua decomposição e condições ambientais ocorridas na sua formação ou deposição.

Os erros ditos aleatórios ocorrem por meio de medições executadas em laboratório ou em campo, originários a partir de erros cometidos pelo operador ou por deficiências na aparelhagem que se utilizou para proceder ao estabelecimento da propriedade do solo. Segundo Flores (2008), erros de cunho aleatório não remetem a uma variação real das propriedades do solo, necessitando serem retirados da análise. A incidência desta forma de erro pode ser reduzida por meio da certificação do uso dos procedimentos corretos e que estejam consoantes com as normas internacionais.

Referindo-se aos erros sistemáticos, estes são classificados em erro estatístico e desvio nos procedimentos nas medidas (“bias”). Enquanto a incerteza presente na média estimada derivada do tamanho limitado da amostra define o erro estatístico na tendência média, o desvio nos procedimentos de medida (“bias”) surge no momento em que a propriedade medida é superestimada ou subestimada em todos os pontos do ensaio de forma uniforme.

Costa (2005) afirma que a incerteza de modelo encontra-se vinculada à diferença existente entre a realidade e a teoria, seguida pelos modelos de previsão, sendo que esta se configura como uma das maiores fontes de incertezas na engenharia geotécnica (Morgenstern, 1995; Whitman, 1996). Em relação à incerteza humana, tal vertente é oriunda da falta de conhecimento ou mostra-se procedente da incipiente comunicação entre os responsáveis pelo

projeto (Flores, 2008). Costa (2005) define que estes erros são normalmente imprevisíveis e aleatórios, podendo ser exemplificados pela falta de cuidado, ignorância de normas ou padrões, ou mesmo informações enganosas. Segundo Santos (2008), a alternativa mais eficaz para reduzir este tipo de incerteza corresponde à verificação de projetos por instituições distintas e um eficiente controle de qualidade.

Conforme Flores (2008), os erros que podem ser quantificados são representados pelas incertezas nos parâmetros, ao passo que as incertezas de modelos e humanas são usualmente desconsideradas, uma vez que são de difícil mensuração.

2.2.2 UTILIZAÇÃO DE BASES ESTATÍSTICAS APLICADAS À GEOTECNIA

Solos e rochas possuem propriedades mecânicas e físicas naturalmente dispersas (Flores, 2008). Geralmente, os engenheiros, no intuito de assegurar a segurança frente a um cenário constituído por incertezas, realizam uma ponderação conservadora das propriedades do solo, agregada a um projeto e a processos construtivos adequados (Costa, 2005). Segundo Santos (2008), a segurança em obras civis era, inicialmente, avaliada por intermédio da intuição, sendo que houve, ao longo do tempo, a inclusão de aspectos relacionados à experiência e à segurança, de forma empírica, com base em obras análogas.

Na maioria das vezes, utiliza-se, na tentativa de alcançar uma margem apropriada de segurança contra desvios inesperados relativos ao desempenho aguardado, a incorporação de um fator de segurança de projeto fundamentado em critérios de experiência e julgamento (Costa, 2005). Conforme Assis et al. (2012), o valor pertinente ao fator de segurança, o qual é congregado de maneira arbitrária, relaciona-se ao risco, que é assumido no projeto desde o seu início.

Griffiths et al. (2002) afirmam que a abordagem do fator de segurança, por meio da análise geotécnica tradicional, pode ser utilizada de duas maneiras, fornecendo como exemplos a equação da capacidade de carga de Terzaghi para fundações e a análise de estabilidade de taludes.

No primeiro caso, obtém-se uma estimativa de valor último, a qual, ao ser dividida pelo fator de segurança, resulta nos níveis de carga permitidos para projeto. Para a segunda alternativa, o fator de segurança atua reduzindo a resistência ao cisalhamento do solo antes da realização do cálculo do equilíbrio limite. Griffiths et al. (2002) ainda ressaltam que, para qualquer uma das formas citadas, este fator de segurança corresponde, implicitamente, a um

termo que abrange e insere todas as questões de variabilidade e incertezas intrínsecas à análise geotécnica.

Rilho (2011) ressalta que a engenharia geotécnica agregou os conceitos vinculados à probabilidade e estatística existentes em outros campos da tecnologia apenas a partir da década de 50 do século passado, na tentativa de desenvolver artifícios mais racionais voltados para a inserção da segurança. Segundo Costa (2005), a quantificação das origens ligadas às incertezas geotécnicas e o estudo dos seus efeitos na confiabilidade de um talude foram possíveis pela introdução de métodos estatísticos. Costa (2005) destaca ainda que a abordagem probabilística apresenta, como ideia subjacente, a utilização da teoria da probabilidade na geração de um procedimento com caráter sistemático para o tratamento de incertezas em termos quantitativos.

Conforme Duncan (2000), na prática geotécnica convencional, de forma lógica, os fatores de segurança utilizados se baseiam na experiência, embora seja comum o emprego do mesmo fator de segurança em determinado tipo de aplicação, sem considerar o nível de incerteza envolvido no seu cálculo. Destaca-se, ainda, que, mesmo não sendo lógico, frequentemente, um mesmo valor para o fator de segurança também é usado em condições que envolvam uma ampla gama de graus de incerteza, quer por tradição, quer por normas. Otálvaro & Cordão Neto (2011) apontam que o fator de segurança real pode se modificar no decorrer da vida útil da obra em virtude da interação existente com agentes externos, incluindo aqueles de natureza atmosférica.

No campo da engenharia geotécnica, a análise de estabilidade de taludes tem sempre representado um importante tópico (Ge et al., 2011). Neste caso, o emprego do fator de segurança, proveniente de um método de valor fixo, é comumente utilizado como indicador para a avaliação da estabilidade das encostas (Bishop, 1955). Ge et al. (2011) afirmam que, em decorrência do alto grau de incerteza de certo número de parâmetros do solo, o fator de segurança resultante de métodos de valor fixo falha ao tentar considerar a aleatoriedade e a variabilidade das variáveis de projeto, não refletindo, de fato, o nível de segurança da encosta.

Incertezas advindas de variadas origens podem ser comparadas, analisadas e combinadas através de procedimentos probabilísticos (Costa, 2005). Conforme Ang & Tang (1975), no campo da engenharia civil, e especialmente na área relativa à geotecnia, são bem conhecidas às possibilidades de variações dos parâmetros e carregamentos em torno de seus valores adotados, acarretando um convívio de riscos de rupturas envolvendo estruturas geotécnicas. Desta forma, Hachich (1998) argumenta que o desenvolvimento de novos

critérios de projeto atrelados à segurança encontra-se diretamente ligado ao uso de modelos probabilísticos, uma vez que o cerne desta questão reside nos aspectos de variabilidade e incerteza.

Segundo Santos (2008), a ausência de unicidade encontrada nos métodos deterministas e semi-probabilistas entre coeficiente de segurança e a própria segurança tende a ser corrigida pelos métodos probabilistas. Assim sendo, com relação a certo problema de engenharia, o qual apresenta determinado grau de incerteza, torna-se possível incorporar um nível de confiabilidade ou segurança, permitindo, desta maneira, uma confrontação mais apropriada entre projetos alternativos e pautada no preceito de confiabilidade (Costa, 2005).

Portanto, Assis et al. (2012) atentam que os métodos probabilísticos e estatísticos mostram-se como uma opção sistemática de inserir, no projeto, a variabilidade proveniente de parâmetros e carregamentos, na intenção de computar o risco da ruptura ou a confiabilidade destas estruturas. Compartilhando desta ideia, Hassan & Wolff (2000) citam que muitos modelos probabilísticos podem ser empregados no sentido de analisar a confiabilidade de encostas.

Entretanto, a utilização de métodos probabilísticos não vem mostrando o destaque esperado, sendo que muitos engenheiros não contemplam a necessidade de adoção destas ferramentas aparentemente promissoras (Tang, 1993; Maia, 2003). Na prática, conforme El-Ramly et al. (2005), a aversão por parte dos profissionais de engenharia, em relação à aplicação de métodos estatísticos na estabilidade de taludes, deve-se, entre outros aspectos, à ausência de trabalhos publicados, demonstrando a implementação e os benefícios das análises probabilísticas, bem como a inexistência de um critério probabilístico de projeto consistente.

2.2.3 MÉTODOS PROBABILÍSTICOS

Conforme Assis et al. (2012), os métodos probabilísticos podem ser considerados como aqueles que possibilitam a avaliação da distribuição de probabilidade de uma variável considerada como dependente, em função do conhecimento das distribuições estatísticas das variáveis independentes que determinam esta primeira variável. A avaliação da distribuição estatística de uma função de desempenho é geralmente realizada por intermédio de uma análise numérica probabilística (Suchomel & Mašin, 2009).

Montoya & Assis (2011b) destacam que, na análise de confiabilidade, a utilização de técnicas com esta natureza estatística proporciona o tratamento da incerteza de uma maneira mais rigorosa nos procedimentos e processos de projeto geotécnico.

Reis (2010) destaca que é necessário um conjunto de dados consistente abrangendo os solos constituintes da região analisada para o uso das análises probabilísticas em problemas de estabilidade de taludes, onde estas informações devem ser provenientes de ensaios de laboratórios e trabalhos de campo. Conforme ainda Reis (2010), a representatividade destas análises estará fortemente relacionada ao montante de incertezas inerentes ao problema, que se encontram envolvidas na solução empregada.

O emprego metódico e rigoroso dos métodos de confiabilidade, apesar de não garantir totalmente que não ocorram rupturas, proporciona ao engenheiro a certeza de que os elementos conhecidos e que se tenha controle, não influirão substancialmente nestas rupturas (Whitman, 1984).

A partir de agora, destacam-se três dos principais métodos probabilísticos: Método de Monte Carlo, Método FOSM e Método das Estimativas Pontuais.

2.2.3.1 MÉTODO DE MONTE CARLO

O primeiro método a ser destacado refere-se àquele denominado de Monte Carlo. Montoya & Assis (2011a) afirmam que este método é definido como uma metodologia que possibilita estabelecer a função de distribuição de frequência da variável considerada como dependente, por meio da geração de números aleatórios uniformes que expressam as variáveis independentes envolvidas.

Korver (1994) determina que a importância do método de Monte Carlo imposto à predição de confiabilidade torna-se clara, quando a complexidade do sistema avaliado faz com que a formulação constituída por modelos exatos seja inviável. Apresentando-se como uma das formas para analisar a propagação da incerteza, o método de Monte Carlo tem como vantagem, conforme explicam Moore & Weatherford (2005), estabelecer como o desempenho ou a viabilidade do sistema modelado são afetados pela variação aleatória ou pelo erro.

Foi durante a Segunda Guerra Mundial, em trabalhos envolvendo a criação da bomba atômica, que o método de Monte Carlo foi devidamente reconhecido como elemento de pesquisa (Hammersley & Handscomb, 1964). Foi formalizado pela publicação do artigo “Monte Carlo Method”, por John Von Neumann e Stanislaw Ulam, em 1949 (Sobol, 1994). Os primeiros cálculos a utilizarem este método em larga escala corresponderam a estudos ligados à dispersão e absorção de nêutrons, processos que se caracterizam por serem randômicos, onde é bastante natural o emprego de números aleatórios (James, 1980).

Marinilli & Cerrolaza (1999) descrevem os passos através dos quais se define a distribuição de uma dada resposta $Y = g(x_1, x_2, x_3, \dots, x_n)$ pelo uso do método de Monte Carlo:

- Procede-se a determinação dos valores de cada parâmetro estocástico $x_1, x_2, x_3, \dots, x_n$, ação que pode ser efetivada pela geração de um número aleatório (z_i) uniformemente distribuído no intervalo $[0,1]$. Com isto, o valor de cada parâmetro estocástico (x_i) pode ser definido utilizando-se a expressão $x_i = F_{x_i}^{-1}(z_i)$, onde $F_{x_i}^{-1}(z_i) = P(x \leq z_i)$ é a função de distribuição de probabilidade acumulada para cada um dos parâmetros;
- Aplica-se o modelo e calcula-se a resposta do problema em termos determinísticos a partir dos valores gerados relativos aos parâmetros estocásticos;
- Repetem-se as duas instruções anteriores, até que se alcance uma quantidade representativa, do ponto de vista estatístico, da variável dependente, de forma que o número de simulações requeridas está relacionado aos momentos estatísticos que deverão ser aproximados e do nível de confiança a ser obtido;
- Por fim, processa-se a definição da média, desvio padrão, moda, coeficiente de variação, função de distribuição de probabilidade para a variável dependente.

Segundo Montoya (2013), ao considerar que $G(x)$ refere-se à função de desempenho que estabelece a variável dependente, o conjunto N de valores conseguidos pela aplicação do segundo e terceiro passos descritos acima são tratados como uma amostra de uma população, da qual é possível definir a média ($G(\bar{X})$) e a variância ($V[G(X)]$) a partir das seguintes expressões:

$$G(\bar{X}) = \frac{1}{N} \sum_{i=1}^N G(X)_i \quad (2.7)$$

$$V[G(X)] = \frac{1}{N} \sum_{i=1}^N [G(X)_i - G(\bar{X})]^2 \quad (2.8)$$

onde $G(X)_i$ é cada um dos N valores obtidos para a variável dependente pela aplicação da função de desempenho.

Montoya (2013) destaca ainda o emprego de programas computacionais na obtenção das medidas para as variáveis independentes, possuindo como informações, os seus valores médios, os seus desvios padrões e a sua forma de distribuição.

Assis et al. (2012) propõem um exemplo para o entendimento da metodologia de Monte Carlo, utilizando o cálculo do valor numérico para a integral relacionada a uma função limitada por $0 \leq f(x) \leq c$, $a \leq x \leq b$, conforme indicado pela expressão abaixo:

$$I = \int_a^b f(x).dx \quad (2.9)$$

Na Fig. 2.2 estão representados os respectivos limites da integral, assim como o retângulo de área $(b - a).c$.

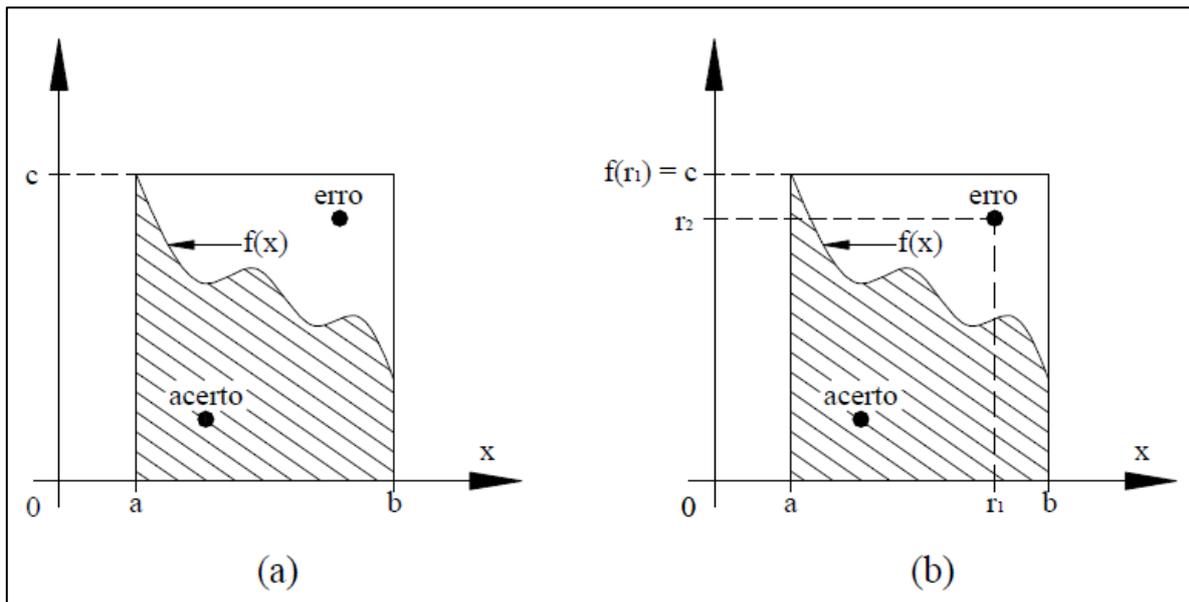


Figura 2.2. Método de Monte Carlo para integração (Maia, 2003).

O valor da integral procurada refere-se à área limitada pela curva $f(x)$ dentro do intervalo $[a,b]$, representada pela área hachurada. Para a estimativa da integral, fez-se uso de uma analogia, baseando-se no arremesso de dardos ou esferas, de forma aleatória, na direção do retângulo acima delimitado. Assim, o valor da integral (I) seria dado por:

$$I \approx c.(b - a). \frac{N_H}{N_H + N_M} \quad (2.10)$$

onde:

N_H representa os dardos que atingiram a região hachurada;

N_M referem-se aos dardos que não atingiram a região.

A razão $N_H/(N_H+N_M)$ pode ser considerada como a probabilidade de acertos. Na analogia, podem-se substituir os dardos por números aleatórios uniformemente distribuídos r_1

e r_2 , sendo que $a \leq r_1 \leq b$ e $0 \leq r_2 \leq c$. Na parte b da Fig. 2.2, a área fora do alvo é representada quando $r_2 > f(r_1)$, enquanto que a condição $r_2 < f(r_1)$ define a região do alvo (área limitada pela curva $f(x)$). A Eq. 2.10 fornece a integral quando o processo acima é repetido para uma extensa quantidade de pares de números aleatórios, sendo esta metodologia denominada de Monte Carlo do “acerto ou erro”.

Um ponto fundamental dentro da metodologia de Monte Carlo está ligado à geração de números aleatórios uniformes, conseguidos por meio de passos determinísticos que envolvem uma lei de recorrência:

$$x_N = A.x_{N-1} \pmod{M} \quad (2.11)$$

onde M e A são números inteiros positivos. A notação $x = y \pmod{M}$ significa que x é o resto da divisão do número y por M . É fixado um valor inicial x_0 (geralmente um valor desconhecido) o qual é, então, multiplicado por A e operado pelo mod M , resultando no novo termo x_1 . A sequência de números aleatórios uniformes (pseudo-aleatórios) é determinada pela aplicação deste processo iterativamente.

Conforme Silva (2006), a maior precisão, quanto aos resultados gerados pelo método de Monte Carlo, está ligada a um número maior de repetições envolvidas, requerendo-se, portanto, muitas iterações para a avaliação da estabilidade de taludes por este artifício. Harr (1987) sugeriu que o número de iterações varia em função da quantidade de variáveis consideradas e do nível de confiança assumido, podendo ser estimada pela equação abaixo:

$$N_{mc} = \left[\frac{(d)^2}{4.(1 - \varepsilon)^2} \right]^m \quad (2.12)$$

onde:

N_{mc} refere-se ao número de iterações de Monte Carlo;

ε é o nível de confiança desejado;

d corresponde ao desvio padrão relativo ao nível de confiança;

m fornece o número de variáveis.

Como são empreendidas, para fins práticos, milhares de iterações, Montoya (2013) chama a atenção para a necessidade, por parte do método, de uma alta velocidade na execução dos cálculos, bem como na necessidade de geração de uma grande quantidade de números aleatórios, aspecto que pode ser superado na maioria das avaliações em estabilidade de taludes de maneira eficiente, dado o nível de desenvolvimento atual em que se encontram os

computadores. Obtém-se uma descrição completa deste método no trabalho de Hammersley & Handscomb (1964).

2.2.3.2 MÉTODO FOSM

O método FOSM (First Order Second Moment), segundo Chen et al. (2007), corresponde a um dos mais utilizados, devido à relativa simplicidade e ao rigor matemático presentes em sua formulação. Este método define a distribuição de probabilidade de uma função de desempenho que possui um número de variáveis aleatórias, pelo uso da série de Taylor, configurando-se como a segunda categoria dos métodos probabilísticos desenvolvidos (Baecher & Christian, 2003).

Conforme Arsyad (2008), a utilização deste método proporciona a aproximação do parâmetro de tendência central, representado pela média, e do parâmetro de dispersão, relativo ao desvio padrão, da variável aleatória. Ang & Tang (1975) destacam que tal abordagem mostra-se eficiente na investigação da propagação das incertezas de segundo momento (variância).

Restrepo (2011) ressalta que a função de desempenho adotada pode corresponder a qualquer relação relevante, sendo que, para fins de análise de estabilidade de taludes, esta deve fazer referência ao cálculo do fator de segurança. Harr (1987) afirma que o ponto básico desta abordagem consiste em expressar a função de desempenho utilizada como uma função de diferentes variáveis aleatórias que estão contidas na análise estatística.

Por meio da expansão da função de desempenho sobre o ponto $x = \bar{x}$, realizada com base na fórmula de Taylor, pode-se apreender, conforme Harr (1987), os fundamentos matemáticos do método FOSM:

$$f(x) = f(\bar{x})(x - \bar{x}) + \frac{f''(\bar{x})}{2!} \cdot (x - \bar{x})^2 + \dots + \frac{f^{(N-1)}(\bar{x})}{(N-1)!} \cdot (x - \bar{x})^{N-1} + R_N \quad (2.13)$$

onde:

$f^{(n)}(\bar{x})$ corresponde à n-ésima derivada, avaliada para $x = \bar{x}$;

R_N refere-se ao resto, o qual pode ser ou não nulo.

Conservando-se apenas termos de 1ª ordem (lineares), para uma função de duas variáveis $F(x,y)$ nos pontos \bar{x} e \bar{y} , a expansão da série de Taylor corresponde à seguinte equação:

$$F(x, y) = F(\bar{x}, \bar{y}) + \frac{\partial F}{\partial x} \cdot (x - \bar{x}) + \frac{\partial F}{\partial y} \cdot (y - \bar{y}) \quad (2.14)$$

De forma que todas as derivadas são estimadas para os pontos $x = \bar{x}$ e $y = \bar{y}$.

Aplicando-se o formulário para distribuições bivariadas, considerando-se \bar{x} e \bar{y} como os respectivos valores esperados das variáveis, obtêm-se as seguintes aproximações:

$$E[F(x, y)] = F(\bar{x}, \bar{y}) \quad (2.15)$$

$$V[F(x, y)] = \left(\frac{\partial F}{\partial x}\right)^2 V(x) + \left(\frac{\partial F}{\partial y}\right)^2 V(y) + 2 \cdot \left(\frac{\partial F}{\partial x}\right) \cdot \left(\frac{\partial F}{\partial y}\right) \cdot \text{cov}(x, y) \quad (2.16)$$

As derivadas também são estimadas pautando-se nos valores esperados das variáveis.

Ao conservar somente os termos lineares da série de Taylor, para o caso de n variáveis aleatórias não correlacionadas, a Eq. 2.16 se transforma em:

$$V[F] = \sum_{i=1}^n \left(\frac{\partial F_i}{\partial x_i}\right)^2 V[x_i] \quad (2.17)$$

onde:

$V[F]$ corresponde à variância de F ;

∂F_i relaciona-se à variação processada na função F , a qual sucede quando se varia ∂x_i para cada um dos n parâmetros x_i ;

∂x_i refere-se à taxa de variação das variáveis envolvidas no estudo, consideradas com características aleatórias;

$V[x_i]$ é a variância de cada uma das variáveis x_i .

O desvio padrão da função F avaliada é dado pela extração da raiz quadrada da sua variância $V[F]$.

Com todos os dados médios de cada variável aleatória, obtém-se o valor médio ou esperado da função:

$$E[F] = F(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N) \quad (2.18)$$

Quando se emprega n variáveis independentes, são exigidas pelo menos $n+1$ análises:

- Uma para o cálculo dos valores médios;
- n análises para a definição das derivadas $(\partial F/\partial x_i)$ para cada variável independente.

Em termos da função de desempenho FS , que define o fator de segurança, as equações anteriores ficam da seguinte forma:

$$V[FS] = \sum_{i=1}^n \left(\frac{\partial FS_i}{\partial x_i} \right)^2 V[x_i] \quad (2.19)$$

$$E[FS] = FS(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N) \quad (2.20)$$

Chama-se a atenção que, assim como para a função F, o desvio padrão da distribuição de FS será encontrado pelo cálculo da raiz quadrada de V[FS]. Miranda (2005) destaca que a aplicação do método está atrelada à necessidade de definir as derivadas parciais da função do fator de segurança em relação às variáveis independentes aleatórias, de forma que isto pode agir no sentido de restringir o uso desta técnica. Conforme Suchomel & Mašin (2009), como uma primeira opção, o modelo pode ser derivado analiticamente, embora, em muitos casos práticos, a execução destes cálculos se torne inconveniente ou mesmo impossível. Desta forma, as derivadas parciais podem ser aproximadas numericamente por meio de diferenças finitas (Christian et al., 1994), sendo que a variação imposta a cada parâmetro deve ser igual ao valor do respectivo desvio padrão σ_{xi} (Duncan, 2000). Considerando-se $E(x_i)$ o valor médio da variável x_i , a aproximação das derivadas parciais por meio de diferenças finitas centrais pode ser dada por:

$$\frac{\partial FS}{\partial x_i} = \frac{FS[E(x_i) + \sigma_{xi}] - FS[E(x_i) - \sigma_{xi}]}{2 \cdot \sigma_{xi}} \quad (2.21)$$

Pode-se discernir melhor o comportamento não linear da função quando se analisa a derivada em um intervalo que compreende os valores de $\pm\sigma_{xi}$, embora pequenos incrementos nos parâmetros x_i (Δx_i) possibilitem avaliar mais precisamente as derivadas em um ponto, segundo encontra-se representado na equação abaixo, a qual usa a diferença central de dois pontos para aproximar numericamente a derivada da função por diferenças finitas:

$$\frac{\partial FS}{\partial x_i} = \frac{FS[E(x_i) + \Delta x_i] - FS[E(x_i) - \Delta x_i]}{2 \cdot \Delta x_i} \quad (2.22)$$

Conforme mostrado na expressão seguinte, Sandroni & Sayão (1993) estabeleceram que o cálculo das derivadas do fator de segurança pode ser obtido admitindo-se um incremento de 10% nas variáveis de entrada:

$$\frac{\partial FS}{\partial x_i} = \frac{FS[E(x_i) + 0,1x_i] - FS[E(x_i)]}{0,1x_i} \quad (2.23)$$

Farias & Assis (1998) também determinam que incrementos da ordem de 10% dos valores médios das variáveis independentes se mostram satisfatórios.

Conforme Griffiths et al. (2002), como o método emprega o truncamento da série de Taylor após os termos de primeira ordem, valendo-se somente da primeira derivada (linear), é dito de primeira ordem. Por utilizar os dois primeiros momentos das variáveis randômicas na expansão modificada para obter os valores dos dois primeiros momentos da variável dependente, denomina-se segundo momento.

Assis et al. (2012) chamam a atenção para o emprego do Método do Índice de Confiabilidade como uma aplicação direta do método FOSM, sendo muito utilizado na área da engenharia geotécnica, principalmente na análise estatística do fator de segurança de taludes e satisfazendo os tópicos abaixo:

- Determinação da variância total do fator de segurança a partir do emprego do método FOSM. O processo corresponde à consequente definição das parcelas de variâncias do FS, advindas de cada parâmetro considerado como probabilístico no desenvolvimento do cômputo. Ao se analisar estas referidas parcelas, a importância relativa de cada variável pode ser determinada;
- Cálculo do índice de confiabilidade do fator de segurança (β_i), definido pela seguinte expressão:

$$\beta_i = \frac{E[FS] - FS_{crit}}{\sigma_{FS}} \quad (2.24)$$

Onde FS_{crit} refere-se ao valor do fator de segurança crítico relacionado à estabilidade do talude (em geral, igual a 1), enquanto σ_{FS} representa o desvio padrão da distribuição de probabilidade conseguida para FS.

Babu (2008) citado por Suchomel & Mašín (2009) atenta que, caso as funções de distribuição das variáveis independentes que possuem características probabilísticas não puderem ser aproximadas por uma distribuição gaussiana e se a função de desempenho for altamente não linear, o método FOSM torna-se impreciso. Segundo Flores (2008), quanto maior for o grau de não linearidade da função, menor será a precisão fornecida pelo método. Conforme ainda Flores (2008), para problemas de estabilidade de taludes, apesar da condição

de linearidade da função de desempenho não ser atendida, esta é admitida como uma aproximação aceitável.

2.2.3.3 MÉTODO DAS ESTIMATIVAS PONTUAIS

Por fim, estabelecem-se as principais características para o método das estimativas pontuais. Segundo Miller et al. (2004), esta abordagem, concebida por Rosenblueth (1975), fornece um procedimento computacional direto empregado na definição das estimativas dos momentos de uma variável aleatória, quando esta pode ser representada em uma equação como resultado de uma operação matemática de outras variáveis randômicas. Apresenta-se como uma alternativa para considerar a natureza aleatória de certas variáveis, não necessitando como input, assim como o método FOSM, da forma particular das funções densidade de probabilidade (Griffiths et al., 2002). É uma técnica simples, fácil de usar e dispensa pouco conhecimento acerca da teoria da probabilidade, transformando-se, conforme indicado por Christian & Baecher (1999), em um elemento de destaque nas análises de confiabilidade geotécnica. O método se baseia no princípio de que se pode estabelecer uma aproximação da distribuição para a variável dependente, por meio do cálculo das suas soluções em vários pontos estimados e combiná-los com o emprego de uma ponderação apropriada, no sentido de fornecer uma aproximação para a distribuição da solução (Valley & Duff, 2011). A Fig. 2.3 ilustra este procedimento.

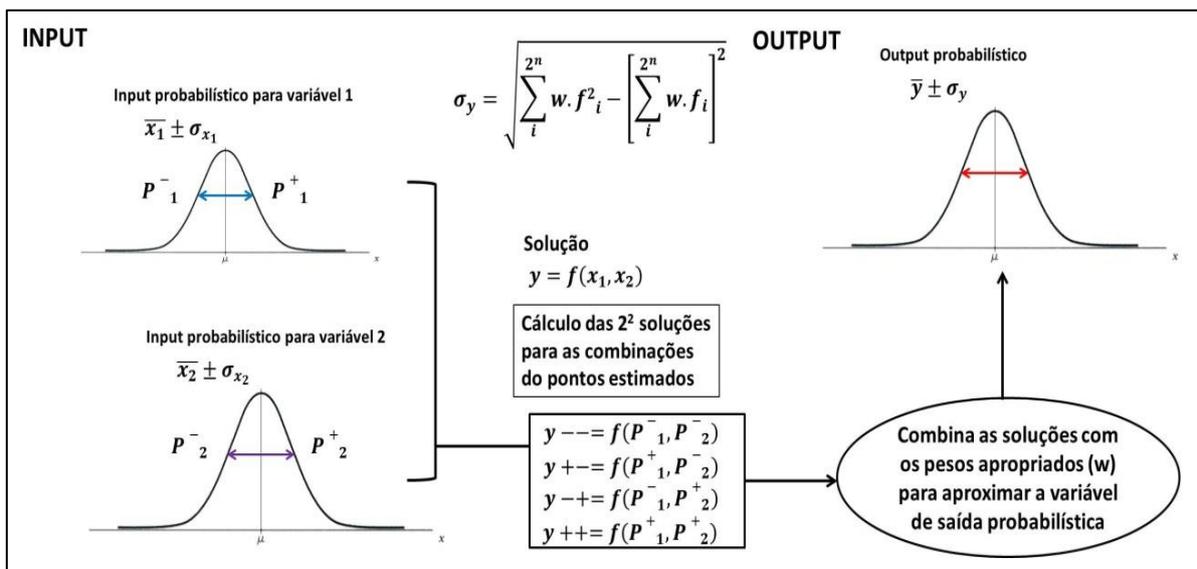


Figura 2.3. Esquema para a aproximação da variável de saída probabilística conseguida pelo método das estimativas pontuais, considerando-se duas variáveis de entrada randômicas (modificado de Valley & Duff, 2011).

Baseando-se no trabalho original de Rosenblueth (1975), o método pode ser formulado pela consideração dos seguintes três casos, destacados, também, por Christian & Baecher (1999). Em muitas considerações, os cálculos são efetivados com base em dois pontos e emprega-se a mesma notação explicitada abaixo:

$$E[Y^m] \approx P_+ \cdot y_+^m + P_- \cdot y_-^m \quad (2.25)$$

onde:

Y refere-se à função determinística de X, de referência, onde $Y = F(X)$;

$E[Y^m]$ estabelece o valor esperado da função Y elevado à potência m;

m determina a ordem do momento estatístico a ser calculado;

y_+ é o valor da função $F(X)$ avaliada em um ponto x_+ , superior que a média μ_x ;

y_- está ligado ao valor de $F(X)$ avaliada em um ponto x_- menor que a média μ_x ;

P_+ e P_- correspondem aos pesos relativos de cada termo.

O objetivo da dedução fundamenta-se em definir valores apropriados para x_+ , x_- , P_+ e P_- .

Tratando-se do primeiro caso, para $F(X)$ como função de uma variável X, Rosenblueth (1975) estabelece que quatro condições devam ser satisfeitas no sentido de que os momentos de ordem inferior relacionados à X possam ser obtidos, no caso em que sua média, variância e coeficiente de simetria sejam conhecidos com precisão:

$$P_+ + P_- = 1 \quad (2.26)$$

$$P_+ \cdot x_+ + P_- \cdot x_- = \mu_x \quad (2.27)$$

$$P_+ \cdot (x_+ - \mu_x)^2 + P_- \cdot (x_- - \mu_x)^2 = \sigma_x^2 \quad (2.28)$$

$$P_+ \cdot (x_+ - \mu_x)^3 + P_- \cdot (x_- - \mu_x)^3 = \nu_x \cdot \sigma_x^3 \quad (2.29)$$

onde:

μ_x é a média da distribuição;

σ_x refere-se ao desvio padrão de X;

ν_x corresponde ao coeficiente de assimetria.

A partir das Eq. 2.28 e 2.29, as seguintes expressões podem ser definidas (Rosenblueth, 1981):

$$x_+ = \mu_x + \left[\frac{v_x}{2} + \sqrt{1 + \left(\frac{v_x}{2} \right)^2} \right] \cdot \sigma_x \quad (2.30)$$

$$x_- = \mu_x + \left[\frac{v_x}{2} - \sqrt{1 + \left(\frac{v_x}{2} \right)^2} \right] \cdot \sigma_x \quad (2.31)$$

$$P_+ = \frac{1}{2} \cdot \left[1 - \frac{v_x}{2} \cdot \frac{1}{\sqrt{1 + \left(\frac{v_x}{2} \right)^2}} \right] \quad (2.32)$$

$$P_- = 1 - P_+ \quad (2.33)$$

Quando a distribuição é simétrica, tem-se:

$$P_- = P_+ = \frac{1}{2} \quad (2.34)$$

$$x_+ = \mu_x + \sigma_x \quad (2.35)$$

$$x_- = \mu_x - \sigma_x \quad (2.36)$$

Para a situação em que a assimetria é insignificante e a distribuição de X é aproximadamente normal, pode-se estimar a variável x em mais de dois pontos. Para uma estimativa de três pontos, por exemplo, as concentrações seriam P_+ em x_+ , P em μ_x e P em x_- . Seguem as seguintes equações:

$$2P_+ + P = 1 \quad (2.37)$$

$$2P_+(x_+ - \mu_x)^2 = \sigma_x^2 \quad (2.38)$$

$$2P_+(x_- - \mu_x)^4 = 3\sigma_x^4 \quad (2.39)$$

A Eq. 2.39 se origina em virtude da medida $3\sigma_x^4$ ser igual ao quarto valor esperado de uma distribuição normal. As soluções das equações acima podem ser definidas como segue:

$$P = \frac{2}{3} \quad (2.40)$$

$$P_- = P_+ = \frac{1}{6} \quad (2.41)$$

$$x_{\pm} = \mu_x \pm \frac{\sigma_x}{3} \quad (2.42)$$

Apoiando-se na analogia ao primeiro caso, tem-se:

$$E[Y^m] \approx P_-(y_-)^m + P_+(y_{\mu})^m + P_+(y_+)^m \quad (2.43)$$

onde y_{μ} refere-se ao valor de Y avaliado em μ_x .

O terceiro caso abordado representa Y como uma função de n variáveis com assimetria nula, podendo ser correlacionadas, resultando da generalização do procedimento conferido ao primeiro caso. A Fig. 2.4 estabelece as localizações dos pontos com seus respectivos pesos para esta situação.

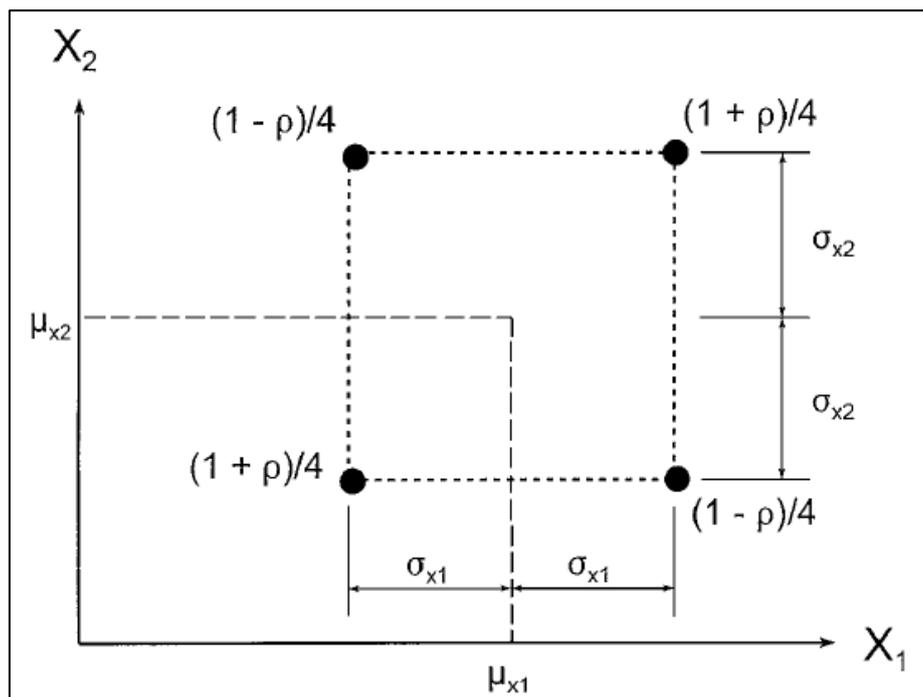


Figura 2.4. Representação dos pontos e pesos para duas variáveis correlacionadas ou não correlacionadas (Christian & Baecher (1999)).

Neste enfoque, o valor de cada variável é avaliado em um desvio padrão acima e um desvio padrão abaixo do seu valor médio, de forma que, para n variáveis, definem-se 2^n pontos selecionados que incluem todas as possíveis combinações.

Como exemplo, para o uso de duas variáveis X_1 e X_2 , os quatro pontos avaliados seriam $(\mu_{x_1} + \sigma_{x_1}, \mu_{x_2} + \sigma_{x_2})$, $(\mu_{x_1} + \sigma_{x_1}, \mu_{x_2} - \sigma_{x_2})$, $(\mu_{x_1} - \sigma_{x_1}, \mu_{x_2} + \sigma_{x_2})$ e $(\mu_{x_1} - \sigma_{x_1}, \mu_{x_2} - \sigma_{x_2})$. O peso de cada um dos quatro pontos avaliados, quando as variáveis são não correlacionadas, é igual a 0,25.

Este valor se altera, caso exista correlação entre as variáveis. Sendo ρ o coeficiente de correlação entre X_1 e X_2 , os pesos seriam, nos dois pontos $(\mu_{x_1} + \sigma_{x_1}, \mu_{x_2} + \sigma_{x_2})$ e $(\mu_{x_1} - \sigma_{x_1}, \mu_{x_2} - \sigma_{x_2})$, iguais a $(1 + \rho)/4$. Para os outros dois pontos, os pesos seriam $(1 - \rho)/4$.

Miller et al. (2004) explicitam os passos a serem seguidos para a aplicação do método das estimativas pontuais, considerando duas variáveis independentes randômicas X_1 e X_2 em uma função de desempenho que fornece o fator de segurança das encostas no intuito de obter a sua média e o seu desvio padrão:

1. Calcular, de forma determinística, o fator de segurança, avaliando a função de desempenho para as combinações possíveis de cada uma das variáveis, conforme equações abaixo:

$$FS_{++} = FS[(\mu_{x_1} + \sigma_{x_1}), (\mu_{x_2} + \sigma_{x_2})] \quad (2.44)$$

$$FS_{--} = FS[(\mu_{x_1} - \sigma_{x_1}), (\mu_{x_2} - \sigma_{x_2})] \quad (2.45)$$

$$FS_{+-} = FS[(\mu_{x_1} + \sigma_{x_1}), (\mu_{x_2} - \sigma_{x_2})] \quad (2.46)$$

$$FS_{-+} = FS[(\mu_{x_1} - \sigma_{x_1}), (\mu_{x_2} + \sigma_{x_2})] \quad (2.47)$$

2. Calculam-se os pesos para os pontos utilizados:

$$P_{++} = P_{--} = \frac{1}{4}(1 + \rho_{12}) \quad (2.48)$$

$$P_{+-} = P_{-+} = \frac{1}{4}(1 - \rho_{12}) \quad (2.49)$$

Onde FS é o valor do fator de segurança obtido mediante a função $F(X)$ e ρ_{12} refere-se ao coeficiente de correlação entre as variáveis de entrada X_1 e X_2 .

3. Cômputo da expectância (média μ_x) de FS:

$$E[FS] = P_{++}.FS_{++} + P_{--}.FS_{--} + P_{+-}.FS_{+-} + P_{-+}.FS_{-+} \quad (2.50)$$

4. Cálculo da variância ($V[FS]$) de FS:

$$V[FS] = E[FS^2] - (E[FS])^2 \quad (2.51)$$

onde $E[FS^2]$ é determinado substituindo o termo FS por FS^2 na Eq. 2.50.

5. O desvio padrão de FS é obtido pelo cálculo da raiz quadrada de $V[FS]$.

Assis et al. (2012) estabelecem que, com a generalização para a situação em que a função de desempenho é dependente de n variáveis aleatórias não correlacionadas entre si, as estimativas para a média e a variância são dadas pelas seguintes equações:

$$E[FS] = \frac{1}{2^n} \sum_{i=1}^{2^n} FS_i \quad (2.52)$$

$$V[FS] = \frac{1}{2^n} \sum_{i=1}^{2^n} [FS_i - E(FS)]^2 \quad (2.53)$$

Na Eq. 2.52 e na Eq. 2.53, obtém-se as medidas FS_i pela substituição, alternadamente, dos valores das variáveis independentes por $\mu_j \pm \sigma_j$, com $j = 1, 2, \dots, n$ na função que determina a dependência entre FS e estas variáveis.

A Fig. 2.5 define o número de combinações possíveis em função da quantidade de variáveis aleatórias envolvidas na análise.

		Número de Variáveis Aleatórias					
		1	2	3	4	...	n
N	ú	1	-	-	-	-	
		2	+	-	-	-	
m	e	3	-	+	-	-	
		4	+	+	-	-	
o	r	5	-	-	+	-	
		6	+	-	+	-	
		7	-	+	+	-	
d	e	8	+	+	+	-	
		9	-	-	-	+	
		10	+	-	-	+	
		11	-	+	-	+	
T	e	12	+	+	-	+	
		13	-	-	+	+	
		14	+	-	+	+	
		15	-	+	+	+	
s	o	16	+	+	+	+	
		.					
		.					
		.					
		2^n					

Figura 2.5. Combinação (2^n) dos pontos particulares $X_{i+} = \mu_i + \sigma_i$ e $X_{i-} = \mu_i - \sigma_i$ (Maia, 2003).

2.3 RISCO DEVIDO A MOVIMENTOS DE MASSA

Aqui serão enfocados pontos importantes relacionados ao risco decorrente de movimentos de massa, como os conceitos fundamentais utilizados neste tema, ressaltando os fundamentos e ações necessárias ao zoneamento associado a deslizamentos.

2.3.1 CONCEITOS FUNDAMENTAIS E AVALIAÇÃO DO RISCO ASSOCIADO A DESLIZAMENTOS

Perdas e danos podem ou não ser provocados, em decorrência de um determinado fenômeno geológico (Cerri & Amaral, 1998). Um fenômeno geológico é denominado acidente quando determina consequências sociais e econômicas. Quando não há registros de perdas e danos associados diretamente ao acontecimento, este é classificado como um evento.

Pela NBR ISO 31000 (ABNT, 2009), risco é definido como o efeito da incerteza nos objetivos, sendo que um efeito é um desvio em relação ao esperado (positivo e/ou negativo). Ainda de acordo com a ABNT (2009), o risco pode ser caracterizado pela referência aos eventos potenciais e às consequências, ou uma combinação destes, sendo muitas vezes expresso em termos de uma combinação de consequências de um evento (incluindo mudanças nas circunstâncias) e a probabilidade de ocorrência associada. Nesse contexto, pode-se entender o termo evento como sendo a ocorrência ou mudança em um conjunto específico de circunstâncias, enquanto consequência é o resultado de um evento que afeta os objetivos. Já probabilidade é a chance de algo acontecer.

Conforme Fell et al. (2005), um glossário de termos referentes à avaliação de risco foi desenvolvido pelo Comitê Técnico em Gerenciamento e Avaliação de Riscos da International Society of Soil Mechanics and Geotechnical Engineering (ISSMGE), de maneira que a utilização da nomenclatura ali apresentada foi encorajada, no sentido de promover uma uniformidade quanto ao emprego destes termos pela comunidade internacional.

Fell et al. (2008a) ressaltam que estas definições devem ser usadas para todo zoneamento, registros e documentos de planejamento de uso do solo, recomendando-se que os conceitos estejam vinculados a estes materiais, na tentativa de evitar interpretações equivocadas. Os principais termos presentes no documento citado, bem como as suas respectivas definições, encontram-se explicitados abaixo.

- Deslizamento: Movimento de uma massa de rocha, detritos ou solo ao longo de uma encosta. O Manual Internacional para Zoneamento de Susceptibilidade, Ameaça e

Risco decorrentes de Deslizamentos destaca que se deve utilizar uma terminologia consistente para classificar e descrever os deslizamentos como a proposta por Cruden & Varnes (1996). Outras classificações são recomendadas como aquelas atribuídas a Varnes (1978) ou Hutchinson (1988), em paralelo com as terminologias descritas pela IAEG (1990).

- Deslizamento ativo: Um deslizamento que está se movendo atualmente, quer seja pela primeira vez, quer seja por uma reativação;
- Deslizamento reativado: Um deslizamento que se apresenta novamente como ativo, após permanecer um período sem atividade;
- Inventário de deslizamentos: Um inventário de localização, classificação, volume, atividade, data de ocorrência e outras características dos deslizamentos em uma área;
- Susceptibilidade de deslizamentos: Uma avaliação quantitativa ou qualitativa da classificação, volume (ou área) e distribuição espacial de deslizamentos que existem ou podem ocorrer potencialmente em uma área. Uma descrição da velocidade e da intensidade dos deslizamentos reais ou potenciais também pode ser incluída na susceptibilidade;
- Ameaça: Refere-se a uma condição com o potencial para causar uma consequência indesejável. Na descrição da ameaça de deslizamento deve-se considerar a localização, volume (ou área), classificação e velocidade dos deslizamentos potenciais, juntamente com a probabilidade de sua ocorrência dentro de um dado período de tempo;
- Elementos em risco: População, edifícios, obras de engenharia, atividades econômicas, serviços de utilidades públicas e outras infraestruturas em áreas potencialmente afetadas por ameaça de deslizamentos;
- Vulnerabilidade: Grau de perda de um determinado elemento ou conjunto de elementos dentro de uma área afetada por deslizamentos. É expressa em uma escala que varia de 0 (representando ausência de perda) a 1 (representando perda total). Conforme Li et al. (2010), deve-se fazer a análise da interação entre uma dada intensidade de deslizamento e os correspondentes elementos afetados para se obter a estimativa da vulnerabilidade. Neste contexto, pode-se citar, por exemplo, a introdução de matrizes de danos (estrutural, corporal e funcional) por Leone et al. (1996), para ameaça de deslizamento, no sentido de avaliar a vulnerabilidade de edificações, estradas, pessoas, dentre outros elementos expostos;

- **Zoneamento:** Divisão da terra em áreas ou domínios homogêneos, juntamente com a sua classificação mediante o grau de susceptibilidade, ameaça ou risco decorrente de deslizamento atual ou potencial, ou a aplicação de certos regulamentos relacionados à ameaça.

Conforme Fell & Hartford (1997), a prevenção ou redução de casos referentes a movimentos de massa depende de processos sistemáticos e rigorosos que gerenciem ou estabilizem as encostas. Portanto, necessita-se do desenvolvimento de metodologias novas e mais eficientes, que permitam um maior entendimento acerca do risco dos movimentos de massa, corroborando, ainda, para a tomada de decisões racionais quanto à alocação de recursos relacionados ao risco dos deslizamentos (Guzzetti, 2000; Sterlacchini et al., 2007).

Dentro da estrutura de gerenciamento de risco proposta por Fell et al. (2005), destaca-se a fase relacionada à análise de risco, representada na Fig. 2.6.

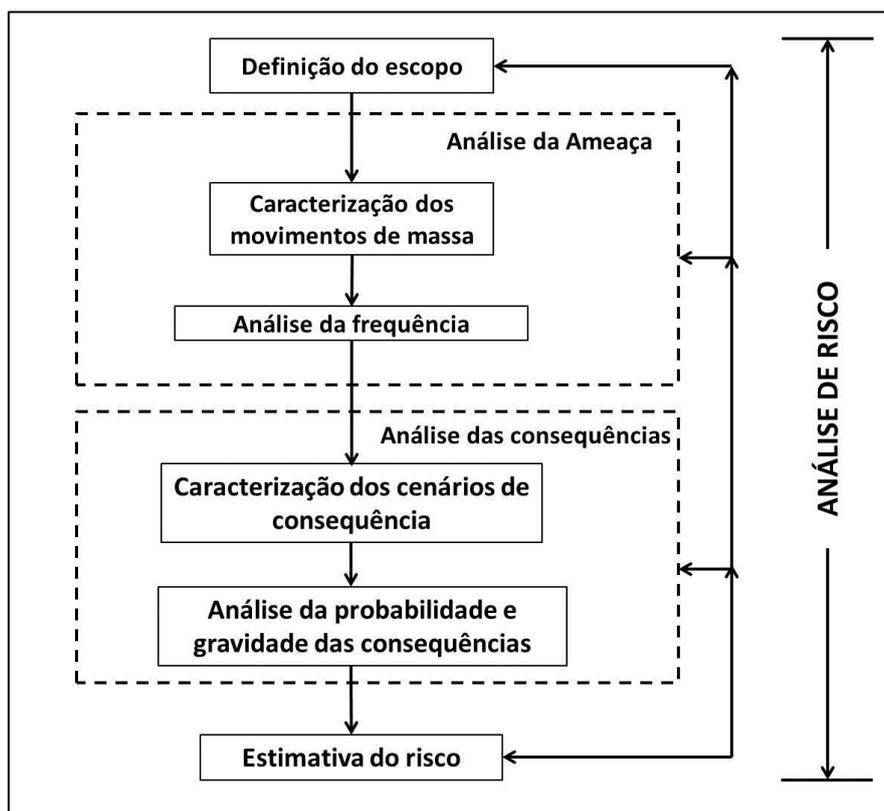


Figura 2.6. Processo de análise de risco devido a deslizamentos (modificado de Fell et al., 2005).

Nesta etapa, Fell et al. (2005) determinam que a análise da ameaça envolva a caracterização dos deslizamentos, composta pela classificação dos eventos, tamanho, velocidade, mecanismos, localização e distância percorrida, assim como a correspondente

frequência de ocorrência, podendo ser dada, por exemplo, pela probabilidade anual. Segundo Anbalagan & Singh (1996), a ameaça refere-se à probabilidade de ocorrência de um fenômeno natural, tal como rastejo, quedas de blocos ou escorregamentos. A união entre a análise das consequências e a análise da ameaça resulta, como produto, na determinação da análise do risco.

A análise das consequências inclui os seguintes itens:

- Determinação e quantificação dos elementos em risco (propriedades e pessoas);
- Probabilidade espaço-temporal;
- Vulnerabilidade, dada pela probabilidade de danos aos elementos em risco, ou pela probabilidade condicional de danos às propriedades, ou pela probabilidade de perdas de vida ou associada a ferimentos.

A proposta de avaliação de risco desenvolvida por Einstein (2003) se traduz em uma etapa inicial de coleta de informações, seguida da etapa determinística, onde há a definição das variáveis mais significativas, seguida de um passo probabilístico, no qual se avalia o risco. Ainda são consideradas as etapas de tomada de decisões ou de atualização de informações envolvidas neste processo. O esquema para esta estrutura está representado pela Fig. 2.7.

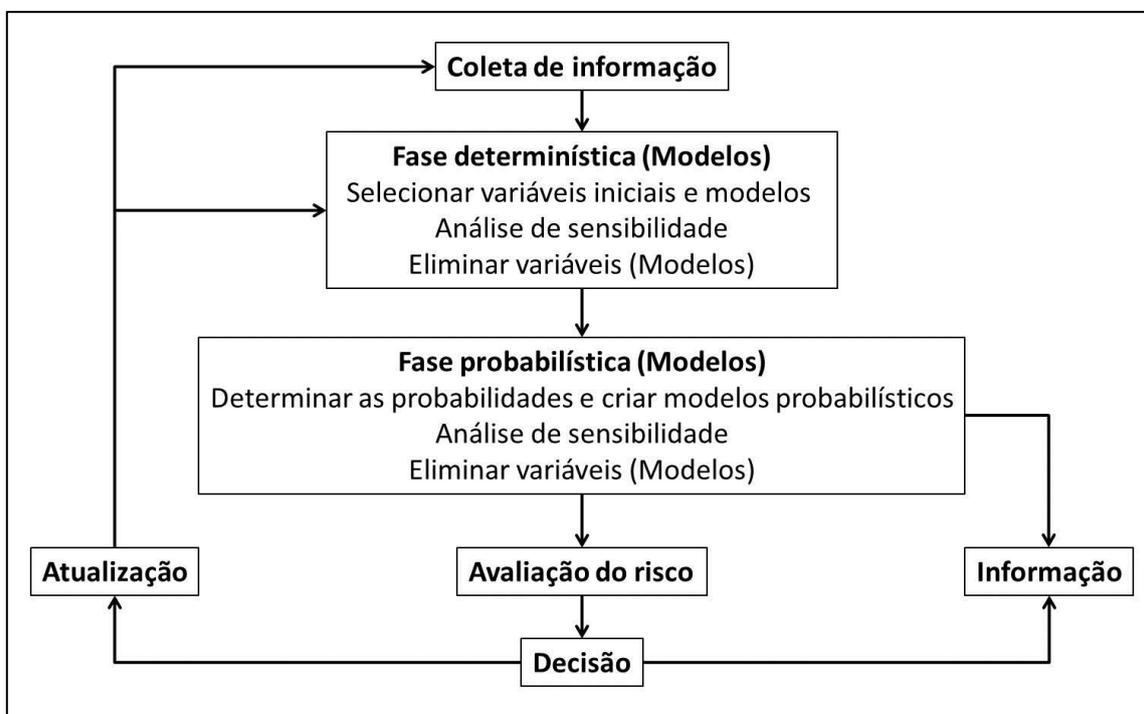


Figura 2.7. Ciclo de análise de decisão em função do risco (modificado de Einstein, 2003).

Assim sendo, Saldivar-Sali & Einstein (2007) afirmam que sistemas voltados para estimativa da ameaça e do risco dos deslizamentos são considerados como instrumentos essenciais para o planejamento urbano e de uso do solo. Corroborando com esta ideia, Akgun et al. (2012) determinam que, nas estratégias do planejamento urbano, deve estar inserida a avaliação do risco dos movimentos de massa, uma vez que estes eventos afetam, de forma adversa, áreas de assentamento.

2.3.2 ZONEAMENTO DE DESLIZAMENTOS

2.3.2.1 DESCRIÇÃO DAS FORMAS DE ZONEAMENTO

Conforme Cascini (2008), o Manual Internacional para Zoneamento de Susceptibilidade, Ameaça e Risco decorrentes de Deslizamentos sugere três formas para mapeamento ou zoneamento de deslizamentos.

A primeira forma associa-se ao zoneamento da susceptibilidade de deslizamento, a qual leva em consideração a classificação, volume (magnitude) ou área e distribuição espacial dos deslizamentos existentes ou potenciais pertencentes à área de estudo, podendo-se incluir também uma descrição da distância percorrida, da velocidade e intensidade dos acidentes. Este tipo de zoneamento geralmente envolve o desenvolvimento de um inventário de eventos de deslizamentos que se processaram no passado em conjunto com uma avaliação das áreas que possuem um potencial a experimentar deslizamentos no futuro, mas sem uma avaliação da frequência (probabilidade anual) de ocorrência de deslizamentos.

O manual ressalta ainda que, em certos casos, o zoneamento da susceptibilidade deverá ser estendido para fora da área de estudo que está sendo zoneada em relação aos aspectos de ameaça e risco, no sentido de abranger áreas que os deslizamentos podem percorrer, com a possibilidade de regressar ainda para a área de estudo. Geralmente, esta medida será necessária para avaliar a propensão das rupturas de taludes e áreas nas quais deslizamentos, a partir de sua fonte, podem percorrer ou regressar.

Considerando os resultados provenientes do mapeamento da susceptibilidade de deslizamentos, o zoneamento de ameaça de deslizamentos atribui uma frequência estimada (probabilidade anual) para os deslizamentos potenciais. Devem-se considerar também todos os deslizamentos que podem afetar a área de estudo, incluindo aqueles que estão acima deste limite, mas que são passíveis de percorrer até a área considerada. De forma similar, aqueles

deslizamentos que se encontram abaixo da área de estudo e que apresentam a possibilidade de uma ruptura regressiva, recaindo no seu interior, também devem ser incorporados na análise.

A ameaça pode ser expressa como a frequência de um particular tipo de deslizamento de certo volume, ou deslizamentos de um tipo, volume e velocidade particular, sendo que a velocidade pode variar com a distância a partir da fonte de deslizamento. Em alguns casos, a ameaça é representada pela frequência de deslizamentos com uma particular intensidade, onde esta última pode ser medida em termos de energia cinética, apresentando-se com maior utilidade para quedas de blocos e fluxo de detritos.

No documento relativo aos comentários sobre o Manual Internacional para Zoneamento de Susceptibilidade, Ameaça e Risco decorrentes de Deslizamentos (Fell et al., 2008b), ressalta-se que o zoneamento da ameaça deve ser realizado para uma determinada área em relação às condições que esta apresenta no período do estudo de zoneamento, de maneira que os efeitos dos desenvolvimentos existentes devem ser considerados na probabilidade de deslizamento.

O zoneamento do risco dos deslizamentos corresponde à terceira forma de zoneamento, a qual toma os resultados do mapeamento da ameaça e faz uma avaliação dos danos potenciais a pessoas (probabilidade anual de perdas de vidas), a propriedades (valor anual de perda de propriedades) e características ambientais (valor anual de perda) para os elementos em risco, levando em consideração a probabilidade espacial e temporal e a vulnerabilidade. Segundo Fell et al. (2008b), para novos desenvolvimentos de mapeamento, uma avaliação destes fatores terá que ser realizada.

Conforme Fell et al. (2008a), para os diferentes tipos de deslizamentos que afetam uma área, é preciso avaliar separadamente a susceptibilidade, a ameaça e o risco, de forma que os resultados devem ser representados em mapas de zoneamento específicos. Isto se torna necessário, já que as medidas e recomendações quanto à mitigação do risco podem diferir para cada tipo de ocorrência de deslizamento.

2.3.2.2 ESCALAS DE MAPAS PARA ZONEAMENTO DE DESLIZAMENTOS

Em se tratando das escalas utilizadas nos mapas referentes ao zoneamento de ameaça e susceptibilidade, Cascini (2008) destaca que estes devem ser preparados em uma escala apropriada para exibir a informação necessária a um nível de planejamento particular.

Somando-se a isto, os dados de entrada usados na confecção dos mapas de zoneamento de deslizamentos devem apresentar resolução e qualidade adequada. Conforme

Fell et al. (2008b), as entradas devem estar em escalas maiores do que aquelas apresentadas pelo mapa de zoneamento.

Segundo, ainda, os comentários sobre o Manual Internacional para Zoneamento de Susceptibilidade, Ameaça e Risco decorrentes de Deslizamentos, dados de entrada mais detalhados, em conjunto com o entendimento dos processos que se desenvolvem na encosta, devem acompanhar o uso de mapas de zoneamento em escalas maiores.

Destaca-se que os valores de frequência de deslizamentos e sua interpretação são afetados pela escala de análise de zoneamento de ameaça de deslizamento (Corominas & Moya, 2008).

Tomando como base os trabalhos de Soeters & Van Westen (1996) e Cascini et al. (2005), pode-se sugerir escalas para o zoneamento de deslizamentos e algumas de suas aplicações. Pequenas escalas, aquelas consideradas menores que 1:100.000, podem ser utilizadas para inventários de deslizamentos e zoneamento de susceptibilidade, em uma área típica maior que 10.000 km² para fornecer informações a tomadores de decisões e ao público em geral.

Abrangendo uma área típica de zoneamento que varia de 1.000 km² a 10.000 km², a adoção de uma escala média (referindo-se ao intervalo de 1:100.000 a 1:25.000) deve ser destinada à criação de inventários de deslizamentos e zoneamento de susceptibilidade para desenvolvimento regional ou projetos de engenharia em grandes escalas. Além disso, pode ser utilizada para a elaboração do nível preliminar de zoneamento de ameaça para áreas locais.

A variação indicativa para grandes escalas está situada entre 1:25.000 e 1:5.000, com uma área típica de zoneamento pertencente ao intervalo de 10 km² a 1.000 km². Como exemplos de aplicações da utilização de grandes escalas, citam-se a elaboração de inventários de deslizamento e o zoneamento de susceptibilidade e ameaça para áreas locais, zoneamento de ameaça do nível intermediário ao avançado para desenvolvimento regional, zoneamento do risco do nível preliminar ao intermediário para áreas locais e estágios avançados para o planejamento de grandes estruturas de engenharia, estradas e ferrovias.

Em uma escala detalhada, pode-se executar o zoneamento de risco e ameaça para nível intermediário e avançado em sítios específicos e áreas locais. É aplicável, também, em fases de projeto de grandes estruturas de engenharia, rodovias e ferrovias. Estas escalas são maiores que 1:5.000 e se referem a uma área típica de zoneamento que abrange desde vários hectares a dezenas de quilômetros quadrados.

2.3.3 DADOS ESPACIAIS UTILIZADOS NA AVALIAÇÃO DE SUSCEPTIBILIDADE, AMEAÇA E RISCO DECORRENTES DE DESLIZAMENTOS

Conforme Van Westen et al. (2005), os dados de entrada fundamentais requeridos para proceder uma avaliação de susceptibilidade, ameaça ou risco advindos de deslizamentos são compostos pelos fatores ambientais e detonantes, ocorrências históricas de deslizamentos e elementos em risco. A Fig. 2.8 retrata a avaliação de risco devido a deslizamentos conforme Van Westen et al. (2008), estabelecendo os insumos básicos necessários para a concretização desse processo.

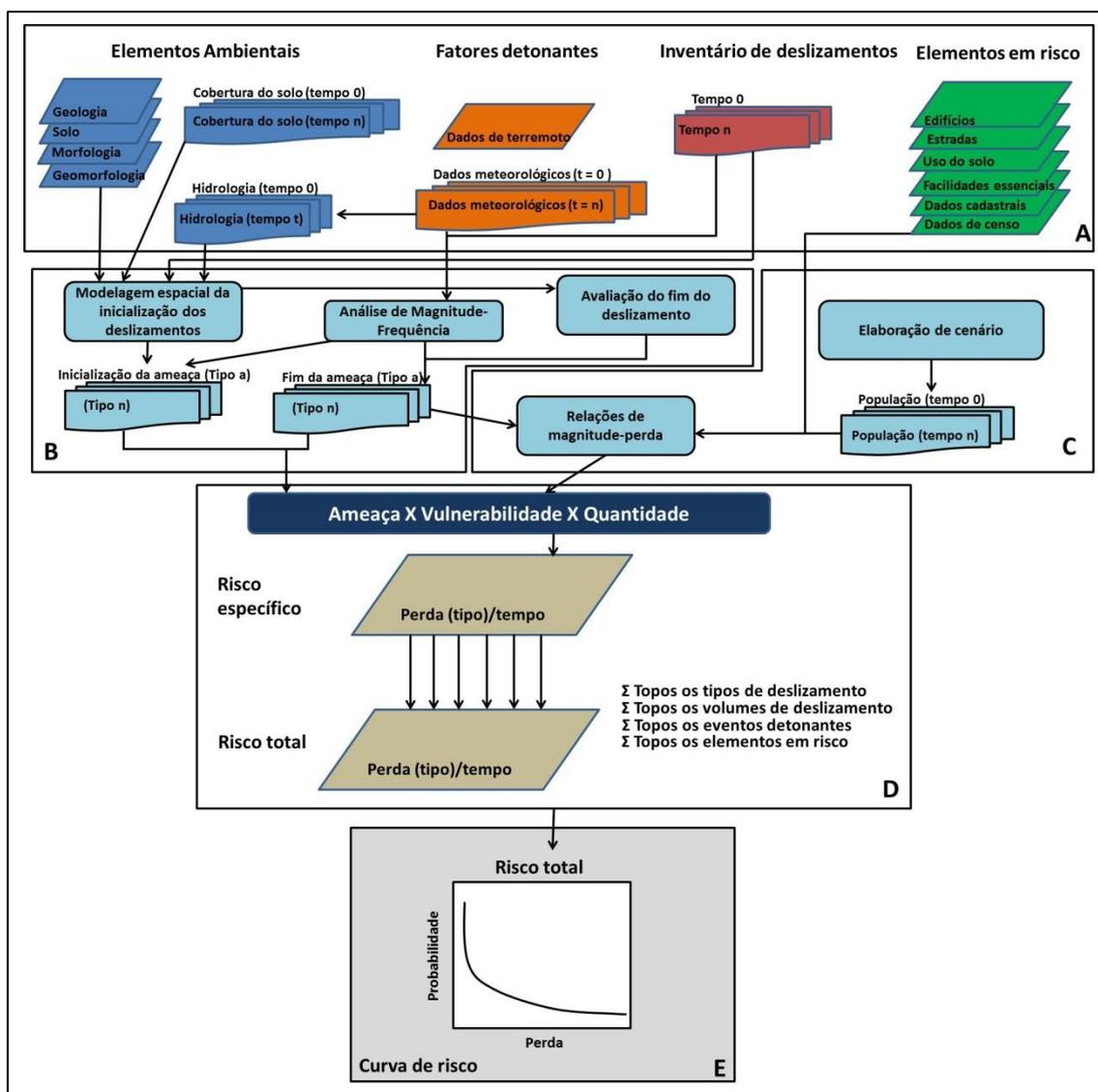


Figura 2.8. Representação esquemática do processo de avaliação de risco devido a deslizamentos. (modificado de Van Westen et al., 2008).

A Fig. 2.8 mostra o conjunto de informações requeridas no contexto dos processos de avaliação de risco devido a deslizamentos, onde se podem destacar as seguintes etapas:

(A): conjunto básico de dados necessários de natureza estática e dinâmica;

(B): componente para modelagem da susceptibilidade e ameaça;

(C): componente de avaliação de vulnerabilidade;

(D): componente de avaliação de risco;

(E): cálculo do risco total sob a forma de curva de risco.

Dentre os elementos de input, os inventários de deslizamento configuram-se como os mais importantes, uma vez que fornecem informações sobre a localização do fenômeno, frequência de ocorrência, mecanismo de ruptura, tipos, fatores causais e danos causados (Van Westen et al., 2008).

Em relação aos fatores deflagradores, estes são compostos por registros de chuvas e terremotos, que devem ser transformados em relações de magnitude-frequência daqueles aspectos que detonam efetivamente deslizamentos (Van Westen et al., 2005). Estes parâmetros mostram-se bastante específicos do local de estudo e apenas podem ser modelados corretamente, pelo uso de modelos determinísticos que necessitam de um considerável conjunto de dados de entrada em relação à caracterização geotécnica do terreno (permeabilidade, ângulo de atrito, coesão e espessura do solo).

Segue, então, uma explanação sobre estes principais dados espaciais, com o estabelecimento de suas principais características e a referência sobre alguns métodos para a obtenção destas informações.

2.3.3.1 INVENTÁRIOS DE DESLIZAMENTOS

Os inventários de deslizamentos se apresentam como a maneira mais simples de mapear escorregamentos (Wieczorek, 1984; Guzzetti et al., 1999). Entralgo (2013) chama a atenção para a existência de amplos estudos sobre as propriedades espaciais e estatísticas destes inventários, destacando a presença de pesquisas, cujo objetivo seria estabelecer uma relação entre a distribuição espacial dos eventos e as características espaciais dos agentes deflagrantes e condicionantes que compõem as regiões atingidas.

Conforme Malamud et al. (2004), pode-se classificar geralmente os inventários de escorregamentos em duas classes. A primeira categoria referencia-se aos inventários de eventos de deslizamentos que se encontram associados a um determinado deflagrador. A segunda classe está relacionada aos inventários históricos (geomorfológicos), que representam

a soma de um ou muitos deslizamentos que se processaram durante um período em uma região.

Guzzetti et al. (1999) destacam que as diferentes técnicas utilizadas para preparar mapas de inventários dependem da razão para a qual o mapa está sendo preparado, da extensão da área de estudo, das escalas dos mapas-base e das fotografias aéreas e dos recursos disponíveis para a realização do trabalho. Conforme Carrara et al. (2003), a ocorrência de deslizamentos e a ameaça a eles associados são, geralmente, avaliados com base em investigações geomorfológicas, realizadas por meio de fotografias aéreas e trabalhos de campo.

Os inventários de deslizamentos podem ser compilados a partir de diferentes fontes e em diferentes escalas (Malamud et al., 2004). Inventários em escalas menores que 1:200.000 podem ser produzidos a partir de perguntas realizadas a organizações públicas, através de pesquisas a jornais, relatórios técnicos e científicos (Radbruch-Hall et al., 1982). Segundo Cardinali et al. (1990), por meio de interpretação de fotos aéreas, também pode-se obter mapas de deslizamentos em pequenas escalas.

Inventários de média escala (1:25.000 a 1:200.000) são preparados, conforme Malamud et al. (2004), utilizando-se interpretação sistemática de fotos aéreas em escalas de impressão de 1:60.000 a 1:20.000 e vinculando verificações locais de campo com informações históricas.

Em relação aos inventários de grande escala, estes são preparados, geralmente, para áreas limitadas por meio de interpretações de fotos aéreas em escala de impressão maiores que 1:20.000 e pelo processamento de investigações de campo extensivas, as quais utilizam uma variedade de técnicas e ferramentas provenientes da geologia de engenharia, da geotecnia e da geomorfologia (Wieczorek, 1984).

Fell et al. (2008a) destacam as principais atividades requeridas para a elaboração de um inventário de deslizamentos para os diversos níveis de métodos relativos à caracterização.

Para o nível básico, a recomendação é que se faça uso de fotografias aéreas e/ou imagens de satélite, bem como mapeamento e incorporação de registros históricos, de forma que o inventário inclua a localização, volume, classificação e, quando possível, à data de ocorrência do escorregamento. Deve estar identificada, também, a relação entre geologia, geomorfologia e topografia.

A apresentação desta informação deve ser disposta sob a forma de mapas de inventários, juntamente com os dados topográficos, incluindo os contornos, os limites das

propriedades, mapeamento das grades, estradas e demais características de destaque, como córregos e cursos d'água.

Além das atividades estabelecidas para o nível básico, devem ser acrescentadas as seguintes ações, quando se estiver trabalhando em um nível intermediário:

- Distinguir diferentes partes do deslizamento;
- Mapear características dos deslizamentos e suas fronteiras;
- Coletar e analisar informação histórica no estado de atividade dos deslizamentos;
- Analisar a evolução passada do uso do solo, no sentido de conhecer se atividades antrópicas provocaram alguma influência na determinação da ocorrência de deslizamentos.

O nível avançado requer, somando-se às atividades desenvolvidas no nível anterior, a preparação de um inventário de dados geotécnicos, bem como a implementação de investigações que definam melhor as condições geotécnicas e a realização de uma análise geotécnica visando ao entendimento dos processos de instabilidade das encostas.

Em relação a outros tipos de fontes de dados, Metternicht et al. (2005) atentam que, para a interpretação visual de deslizamentos, necessita-se de imagens estereoscópicas com resolução variando de alta a muito alta.

Segundo IGOS (2004), imagens com resolução muito alta têm se mostrado como a melhor opção, atualmente, para o mapeamento de deslizamentos a partir de imagens de satélites. No entanto, a interpretação de fotos aéreas ainda se configura como sendo a técnica mais utilizada para o mapeamento de deslizamentos (Tribe & Leir, 2004).

Cheng et al. (2004) citam que imagens multi-espectrais (LANDSAT e ASTER, por exemplo) têm se mostrado mais aplicáveis ao mapeamento de deslizamentos, baseado na classificação de imagens em locais onde as ocorrências se mostram recentes e sem vegetação.

2.3.3.2 FATORES DEFLAGRADORES

Malamud et al. (2004) afirmam que os deslizamentos encontram-se frequentemente associados a um determinado deflagrador, como um terremoto, um rápido derretimento de neve ou uma chuva intensa. As rupturas de talude, variando de uma ruptura simples a dezenas de milhares, ocorrem, em geral, dentro de minutos após o terremoto deflagrador, horas ou dias após o degelo deflagrador, e dias ou semanas após uma chuva intensa que se mostra como o agente deflagrador.

As informações relacionadas com os elementos deflagradores geralmente apresentam uma importância mais temporal que espacial, salvo quando se trata de grandes áreas representadas em pequenas escalas (Van Westen et al., 2008). Estes tipos de dados mostram-se relacionados a registros de chuvas, temperaturas e terremotos, ao longo de um período suficientemente extenso, e à avaliação das relações de magnitude-frequência. Estações meteorológicas individuais captam os registros de chuvas e temperatura, enquanto os dados referentes aos terremotos estão normalmente disponibilizados em catálogos de terremotos.

Uma vez que informações suficientes de medição encontrem-se disponíveis, pode-se obter a distribuição espacial ao longo de toda área de estudo pela realização da interpolação dos dados pontuais. Como dado de entrada de um modelo dinâmico de estabilidade de encostas, pode-se gerar um mapa de precipitação máxima em 24 h para diferentes períodos de retorno.

Caso não existam dados disponíveis de estações meteorológicas para grandes áreas, podem-se utilizar estimativas gerais de chuva provenientes de imagens de satélites, tais como Tropical Rainfall Measuring Mission (TRMM) Multi-satellite Precipitation Analysis (TMPA), a qual é empregada para emitir alertas de deslizamentos.

2.3.3.3 FATORES AMBIENTAIS

Van Westen et al. (2008) determinam que, ao realizar uma avaliação de susceptibilidade, a escolha dos fatores ambientais que são utilizados neste processo depende do tipo de deslizamento, do tipo de terreno e da disponibilidade de dados existentes e recursos, de maneira que um bom entendimento dos diferentes mecanismos de ruptura mostra-se essencial.

Julião et al. (2009) citam que a litologia, formações superficiais, declividade, exposição e curvatura das vertentes, juntamente com a cobertura vegetal e uso do solo são variáveis necessárias ao desenvolvimento de uma carta de susceptibilidade à ocorrência de movimentos de massa, sendo que todos estes fatores apresentam um caráter ambiental.

Conforme Souza (2004), a metodologia atribuída à Escola de Engenharia de São Carlos determina que os atributos a serem utilizados na elaboração da cartografia geotécnica, tanto com a finalidade de solucionar problemas específicos, como na previsão de certas condições, devem ser escolhidos de acordo com a influência destes elementos em relação ao caso particular em análise. De forma mais específica, a carta derivada voltada para estabilidade de taludes, onde um dos objetivos seria prever possíveis problemas vinculados à

ocorrência de movimentos de materiais rochosos ou inconsolidados, utiliza os seguintes atributos na sua elaboração:

- Condições geomorfológicas: declividade, formas e dimensões das encostas/landforms, amplitude de relevo;
- Material inconsolidado: origem, profundidade, variação com a profundidade, resistência mecânica, textura, características físicas, condutividade hidráulica, umidade característica, presença de matacões, camadas compressíveis ou endurecidas, compactidade, expansibilidade;
- Substrato rochoso: origem, profundidade, variação com a profundidade, resistência mecânica, textura, características físicas, condutividade hidráulica, umidade característica, presença de matacões, camadas compressíveis ou endurecidas, compactidade, expansibilidade;
- Condições hidrogeológicas: profundidade do nível d'água, áreas de recarga, descarga e acúmulo de águas, infiltração, movimento da água subterrânea, fontes naturais;
- Processos geodinâmicos: movimentos de massa, subsidências, sismicidade;
- Feições do tecnógeno: áreas de aterros e entulhos, depósitos relacionados à exploração mineral, antigos depósitos de rejeitos e resíduos, processos geodinâmicos induzidos, práticas conservacionistas, obras que interferem no fluxo das águas;
- Climáticos: pluviosidade (intensidade e acumulada), temperatura, vento, insolação e evapotranspiração;
- Biológicos: profundidade das raízes, capacidade de retenção d'água pela vegetação, influência da vegetação e animais nos processos geodinâmicos;
- Antrópicos: uso do solo (alteração/concentração de fluxos d'água superficial e sub-superficial, impermeabilizações e carga).

Conforme Van Westen et al. (2005), a geração da representação digital da elevação da superfície desempenha um papel importante, uma vez que a topografia é um dos principais fatores na análise de ameaça de deslizamentos. Diferentes técnicas podem ser empregadas na elaboração de modelos digitais de terreno, tais como digitação de contornos a partir de mapas topográficos existentes, nivelamento topográfico, medida eletrônica de distância e fotogrametria digital.

Em se tratando de dados geomorfológicos, ressalta-se que os mapas referentes a estas informações são produzidos em variadas escalas no sentido de mostrar as unidades de terreno

baseadas na sua forma, material, processos e gênese, como verificado no trabalho de De Graaff et al. (1987). Para estes mapas, não há geralmente uma legenda determinada, sendo que se pode esperar uma variação quanto ao conteúdo do documento, dependendo da experiência do profissional que elabora o mapa (Van Westen et al., 2008). Segundo Gustavsson et al. (2006), estes mapas muito detalhados são ricos em informações, requerendo um extenso mapeamento de campo e tornando-se difíceis de serem convertidos para um formato digital.

A análise quantitativa das formas do terreno a partir de um Modelo Digital de Elevação, denominada geomorfometria ou análise digital do terreno, desponta, segundo Pike (2000), como um novo e importante campo da geomorfologia. Este trabalho pode estar concentrado na classificação automática de unidades de terreno, considerando características morfométricas em pequenas escalas (Miliarexis, 2001) ou na obtenção de faces de encostas em escalas médias que podem ser usadas em análises estatísticas, como unidades de mapeamento básico (Carrara et al., 1995).

Outra informação ambiental importante na avaliação de ameaça e riscos atrelados a deslizamentos refere-se ao uso do solo, sendo que este é um fator considerado como estático em muitos estudos de deslizamentos. Poucas pesquisas envolvem a mudança constante do uso do solo como um fator a ser considerado nas análises (Van Beek & Van Asch, 2004). Algumas atividades antrópicas impõem mudanças na cobertura vegetal e no uso do solo, que podem provocar um importante impacto na atividade dos deslizamentos (Glade, 2003).

2.3.3.4 ELEMENTOS EM RISCO

Para a determinação do risco e, conseqüentemente, para o zoneamento do risco decorrente de deslizamento ser implementado, deve-se avaliar os elementos que se encontram em risco (Fell et al., 2008a). Conforme Van Westen et al. (2008), dependendo da necessidade do estudo, pode-se realizar inventários de elementos sujeitos ao risco em variados níveis. A Tab. 2.2 lista as atividades necessárias para avaliar os elementos em risco, segundo o Manual Internacional para Zoneamento de Susceptibilidade, Ameaça e Risco decorrentes de Deslizamentos.

Tabela 2.2. Atividades requeridas para a avaliação dos elementos em risco segundo Fell et al. (2008a).

Métodos para avaliar elementos em risco	Atividades
Básico	Fazer uma avaliação da: população que vive, trabalha e trafega pela área; propriedades como casas, prédios, estradas, ferrovias e serviços permanentes na área; e propriedades tais como veículos que trafegam pela área. Para construções existentes, basear a avaliação no atual e proposto uso do solo. Para novas construções, estimar a partir do proposto uso do solo e ocupação. Quando possível, avaliar elementos do meio ambiente que podem ser afetados por deslizamentos. Classificações genéricas baseadas nos principais usos do solo.
Intermediário	Como acima, apresentando maior grau de detalhamento. Consequências econômicas podem ser incluídas.
Avançado	Como acima em detalhe. Consequências econômicas serão estimadas, bem como as implicações de perdas de estradas que dão acesso às cidades até que os reparos sejam realizados.

Segundo Van Westen et al. (2008), muitas vezes as unidade básicas para a análise de risco podem ser derivadas bancos de dados cadastrais já existentes, e os dados da população podem ser provenientes de dados de censos.

Van Westen et al. (2008) destacam ainda que, mesmo com a existência de informações digitais disponíveis, um considerável trabalho deve ser realizado no intuito de caracterizar os tipos de construções, mapear temporalmente as ocupações de construção e coletar

informações sobre a população através de pesquisas de campo, com o objetivo de desenvolver um banco de dados baseado em sistemas de informação geográfica.

2.4 BASE TEÓRICA DOS MODELOS EMPREGADOS NAS ANÁLISES

Nesta parte do documento, o foco está definido na apresentação dos trabalhos advindos da literatura técnica especializada que serviram de fundamentos para a criação do sistema de análise, que é o alvo maior desta pesquisa.

Foram os modelos e técnicas inclusos nos trabalhos que serão citados, que tiveram sua organização programada e modificada computacionalmente, de maneira estruturada, propiciando a obtenção dos variados componentes que, combinados, estabelecem o conceito de risco.

Especificamente, aqui será mostrada a formulação original concernente a estes modelos, destacando suas características, suas variáveis, bem como as técnicas e o tratamento das informações que são usados por parte de cada método.

O objetivo é estabelecer o arcabouço teórico empregado, proporcionando a noção do quão sólido e robusto, do ponto de vista técnico, é o conteúdo utilizado. Essa estrutura metodológica detalhada, para cada etapa, está explicada nos pontos que seguem, iniciando-se pela componente da ameaça.

2.4.1 TEORIAS APLICADAS PARA O CÁLCULO DA AMEAÇA

Em se tratando da determinação da quantificação da ameaça, quatro modelos serviram de embasamento para alicerçar esta fase do processo. Na maioria dos casos, estas abordagens possuem características de tratamento em termos determinísticos, fornecendo, como produto final, valores únicos para indicar a condição de estabilidade das encostas.

A incorporação da variabilidade dos parâmetros no cálculo da estabilidade dos taludes se processou durante a implementação das rotinas computacionais, e consiste em alterações impostas ao foco inicial encontrado nos trabalhos originais. O referido tema será abordado, posteriormente, no capítulo de metodologia, voltado especificamente para explicação destes aspectos. Por enquanto, nas subseções abaixo, estão expostas as metodologias, assim como os autores a serem citados as conceberam ou as utilizaram em seus trabalhos, sem qualquer alteração prévia, proposta até então.

2.4.1.1 CURVAS DE PROBABILIDADE DE RUPTURA EM FUNÇÃO DA PRECIPITAÇÃO ACUMULADA

A primeira pesquisa a ser mencionada, a respeito do cálculo da ameaça, relaciona-se à metodologia estabelecida no trabalho desenvolvido por Azevedo (2011), a qual se concentrou em obter e examinar informações pertinentes que proporcionassem a análise e o consequente entendimento da relação entre chuvas e escorregamentos no município do Rio de Janeiro.

O foco desta abordagem pautou-se na busca de um método que possibilitasse averiguar a possível interação entre as séries históricas de chuvas ocorridas na região de estudo e os registros de acidentes geotécnicos deflagrados por precipitações.

O tratamento da extensa gama de registros históricos coletados junto aos órgãos e instituições competentes do Rio de Janeiro, com a sua consequente organização, resultou em uma metodologia, a partir da qual foi passível o estabelecimento de etapas que permitiram a extração de novas informações, o que possibilitou nortear inferências instituídas em relação à interação existente no par precipitação e eventos de movimentos de massa.

Conforme Azevedo (2011), o desenvolvimento da composição da abordagem metodológica empregada no trabalho consistiu em duas etapas principais, relativas à organização de dados e, posteriormente, a averiguação da relação entre acidentes e chuvas.

A primeira parte citada, que se refere à organização de dados, consistiu, principalmente, na obtenção das informações básicas necessárias para o desenvolvimento do trabalho e na sua posterior depuração, com a filtragem dos elementos que possuíssem melhor exatidão.

Isso resultou na organização de um banco de dados, contando com aspectos geográficos, como topografia, geologia, sub-bacias hidrográficas e uso do solo, além da base de dados pluviométricos provenientes do sistema Alerta-Rio e de registros de deslizamentos efetuados pela Fundação Geo-Rio entre os anos de 1998 e 2002.

Azevedo (2011) realizou uma triagem dos dados básicos que seriam efetivamente incorporados à análise, uma vez que nem todos os elementos que foram obtidos forneciam características satisfatórias referentes a alguns critérios considerados como essenciais, para garantir resultados consistentes. Em relação aos registros de escorregamentos, adotou-se como regra de separação para esta categoria, a existência de precisão espacial e temporal dos dados. Desta forma, procurou-se selecionar os acidentes que possuíam, além da hora e dia

definidos (critério temporal), aqueles passíveis do georreferenciamento a partir do endereço fornecido no relatório da Geo-Rio (critério espacial).

O número total de registros presentes nos relatórios obtidos junto à Fundação Geo-Rio foi contabilizado em 1.079 escorregamentos, sendo que, uma vez imposta uma seleção baseada no critério de hora e data definidas (definição temporal), o número de acidentes reduziu-se a 372.

Apenas 172 deslizamentos foram possíveis de serem localizados espacialmente, com a obtenção de suas respectivas coordenadas geográficas. Observou-se que a maior concentração de pontos relacionados a escorregamentos ficou situada nas zonas sul e norte da cidade, optando-se, com o respaldo deste fato, pela definição de uma sub-região do município denominada de área piloto, representada pela Fig. 2.9, na qual as análises da pesquisa foram desenvolvidas, de forma que se contabilizou, para esta extensão, 163 registros devidamente catalogados. Com a definição da área de estudo propriamente dita, verificaram-se quais estações pluviométricas do sistema da Geo-Rio encontravam-se nesta região com o intuito de associar cada evento de escorregamento localizado no interior da área piloto a uma determinada estação pluviométrica e, conseqüentemente, a certo conjunto de dados de chuva. Assim, verificou-se a presença de 23 estações pertencentes à área piloto.

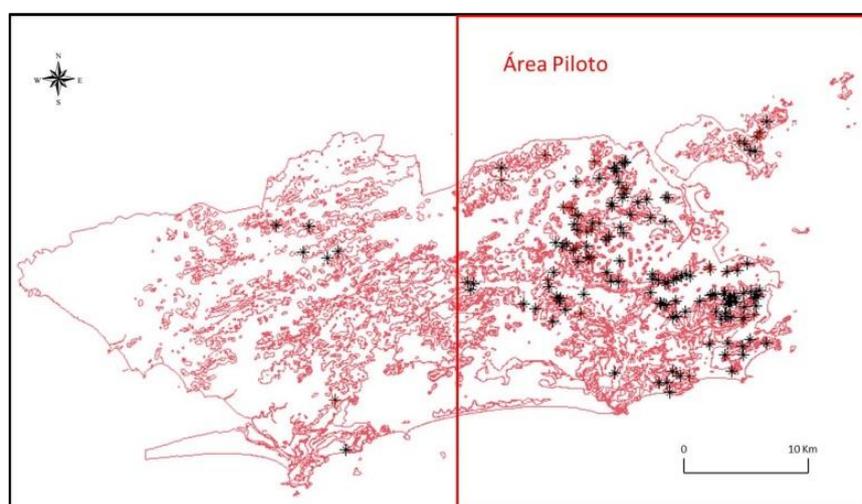


Figura 2.9. Área piloto de estudo utilizada na pesquisa de Azevedo (2011).

A partir disto, a área de influência para cada uma destas estações foi definida pela geração dos respectivos polígonos de Thiessen. Houve a modificação de alguns polígonos, levando-se em consideração a ação promovida pela topografia, bem como a delimitação das

sub-bacias hidrográficas. Na Fig. 2.10 estão representadas as áreas de influência das 23 estações presentes na área piloto.

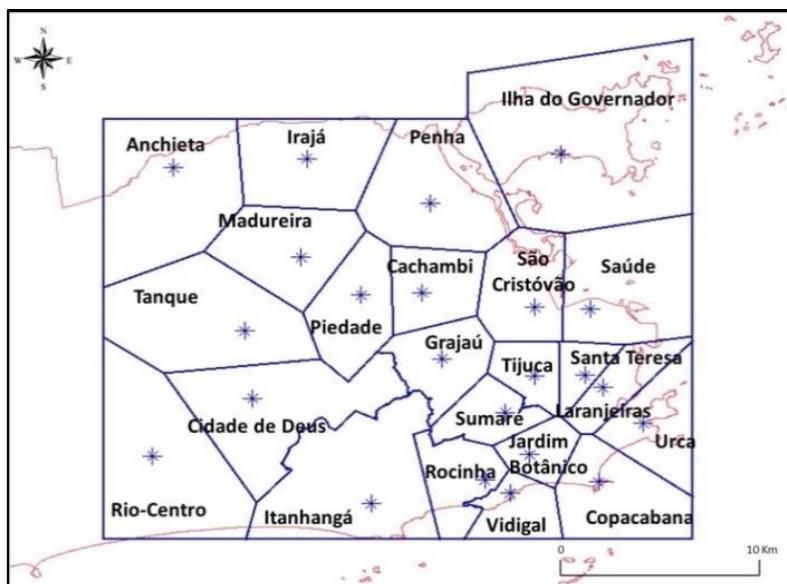


Figura 2.10. Áreas de influência das estações pluviométricas para a área piloto de estudo (Azevedo, 2011).

Trabalhando no interior da área piloto, o passo seguinte correspondeu à associação de cada evento de escorregamento a uma determinada estação pluviométrica. Desta forma, depois de estabelecido este vínculo e de posse dos dados históricos de chuva devidamente organizados em matrizes para cada posto, processou-se o cálculo dos acumulados de chuva associados a cada evento de escorregamento, tomando-se como base os intervalos de tempo utilizados pelo sistema Alerta Rio na composição dos seus níveis de alerta. Estes limites foram cedidos pela Fundação Geo-Rio e encontram-se citados na Tab. 2.3.

Tabela 2.3. Níveis de alerta para escorregamentos do sistema Alerta Rio (Azevedo, 2011).

	NÍVEL MÉDIO	NÍVEL ALTO	NÍVEL MUITO ALTO
Acumulado em 1 h	10 a 30 mm	30 a 50 mm	Maior que 50 mm
Acumulado em 24 h	50 a 100 mm	100 a 175 mm	Maior que 175 mm
Acumulado em 96 h	100 a 175 mm e 10 a 30 mm/24h	175 a 250 mm e 30 a 50 mm/24h	Maior que 250 mm e maior que 100 mm/24h

A computação dos acumulados pluviométricos relacionados aos períodos de 1 h, 24 h e 96 h foi realizada com o auxílio de funções desenvolvidas no programa Matlab (Matrix

Laboratory), específicas para este desígnio e que apresentavam, como parâmetros de entrada, os dados de chuva e de acidentes geotécnicos, devidamente organizados em matrizes.

No entanto, observou-se que os valores acumulados calculados de muitos registros não se enquadravam em nenhum dos níveis estabelecidos pelo sistema, apresentando precipitação acumulada calculada menor que os limites inferiores propostos para o nível médio. Azevedo (2011) optou, então, por criar um intervalo teórico, denominado nível baixo, que contemplasse as precipitações acumuladas inferiores ao nível médio para os períodos de 1 h, 24 h e 96 h, permitindo, assim, a associação dos escorregamentos presentes nesta situação. Este novo nível incorporado à análise está definido como mostrado na Tab. 2.4.

Tabela 2.4. Definição do nível baixo de ocorrência de deslizamentos (Azevedo, 2011).

Precipitação acumulada	Nível Baixo
1 h	Entre 0 e 10 mm
24 h	Entre 0 e 50 mm
96 h	Entre 0 e 100 mm e entre 0 e 10 mm/24h

Para os resultados do nível de 96 h, observou-se que a maioria dos deslizamentos não se enquadraram nos limiares estabelecidos pelo sistema de alerta. Desta forma, por se tratar de um critério conjugado, inúmeras possibilidades envolvendo o acumulado de 96 h que melhor representassem a distribuição dos deslizamentos, mostraram-se possíveis. Como a adoção de um novo intervalo de precipitação representaria o abandono da proposta inicial, Azevedo (2011) não deu prosseguimento a esta abordagem, seguindo apenas com os valores acumulados em 1 h e 24 h.

Pelo fato de algumas estações pluviométricas possuírem, individualmente, um número baixo de ocorrências de deslizamentos vinculadas às suas áreas de influência e, no intuito de tornar estas informações mais robustas, do ponto de vista quantitativo, procedeu-se o agrupamento dos dados de postos adjacentes que se encontravam sob esta condição, resultando, ao final, que um mesmo gráfico pudesse representar o comportamento de estabilidade das encostas para distintas estações.

Para a determinação da probabilidade de ruptura associada a cada nível de alerta, necessitou-se investigar a incidência dos casos nos quais os acumulados de precipitação satisfizeram algum dos níveis estabelecidos pelo sistema de alerta, mas que não resultaram, de fato, em deslizamentos. A definição destes episódios, denominados Falsos Positivos,

correspondeu, primeiramente, no cálculo de todos os acumulados de chuva para 1 hora e 24 horas, no período de 1998 a 2002, para cada estação, com o auxílio das mesmas funções desenvolvidas no programa Matlab utilizadas para a computação dos acumulados de precipitação, relativos aos acidentes presentes no inventário de escorregamentos. Instaurou-se a triagem destas informações, separando os acumulados encontrados por nível de alerta. Ao final, alcançou-se a separação dos falsos positivos por nível de alerta para cada um dos postos estudados, podendo-se, ainda, mensurar as suas quantidades em cada um destes níveis.

A probabilidade de ruptura foi processada, para cada nível de alerta, mediante a razão entre a quantidade de escorregamentos e o número total de acumulados pluviométricos, este último representado pela soma do número de falsos positivos e de escorregamentos. Em relação aos níveis de alerta, associou-se a probabilidade de ruptura obtida ao valor médio do respectivo intervalo estabelecido pelos limites de cada nível de alerta proposto.

Com isso, foi possível representar graficamente a tendência destas duas grandezas (precipitações como abscissas e probabilidades de ruptura como ordenadas), executando-se, ainda, o ajuste da função que melhor representasse o conjunto de dados. As curvas de probabilidade de ruptura obtidas em função da precipitação acumulada de 1 h e 24 h para os grupos associados de estações estão representadas na Fig. 2.11 e na Fig. 2.12, respectivamente.

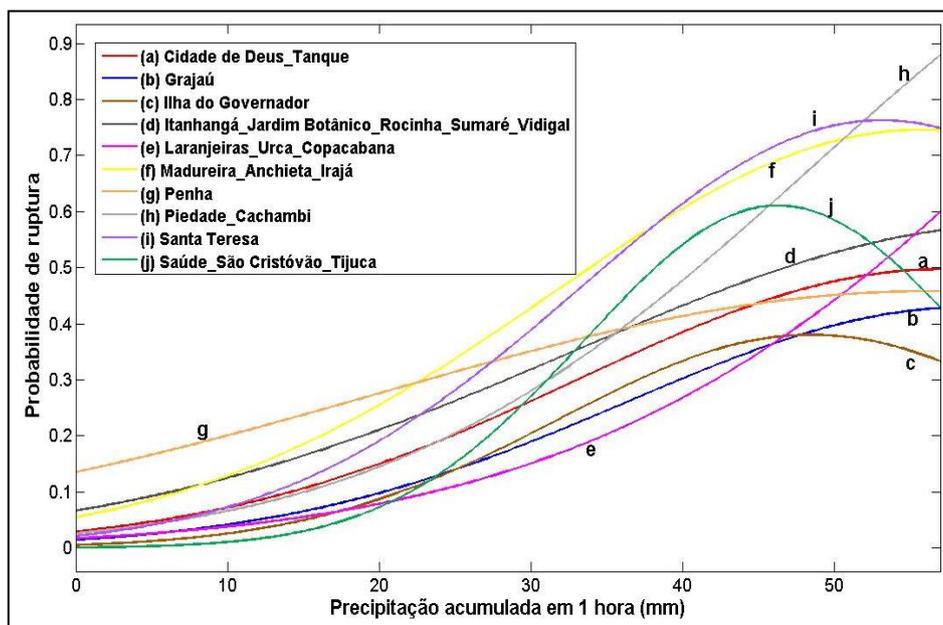


Figura 2.11. Curvas de probabilidade de ruptura contra precipitação acumulada de 1 hora (Azevedo, 2011).

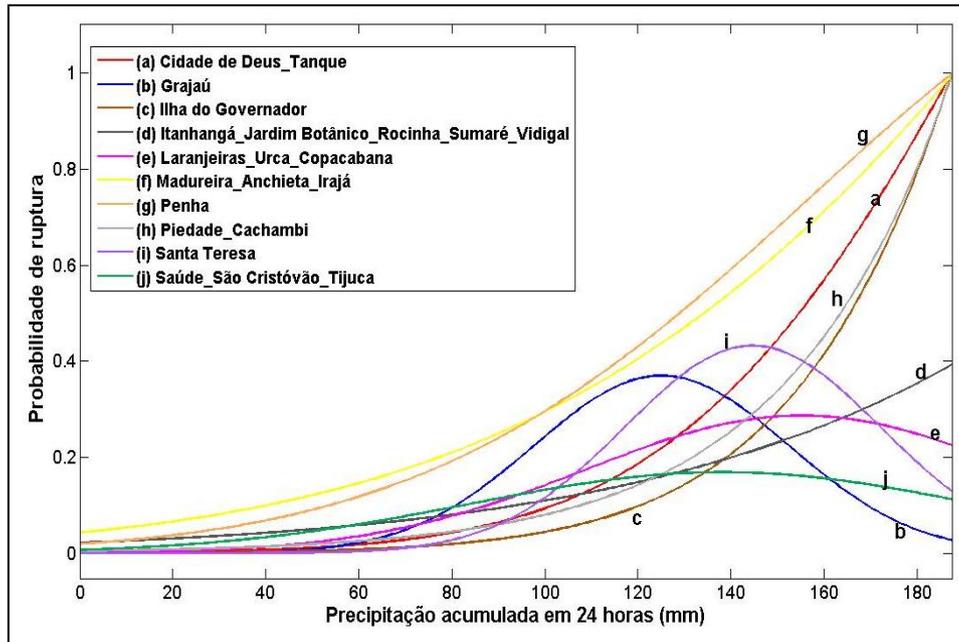


Figura 2.12. Curvas de probabilidade de ruptura contra precipitação acumulada de 24 horas (Azevedo, 2011).

Para o ajuste destas funções, foram usadas curvas gaussianas que seguem a equação abaixo.

$$PR = a.e^{\left[-\left(\frac{p-b}{c}\right)^2\right]} \quad (2.54)$$

Onde PR corresponde à probabilidade de ruptura fornecida por uma determinada curva, limitada em um intervalo que varia de 0 a 1. A medida p é a precipitação acumulada, em mm, referente a um determinado período de tempo (1 h ou 24 h);

Os valores a, b e c referem-se aos coeficientes específicos que caracterizam as diferentes funções de probabilidade de ruptura para a região analisada, e que foram previamente alcançados pela aplicação da metodologia explanada.

2.4.1.2 MODELO DE BASE FÍSICA PARA PREVISÃO DE DESLIZAMENTOS SHIA_LANDSLIDE

Em sua tese de doutorado, Aristizábal (2014) propôs um modelo conceitual e físico, denominado SHIA_Landslide, para a previsão de escorregamentos rasos detonados por chuvas em ambientes tropicais montanhosos e em terrenos complexos, suportados por aspectos geológicos e hidrológicos que ocorrem em escala de bacia.

Conforme Aristizábal (2014), a abordagem oferece uma nova perspectiva para a análise desta categoria de problemas em escala de bacias hidrográficas, empregando este completo e abrangente modelo, que envolve o conceito de tanque hidrológico, incluindo o armazenamento de água no solo de forma acoplada a uma análise geotécnica de estabilidade de taludes infinitos em condições saturadas. A implementação do algoritmo foi realizada na linguagem FORTRAN, com a aplicação a um caso real em uma região andina da Colômbia, como forma de testar a sua capacidade preditiva.

O módulo hidrológico utilizado no modelo baseia-se em uma metodologia desenvolvida por Vélez (2001), denominada Simulação Hidrológica Distribuída e Aberta (em espanhol, Simulación Hidrológica Abierta - SHIA), formado por um componente de balanço hídrico, responsável por simular os processos hidrológicos dominantes na bacia, e um elemento de direcionamento, capaz de retratar o fluxo de água através da rede de rios.

O procedimento corresponde, em um primeiro momento, ao estabelecimento de uma malha tridimensional de tanques conectados entre si para representar o terreno, em se tratando das suas condições de superfície, subsolo e fluxos de base, sendo que esta malha é drenada para um tanque correspondente, o qual se localiza a jusante, seguindo a direção do fluxo da superfície até que se possa atingir a rede de canais.

No sentido de controlar os processos envolvidos na infiltração e percolação, a bacia hidrográfica estudada é discretizada pelo programa computacional em grades organizadas em camadas horizontais regulares. Cada célula da grade pode ser composta por três camadas, cada uma com um diferente valor de condutividade hidráulica saturada, dependendo do perfil de intemperismo tropical assumido. Assim, podem-se considerar na configuração, em ordem decrescente de permeabilidade, os materiais referentes ao solo residual (K_s), saprólito (K_p) e rocha (K_{pp}), com as suas respectivas condutividades hidráulicas entre parênteses.

O módulo hidrológico considera o movimento da água entre as camadas de solo (direção vertical) e células adjacentes (direção horizontal), simulando, assim, o fluxo através da bacia discretizada. Um sistema de cinco tanques interconectados está presente em cada célula, sendo que este sistema ainda se comunica com os respectivos tanques nas células a jusante. Isto retrata o fluxo e o armazenamento de água, envolvendo os processos hidrológicos de interceptação, detenção, infiltração, evapotranspiração, escoamento superficial, percolação, escoamento subsuperficial, e o retorno do fluxo de base aos canais do sistema de drenagem.

O modelo define, primeiramente, a parcela da chuva que é interceptada pela vegetação da bacia e a parcela da água capilar que penetra no solo. A partir disto, estima-se que uma parcela da chuva infiltra por gravidade, enquanto o restante irá para o escoamento superficial. O armazenamento de água gravitacional do solo encontra-se dividido pelo solo residual, com uma alta permeabilidade, e pelo saprólito, que possui uma baixa permeabilidade e uma resposta mais lenta.

O primeiro tanque (T1) refere-se à interceptação e à retenção de água em poças, assim como o armazenamento da água por capilaridade na parte superior do solo, sendo denominado de armazenamento estático. Para este tipo de tanque, não ocorre transferência horizontal de água pela bacia, nem escoamento superficial.

A chuva, indicada pelo termo R_1 , é inicialmente estocada no tanque de armazenamento estático, até que seja atingida a sua capacidade máxima. Para um passo de tempo, a quantidade de água que entra no armazenamento estático (D_1) depende da capacidade máxima de T1 (S_{1max}), sendo dada pela seguinte expressão:

$$D_1 = Min \left\{ R_1 \left[1 - \left(\frac{S_1^*}{S_{1max}} \right)^2 \right], S_{1max} - S_1^* \right\} \quad (2.55)$$

Nessa equação, Min representa o valor mínimo entre aqueles apresentados, enquanto S_1^* é o volume de água em T1 no final do passo anterior. À medida que o volume de água cresce em T1, o conteúdo de água que deve entrar no tanque tende a diminuir, de maneira que o máximo volume que poderia entrar no tanque acontece quando este se encontra vazio, representando a condição do solo completamente seco.

A definição S_{1max} depende da capacidade de campo (W_{fc}), que representa o conteúdo de água do solo, em que toda a água livre do solo encontra-se drenada pela ação da gravidade, e do ponto de murcha do solo (W_{pwp}), o qual é verificado quando o conteúdo volumétrico de água é muito baixo, impedindo que a planta remova água do solo, fazendo com que ela murche e morra. A equação abaixo determina o valor de S_{1max} .

$$S_{1max} = (W_{fc} - W_{pwp}) Z_r \quad (2.56)$$

Nesta equação, Z_r define a profundidade das raízes. Aristizábal (2014) observa que S_{1max} pode ser definido como a água armazenada no solo disponível à planta, a qual pode aumentar de acordo com a capacidade oferecida pela cobertura superficial, em se tratando da

retenção de água. A Fig. 2.13 retrata o modelo hidrológico, ressaltando os respectivos tanques que o compõem.

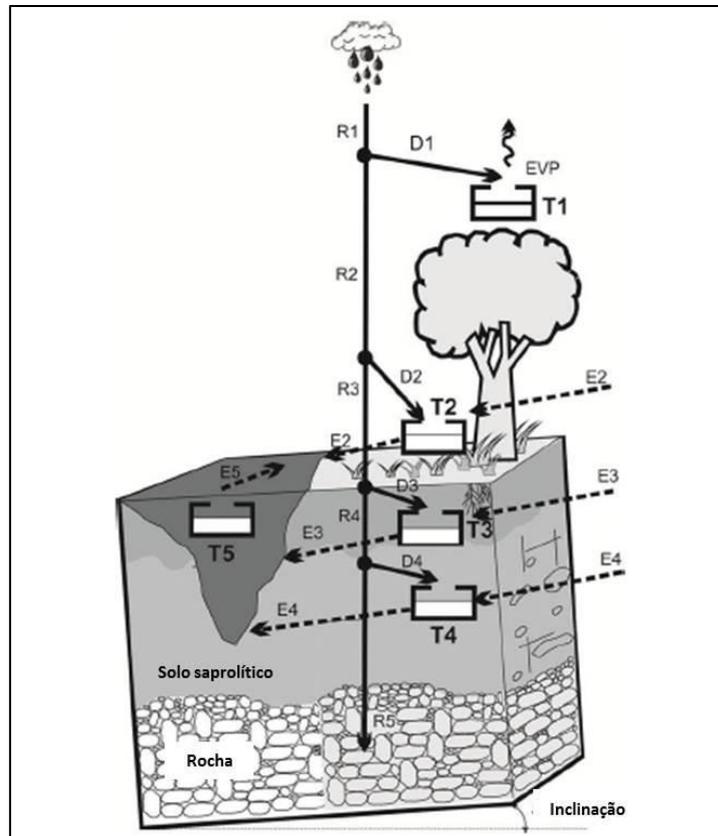


Figura 2.13. Modelo hidrológico conceitual com seus respectivos tanques (modificado de Aristizábal, 2014).

Na Fig. 2.13, tem-se os seguintes componentes: armazenamento estático (T1), armazenamento superficial (T2), armazenamento gravitacional (T3), aquífero (T4), canal (T5), chuva (R1), excedência (R2), infiltração (R3), percolação (R4), fluxo de saída de água subterrânea (R5), escoamento superficial (E2), escoamento subsuperficial (E3), fluxo de base (E4), fluxo de canal (E5), fluxo de entrada nos tanques (D1 a D5) e evapotranspiração (EVP).

Quando o volume de água em T1 se aproxima da sua capacidade máxima, o excedente encaminhado para os demais tanques é incrementado. O excesso de água, que é enviado para o segundo tanque (T2) a partir de T1, é dado pela seguinte expressão:

$$R_2 = R_1 - D_1 \quad (2.57)$$

A atualização do volume de água em T1 (S_1) para cada iteração no tempo leva em consideração a capacidade máxima S_{1max} do tanque:

$$S_1 = \text{Min}(S_1^* + R_1 - R_2, S_{1\text{max}}) \quad (2.58)$$

A evapotranspiração (E_1) configura-se como o único fluxo de saída deste primeiro tanque, sendo definida conforme a seguinte equação.

$$E_1 = \text{Min}\left\{ \text{Evp} \left(\frac{S_1}{S_{1\text{max}}} \right)^{0,6}, S_1 \right\} \quad (2.59)$$

Onde Evp é a evaporação potencial, sendo estabelecida de acordo com a área avaliada.

Denomina-se armazenamento superficial, o segundo tanque T2 que compõe o modelo. Este elemento retrata a água que se situa na superfície da encosta, não chegando a infiltrar e que percola sob a inclinação do talude. A quantidade de água que permanece e infiltra (R_3) é proposta por:

$$R_3 = \text{Min}(R_2, K_s) \quad (2.60)$$

Onde K_s refere-se à permeabilidade saturada da camada superior, o que significa que a permeabilidade saturada e a capacidade do solo residual em receber mais água definem o controle da infiltração.

Em um intervalo de tempo, o tanque T2 de armazenamento de água recebe o seguinte volume:

$$D_2 = R_2 - R_3 \quad (2.61)$$

Já a atualização do conteúdo de T2 é indicada pela expressão abaixo.

$$S_2 = S_2^* + D_2 + Z_3 \quad (2.62)$$

O termo S_2^* é o volume de água em T2 no fim do passo anterior, enquanto Z_3 corresponde à quantidade de água excedente do tanque 3, quando este está totalmente cheio.

Podem-se usar diferentes caminhos para o cálculo do fluxo de superfície em cada célula (E_2), baseando-se, por exemplo, no uso da velocidade constante com um reservatório linear ou uma abordagem não linear. Para velocidade constante, E_2 pode ser estimado pautando-se na fórmula aqui expressa:

$$E_2 = \alpha S_2 = \left(1 - \frac{dx}{v_2 \cdot dt + dx} \right) S_2 \quad (2.63)$$

O termo α corresponde ao coeficiente de descarga do reservatório linear, definido como função do tamanho da célula (dx), resolução temporal (dt) e velocidade na superfície da encosta (v_2).

Aristizábal (2014) cita que muitos autores recomendam um fluxo uniforme como o representado pela equação de Manning na abordagem não linear para a definição da velocidade do fluxo superficial, de acordo com a seguinte equação.

$$V_2 = \frac{\xi \cdot A \left(\frac{2}{3}\right)^{e_1} \beta^{\frac{1}{2}}}{n} \quad (2.64)$$

Onde A refere-se à área da seção transversal na qual ocorre o fluxo, β é a inclinação da encosta, n corresponde ao coeficiente de Manning e os parâmetros ξ e e_1 associam-se ao tipo de superfície.

A área A se modifica com a velocidade e o fluxo de saída da seguinte forma:

$$A = \frac{S_2}{dx + v \cdot dt} \quad (2.65)$$

Assumindo-se um valor para a velocidade inicial ($v_{inicial}$), obtém-se a área da seção e, a partir deste valor, a velocidade (v_{cal}) é calculada. Repete-se este processo iterativamente por duas vezes para um mesmo intervalo de tempo, no intuito de gerar um valor convergente para a velocidade (v_{med}), segundo a fórmula:

$$v_{med} = \frac{2 \cdot v_{cal} + v_{inicial}}{3} \quad (2.66)$$

Utilizando a velocidade do escoamento superficial obtida, pode-se determinar o fluxo de saída deste tanque em direção a uma célula a jusante, pela aplicação da expressão seguinte.

$$E_2 = A \cdot v_{med} \cdot \frac{dt}{dx} \quad (2.67)$$

Assim, a atualização do volume de T2, em certo passo de tempo, é dada por:

$$S_2 = S_2 - E_2 \quad (2.68)$$

O terceiro tanque T3 está ligado ao armazenamento de água gravitacional no solo residual, localizando-se entre a capacidade do solo e a saturação. A máxima capacidade de T3 pode ser determinada por:

$$S_{3max} = (W_s - W_{fc}) Z_s \quad (2.69)$$

Onde W_s é definido como a saturação (situação que ocorre quando todos os vazios interconectados do solo encontram-se totalmente preenchidos por água). Z_s é a espessura do solo. O volume que percola em direção ao saprólito será:

$$R_4 = \text{Min}(R_3, K_p) \quad (2.70)$$

Na sequência, o volume de água que chega ao tanque T3 (D_3) e o conteúdo que é atualizado (S_3) são dados, respectivamente, pela expressões abaixo.

$$D_3 = R_3 - R_4 \quad (2.71)$$

$$S_3 = \text{Min}\{S_3^* + D_3, S_{3\text{max}}\} \quad (2.72)$$

Na última das equações mostradas, S_3^* refere-se ao volume de água em T3, ao final do passo anterior. No momento em que se atinge a capacidade máxima de T3, um excedente, denominado Z_3 , é produzido conforme a equação abaixo, sendo que este volume destina-se ao escoamento superficial.

$$Z_3 = \text{Max}\{0, S_3 + D_3 - S_{3\text{max}}\} \quad (2.73)$$

O operador Max retorna o maior dos valores apresentados como argumentos.

Assim, como para o tanque 2, Aristizábal (2014) estabelece que é possível usar, como alternativa, uma abordagem linear ou não linear no cálculo do fluxo subsuperficial. Enquanto que, para o primeiro caso, emprega-se a Eq. 2.63, em que a velocidade constante é fornecida pela permeabilidade da parte superior do solo, no enfoque não linear, a velocidade do escoamento subsuperficial é obtida segundo a equação abaixo, como um fluxo lateral em regiões montanhosas cobertas por floresta, de acordo com Kubota & Sivapalan (1995).

$$V_3 = \frac{K_s \cdot \sin \beta}{(b+1)(S_{3\text{max}})^b} (S_3)^b \quad (2.74)$$

onde K_s é a condutividade hidráulica saturada, β refere-se à declividade e b corresponde a um parâmetro que depende do tipo do solo.

Em relação a uma bacia montanhosa composta por florestas, correspondendo a uma permeabilidade homogênea ao longo de todo perfil de intemperismo, Kubota & Sivapalan (1995) definem que o valor de b deva ser igual a 2. O volume do tanque, dado por S_3 , é definido por:

$$S_3 = A \cdot dx \quad (2.75)$$

A área da seção transversal (A), a velocidade média e o volume de água que sai de T3 para uma célula a jusante, de acordo com a velocidade do escoamento subsuperficial, podem ser estimados segundo a técnica aplicada ao tanque 2. A atualização do volume em T3, levando em conta o fluxo de saída E_3 , é efetuada conforme a expressão abaixo:

$$S_3 = S_3 - E_3 \quad (2.76)$$

O quarto tanque T4 modela o fluxo e o armazenamento no aquífero. O volume do escoamento das águas subterrâneas é fornecido por:

$$R_5 = \text{Min}(R_4, K_{pp}) \quad (2.77)$$

Nesta expressão, K_{pp} é o fluxo subterrâneo, que pode ser entendido como as perdas de água. A equação abaixo define o volume de água que entra em T4 durante um intervalo de tempo.

$$D_4 = R_4 - R_5 \quad (2.78)$$

Pode-se, então, definir o volume atualizado em T4 para cada intervalo de tempo como sendo igual a:

$$S_4 = S_4^* + D_4 \quad (2.79)$$

Pode-se estimar o fluxo, a partir deste tanque de armazenamento, para um pixel a jusante pelo emprego da Eq. 2.63, tomando como referência o próprio nível do tanque com um coeficiente de descarga que pode ser relacionado com a condutividade hidráulica saturada do aquífero. A fórmula utilizada para que o volume de água em T4 seja atualizado, considera o fluxo de saída no passo de tempo considerado:

$$S_4 = S_4 - E_4 \quad (2.80)$$

O último tanque retrata o canal de escoamento presente na célula, de maneira que a rede de drenagem e os modelos de fluxo presentes na bacia definem como cada célula está conectada com um elemento a jusante. Fluxo superficial, fluxo subsuperficial e fluxo de base são coletados por meio da rede de canais do rio, representados por T5. As formas para o cálculo da velocidade de escoamento em T5 são similares às aquelas já mostradas para os tanques anteriores.

Enquanto as conexões verticais entre os tanques retratam fenômenos de chuva, evapotranspiração, infiltração e percolação, uma sub-rotina do modelo é responsável por

determinar, simultaneamente a estes processos, a transferência horizontal de água entre células adjacentes (correspondente ao escoamento superficial, fluxo interno e fluxo de base), segundo a direção de fluxo entre estes pixels, o qual é estabelecido pela topologia da bacia. A Fig. 2.14 retrata os tipos de célula do modelo e as conexões entre os tanques.

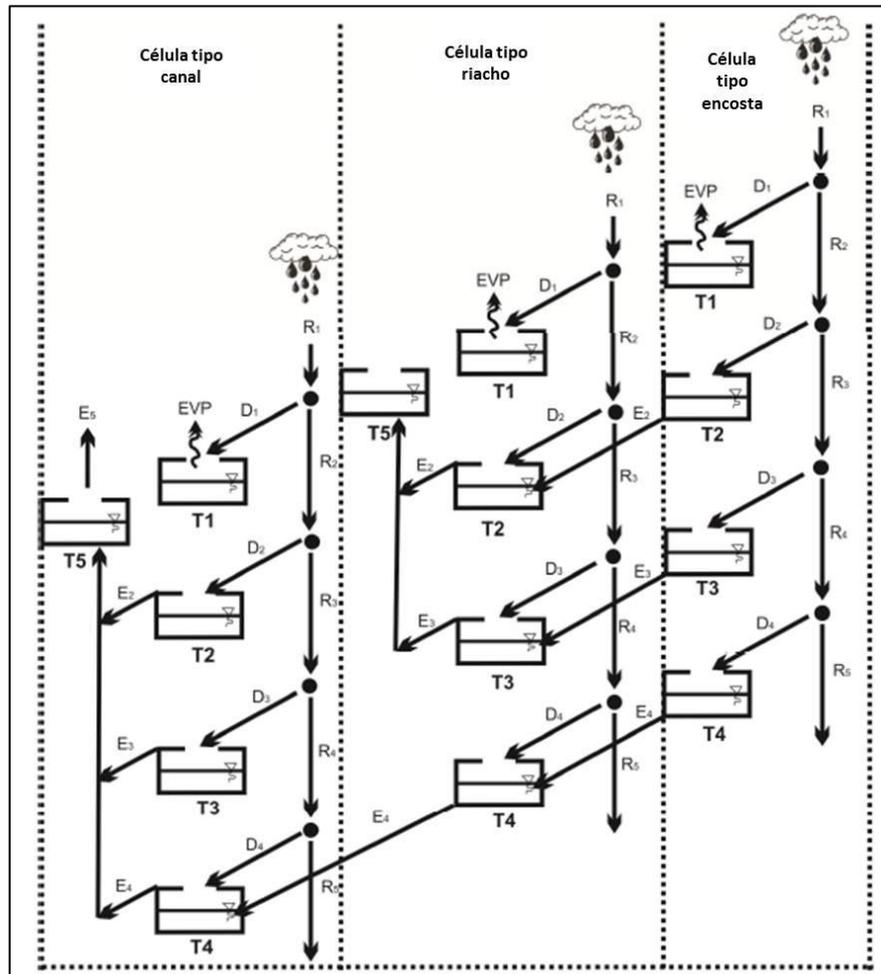


Figura 2.14. Conexão entre tanques do módulo hidrológico (modificado de Aristizábal, 2014).

Conforme mostrado na Fig. 2.14, existem três possibilidades quanto ao tipo de células que podem formar a grade da bacia, de forma que a conexão existente entre os tanques dependerá efetivamente da categoria em que o pixel esteja inserido. As células podem ser classificadas em:

- Taludes;
- Canais efêmeros ou riachos;
- Canais perenes.

Destaca-se que apenas as células dos dois últimos tipos de canais citados possuem o tanque T5.

A atribuição do tipo de cada célula que compõe a bacia estudada irá depender da área acumulada (limiar de área), a qual é definida com base em estudos locais e observações de campo. São requeridos dois limites como parâmetros de entrada: a área mínima acumulada necessária para formar um canal efêmero, situado entre os fluxos superficiais e subsuperficial, e a área acumulada mínima requerida para formar um canal perene, entre o fluxo subsuperficial e o fluxo de base.

Para células do tipo encosta, o fluxo horizontal entre tanques acontece em um mesmo nível para elementos correspondentes, onde é possível ocorrer a transferência, menos para o tanque T1, já que este não transfere água de maneira similar aos demais, uma vez que o único fluxo de saída corresponde à evapotranspiração. O fluxo horizontal se desenvolve apenas entre os tanques T4 para células da grade classificadas como canais efêmeros. Os demais fluxos correspondentes a T2 e T3 seguem para T5. Para canais perenes, o fluxo se estabelece apenas entre tanques T5. Enquanto uma célula definida como encosta pode escoar para qualquer tipo de classe, as células da categoria riacho podem ser drenadas por outras células deste mesmo tipo ou por elementos de canais perenes. Células do tipo canal escoam apenas para esta mesma configuração de célula.

Em se tratando do módulo geotécnico adotado por Aristizábal (2014), este se baseia no uso de um modelo de estabilidade de taludes infinitos, hipótese aceita em casos onde a espessura do solo residual é limitada, quando comparada com o comprimento do talude. Por meio deste modelo e em termos de tensões efetivas, o fator de segurança do talude FS pode ser representado pela equação abaixo.

$$FS = \frac{c' + (\gamma \cdot Z \cdot \cos^2 \beta - u) \tan \phi}{\gamma \cdot Z \cdot \sin \beta \cdot \cos \beta} \quad (2.81)$$

Onde c' refere-se à coesão efetiva, ϕ relaciona-se ao ângulo de atrito efetivo, γ é o peso específico do solo, Z corresponde à espessura do solo medida verticalmente, β é a declividade do talude e u está ligado à poropressão.

A poropressão em taludes naturais, com um fluxo estacionário que se desenvolve paralelo à encosta e com o nível freático situado à distância Z_w acima da superfície de deslizamento, pode ser escrito, conforme Graham (1984), pela seguinte expressão:

$$u = \gamma_w \cdot Z_w \cdot \cos^2 \beta \quad (2.82)$$

Na equação anterior, γ_w representa o peso específico da água. Assim, o fator de segurança fica definido como:

$$FS = \frac{c' + (\gamma \cdot Z - \gamma_w \cdot Z_w) \cdot \cos^2 \beta \cdot \tan \phi}{\gamma \cdot Z \cdot \sin \beta \cdot \cos \beta} \quad (2.83)$$

Quando a condição de equilíbrio limite é atingida ($FS = 1$), tem-se:

$$c' + (\gamma \cdot Z - \gamma_w \cdot Z_w) \cdot \cos^2 \beta \cdot \tan \phi = \gamma \cdot Z \cdot \sin \beta \cdot \cos \beta \quad (2.84)$$

A solução da equação acima, em função da variável Z_w , fornece o valor crítico da camada saturada capaz de provocar o deslizamento:

$$Z_{wcrit} = \frac{\gamma}{\gamma_w} \cdot Z \cdot \left(1 - \frac{\tan \beta}{\tan \phi} \right) + \frac{c'}{\gamma_w \cdot \cos^2 \beta \cdot \tan \phi} \quad (2.85)$$

Em cada intervalo de tempo, a avaliação da estabilidade dos taludes em cada uma das células da malha é processada pela comparação entre a altura do nível freático e o valor crítico Z_{wcrit} . A altura do lençol freático pode ser determinada a partir do conteúdo de água presente no armazenamento gravitacional fornecido pelo componente hidrológico do modelo, utilizando-se a seguinte equação:

$$Z_w = \frac{S_3}{(W_s - W_{fc})} \quad (2.86)$$

O modelo atua definindo, inicialmente, as condições extremas de estabilidade, na tentativa de impor uma melhor eficiência computacional ao processo, estabelecendo, previamente, as espessuras mínimas e máximas limites para o solo residual. Isto possibilitará categorizar aquelas células ditas incondicionalmente instáveis e aquelas incondicionalmente estáveis, uma vez que estes dois estados não dependem do conteúdo de água.

Para a condição $Z_w = Z$ na Eq. 2.84, obtém-se a profundidade imune à instabilidade, denominada Z_{min} :

$$Z_{min} = \frac{c'}{\gamma_w \cdot \cos^2 \beta \cdot \tan \phi + \gamma \cdot \cos^2 \beta \cdot (\tan \beta - \tan \phi)} \quad (2.87)$$

Já a espessura Z_{max} é definida quando $Z_w = 0$ é imposta a Eq. 2.84:

$$Z_{max} = \frac{c'}{\gamma \cdot \cos^2 \beta \cdot (\tan \beta - \tan \phi)} \quad (2.88)$$

O solo é sempre instável para espessuras superiores a Z_{\max} . Para esta espessura Z_{\max} , segundo Iida (1999), a profundidade saturada necessária para deflagrar um deslizamento é nula, com o solo mostrando-se instável independente da ocorrência de chuvas. As condições extremas de estabilidade adotadas estão expressas na Fig. 2.15.

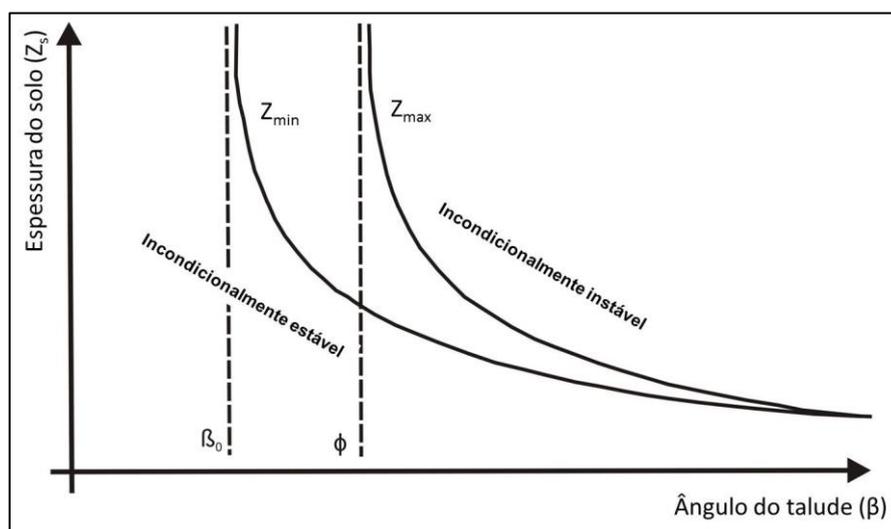


Figura 2.15. Representação da susceptibilidade à ocorrência de deslizamentos em função da inclinação da encosta e da espessura do solo (modificado de D'Odorico & Fagherazzi, 2003).

Deve-se estabelecer o máximo ângulo de inclinação β_0 , em que o talude é sempre estável, em decorrência da necessidade para a deflagração do deslizamento de uma camada saturada maior que a espessura do solo. Enquanto $\beta < \beta_0$, o talude permanece estável. Para a definição de β_0 , deve-se atribuir à Eq. 2.84 que $c' = 0$ e $Z_w = Z$, resultando em:

$$\beta_0 = \tan^{-1} \left[\tan \phi \cdot \left(1 - \frac{\gamma_w}{\gamma} \right) \right] \quad (2.89)$$

O fluxograma relativo ao módulo geotécnico empregado encontra-se representado pela Fig. 2.16. Aristizábal (2014) ressalta também que há a necessidade de incorporação de fatores de correção ao modelo como parte do input, os quais, conforme Francés et al. (2007), consideram os efeitos do tempo e da escala espacial, assim como os erros do modelo e dos dados de entrada, deixando as características hidrológicas livres destes problemas, além de manter o significado físico dos parâmetros. A Tab. 2.5 traz o intervalo de variação dos parâmetros de correção recomendados por Vélez (2001) e Francés et al. (2007).

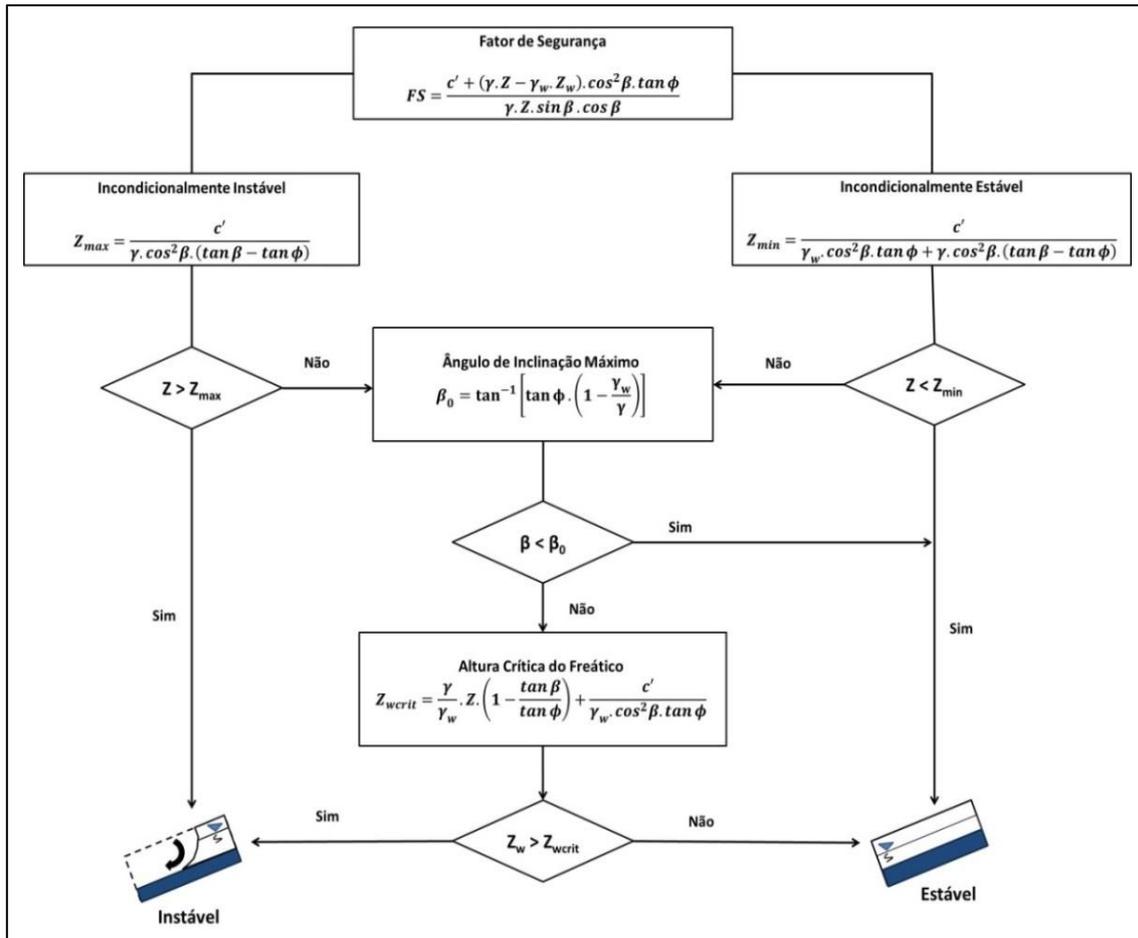


Figura 2.16. Módulo geotécnico do modelo SHIA_Landslide (modificado de Aristizábal, 2014).

Tabela 2.5. Parâmetros de correção e intervalos de valores definidos por Vélez (2001) e Francés et al. (2007) (modificado de Aristizábal, 2014).

Parâmetro de célula	Correção	Decomposição	Cx MIN	Cx MAX
Armazenamento estático máximo (S_{1max})	C_1	$S_{1max}^* = C_1 \cdot S_{1max}$	0,1	1,5
Evapotranspiração (EVP)	C_2	$EVP^* = C_2 \cdot EVP$	0,5	2,0
Capacidade de infiltração (K_s)	C_3	$K_s^* = C_3 \cdot K_s$	0,0	1,0
Capacidade de percolação (K_p)	C_4	$K_p^* = C_4 \cdot K_p$	0,0	2,0
Capacidade de fluxo subterrâneo (K_{pp})	C_5	$K_{pp}^* = C_5 \cdot K_{pp}$	0,0	10
Velocidade de fluxo superficial (V_2)	C_6	$V_2^* = C_6 \cdot V_2$	0,1	2,0

Parâmetro de célula	Correção	Decomposição	Cx MIN	Cx MAX
Velocidade subsuperficial (V_3)	C_7	$V_3^* = C_7 \cdot V_3$	1,0	1000
Velocidade do fluxo de base (V_4)	C_8	$V_4^* = C_8 \cdot V_4$	1,0	1000
Velocidade de canal (V_5)	C_9	$V_5^* = C_9 \cdot V_5$	0,5	1,5
Coesão (c')	C_{10}	$c'^* = C_{10} \cdot c'$	-	-
Ângulo de atrito (ϕ)	C_{11}	$\phi^* = C_{11} \cdot \phi'$	-	-
Espessura do solo (Z_s)	C_{12}	$Z_s^* = C_{12} \cdot Z_s$	-	-
Armazenamento gravitacional (S_{3max})	C_{13}	$S_{3max}^* = C_{13} \cdot S_{3max}$	0,1	1,5
Fator de segurança (FS)	C_{14}	$FS^* = C_{14} \cdot FS$	-	-

2.4.1.3 MODELO SLIDE PARA SISTEMA DE ALERTA DE DESLIZAMENTOS PROVOCADOS POR CHUVAS

O artigo técnico conferido a Liao et al. (2010) configura-se como a próxima referência incorporada ao trabalho que se relaciona com a análise de ameaça. No trabalho citado, é proposto um protótipo para um sistema de alerta para previsão de escorregamentos rasos deflagrados pela ação de precipitações, voltado para a ilha de Java, na Indonésia, com base em um modelo de base física, denominado SLIDE (SLope – Infiltration – Distributed Equilibrium).

O modelo SLIDE procede de uma modificação realizada no trabalho proposto por Montrasio & Valentino (2008), os quais, baseando-se no método do equilíbrio limite referente a taludes infinitos, consideram hipóteses simplificadas para o fluxo subsuperficial, definindo uma correlação direta entre o fator de segurança das encostas e a quantidade de precipitação. Segundo, ainda, Montrasio & Valentino (2008), esta formulação permite descrever os principais fatores que influenciam a inicialização do movimento, sem que haja a necessidade da incorporação de muitos parâmetros. Para verificar a capacidade do modelo em prever o momento em que escorregamentos rasos ocasionados por efeito das chuvas são detonados, Montrasio & Valentino (2008) empregaram a formulação em um processo de retroanálise em estudos de casos situados no norte da Itália.

Voltando ao trabalho de Liao et al. (2010), o sistema de alerta é composto pela integração de três componentes principais:

- Mapeamento e a identificação da susceptibilidade a partir de um conjunto de dados geoespaciais, relacionados à superfície do terreno (informação topográfica, mapas de propriedades do solo, grupo de informações originárias de um inventário local de deslizamentos, entre outros);
- Um sistema de monitoramento e estudo de precipitações por satélite e um modelo de previsão de precipitação. O primeiro é representado pela Tropical Rainfall Measuring Mission (TRMM), realizada de forma conjunta entre a National Aeronautics and Space Administration (NASA) e a Agência de Exploração Aeroespacial Japonesa, enquanto o segundo corresponde ao Weather Research Forecast Model, um sistema de previsão numérica do tempo;
- Um modelo de previsão de deslizamentos deflagrados por chuvas, denominado SLIDE.

A terceira componente do sistema é, de fato, aquela responsável pela execução das análises de estabilidade e a que foi devidamente efetivada na pesquisa atual. Para desenvolver o modelo SLIDE, Liao et al. (2010) apoiaram-se nos estudos de Fredlund et al. (1996) e Montrasio & Valentino (2008), os quais apresentam uma formulação matemática que traduz o fenômeno físico para os processos desencadeados por eventos de chuva. A influência imposta à resistência ao cisalhamento dos solos, decorrente tanto das forças desestabilizadoras provocadas pelo fluxo subsuperficial quanto da contribuição da saturação parcial, é destacada por este modelo, conforme indicado na Fig. 2.17.

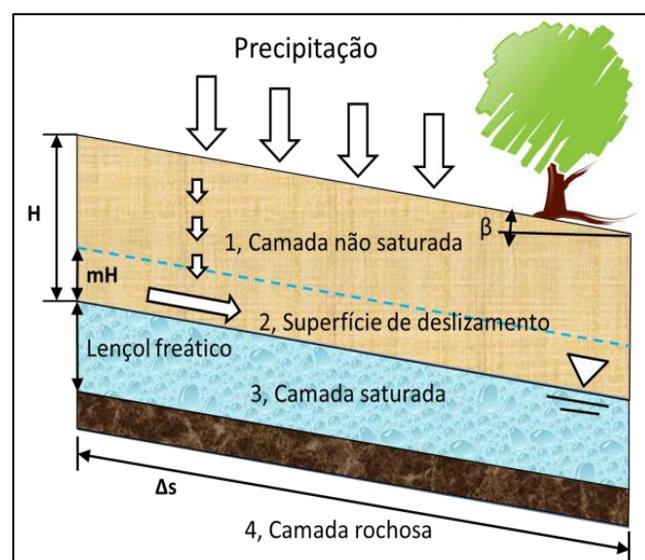


Figura 2.17. Esquema representativo para o modelo SLIDE de talude infinito (modificado de Liao et al., 2010).

O modelo proporciona a incorporação da ação da quantidade de precipitação na estabilidade da encosta, fornecendo a relação entre esta variável e a expressão que fornece o valor do fator de segurança, por meio de uma formulação matemática simples, a qual é expressa por:

$$FS = \frac{\cot \beta \cdot \tan \phi' [\Gamma + m(n_w - 1)] + C' \Omega}{\Gamma + m \cdot n_w} \quad (2.90)$$

Na Eq. 2.90, β é o ângulo de inclinação do talude e ϕ' refere-se ao ângulo de atrito efetivo do solo, ambos expressos em graus. Destacam-se, ainda, as funções integrantes do fator de segurança nas equações seguintes:

$$\Gamma = G_s \cdot (1 - n) \cdot n \cdot S_r \quad (2.91)$$

$$n_w = n \cdot (1 - S_r) \quad (2.92)$$

$$\Omega = \frac{2}{\sin(2\beta) \cdot H \cdot \gamma_w} \quad (2.93)$$

Nestas fórmulas, G_s é a gravidade específica, adimensional, n relaciona-se à porosidade, S_r é o grau de saturação, H é a espessura do solo e γ_w corresponde ao peso específico da água, igual a 9,8 kN/m³. Na expressão para o fator de segurança, C' representa a coesão total, que inclui a presença da coesão efetiva e da coesão aparente, esta última ligada à sucção mátrica indicada por Fredlund et al. (1996). A coesão total é dada pela expressão seguinte:

$$C' = [c' + c_\phi] \Delta s = [c' + A \cdot (1 - \lambda \cdot m^\alpha)] \Delta s \quad (2.94)$$

Onde c' corresponde à coesão efetiva e Δs é tomado como comprimento unitário do talude. O fator A representa um parâmetro dependente do tipo de solo e do pico da tensão cisalhante na ruptura, λ é um coeficiente de intensidade relacionado ao tipo de solo e α mostra-se como um parâmetro que retrata a tendência não linear da curva de coesão, sendo as três variáveis adimensionais.

Conforme Montrasio & Valentino (2008), o parâmetro m está ligado à espessura adimensional da parte saturada pertencente à camada, variando de 0 a 1. A espessura da camada saturada (mH) pode ser escrita como função da intensidade da precipitação. Liao et al. (2010) definem a grade composta pelo talude infinito como um tanque de equilíbrio hídrico que leva em consideração, de maneira simultânea, tanto o ganho de água pela

infiltração das chuvas e pela percolação, quanto a sua perda em virtude do fluxo de saída e da evapotranspiração através da rede. Segundo ainda Liao et al. (2010), testes de determinação do nível do lençol freático para o estabelecimento do valor inicial de m podem ser utilizados. A equação abaixo é, então, aplicada para o cálculo do balanço hídrico em cada passo de tempo requerido pelas análises:

$$\begin{cases} m_1 = 0 \\ O_t = K_t \cdot \sin \beta \cdot m_t \cdot H \cdot \cos \beta \cdot \Delta t \\ \Delta m_t = \frac{(I_t - O_t)}{n \cdot H \cdot (1 - S_r)} \\ m_{t+1} = m_t + \Delta m_t \end{cases} \quad (2.95)$$

Nestas equações, o índice t indica o passo de tempo em que se encontra a análise, Δt refere-se ao intervalo de tempo utilizado, m_1 corresponde ao valor inicial do parâmetro m e m_t é calculado em cada passo de tempo. O_t representa a saída de água de uma porção finita para um talude de comprimento finito L . A intensidade da chuva é dada por I_t , enquanto que capacidade global de drenagem decorrente da permeabilidade intrínseca do solo e da existência de numerosos caminhos preferenciais de fluxo corresponde ao termo K_t . O conjunto de parâmetros de entrada para o modelo e as respectivas unidades utilizadas por Liao et al. (2010) em seu trabalho estão resumidos na Tab. 2.6.

Tabela 2.6. Parâmetros e símbolos que são usados para a avaliação do modelo (modificado de Liao et al., 2010).

Propriedade	Símbolo	Unidade
Propriedades da encosta		
Declividade do talude	β	graus
Profundidade do deslizamento (nível freático), vertical	H	m
Espessura adimensional inicial	m_1	adimensional
Comprimento unitário do talude	Δs	1
Propriedades do solo		
Coefficientes	A e λ	adimensional
Ângulo de atrito efetivo	ϕ'	graus
Coesão efetiva	c'	kPa

Propriedade	Símbolo	Unidade
Coeficiente de coesão	α	adimensional
Gravidade específica	G_s	N/m^3
Porosidade	n	1
Grau de saturação	S_r	%
Peso específico da água	γ_w	N/m^3
Capacidade de drenagem	K_t	m/s
Propriedades da chuva		
Intensidade da chuva	I_t	mm/h
Duração da chuva	Δt	h

2.4.1.4 MODELAGEM MATEMÁTICA E NUMÉRICA UTILIZADAS EM ALERTAS DE DESLIZAMENTOS E FLUXO DE DETRITOS

A pesquisa conferida a Papa et al. (2011), com base no trabalho de Iverson (2000), cujo foco principal concentrou-se na criação de um sistema capaz de prover alertas acerca da ocorrência de fluxos de detritos em regiões desprovidas de uma base de dados histórica, ou no caso de situações em que mudanças climáticas e ambientais acarretam este tipo de evento, foi admitida, no âmbito da ameaça, para a inclusão no presente trabalho.

O trabalho fundamenta-se na simulação de um grande número de casos envolvendo todo intervalo que governa as variáveis de entrada ditas dinâmicas, ligadas à precipitação, considerando as variadas possibilidades de combinação entre elas. Conforme Papa et al. (2011), o volume total de detritos, disponível para o fluxo, é estimado para todas as combinações possíveis de intensidade de chuva, duração e precipitação antecedente.

A abordagem resulta na criação de um banco de dados, no sentido de obter curvas de limiares de precipitação, de forma que, ao serem operadas em tempo real, quando uma determinada medida de chuva, quer prevista quer medida, ultrapasse um dado limiar, pode-se estimar a correspondente probabilidade associada à ocorrência de fluxos de detritos. Assim, alertas a respeito de possíveis ocorrências de fluxos de detritos podem ser executados de maneira concordante com os resultados verificados.

A abordagem conferida a Papa et al. (2011) apoia-se em uma metodologia de simulação bem simplificada, demandando um curto período de tempo despendido ao processamento computacional e um conjunto limitado de insumos básicos. É conferido

destaque a estas características, uma vez que a simulação de uma grande quantidade de possíveis ocorrências de fluxo de detritos, dependente de variados valores de entrada de dados relacionados às propriedades da precipitação, é o pilar da abordagem proposta. Desta forma, é necessário que o modelo empregado para avaliar os possíveis casos de instabilidade deva ser extremamente rápido. Além disso, deve-se dar atenção à questão da disponibilidade restrita de informações em situações reais.

Segundo Papa et al. (2011), fez-se uso de um modelo matemático pautado na análise de estabilidade de taludes infinitos, conforme o trabalho de Iverson (2000), o qual simula a possível deflagração das corridas de detritos em um elemento genérico da bacia em estudo. O cômputo do fator de segurança, indicado por FS, em qualquer tempo (t) e para qualquer profundidade a partir da superfície (Z), é fornecido pela seguinte expressão:

$$FS(Z,t) = \frac{\tan \varphi}{\tan \alpha} + \frac{c - \psi(Z,t) \cdot \gamma_w \cdot \tan \varphi}{\gamma_s \cdot Z \cdot \sin \alpha \cdot \cos \alpha} \quad (2.96)$$

Nesta equação, α é a inclinação do talude, c refere-se à coesão do solo, Z é a coordenada vertical orientada positivamente para baixo, φ relaciona-se ao ângulo de atrito interno, γ_s é o peso específico do solo, γ_w corresponde ao peso específico da água, igual a 9,8 kN/m³, e $\psi(Z,t)$ é a carga hidráulica total, sendo esta dependente da profundidade e do tempo. O solo, ao longo da profundidade Z , é considerado instável quando o fator de segurança atinge um valor considerado crítico (por exemplo, FS = 1).

Papa et al. (2011) citam trabalhos, como Wieczorek & Glade (2005), nos quais se observaram muitos episódios de corridas de detritos detonados pela ação de um longo período de precipitação com baixa intensidade, seguido por um curto limite de tempo com intensas chuvas. Tomando-se este fato como referência, a determinação da carga hidráulica total, considerada como desencadeadora do processo, é realizada pela sobreposição dos efeitos da chuva com menor intensidade, considerada como antecedente, e o evento pluviométrico mais concentrado. A carga hidráulica total que corresponde à chuva antecedente é empregada como condição inicial para o cálculo ao longo do tempo da carga hidráulica total, relativa ao evento pluviométrico intenso.

Considera-se o estabelecimento do estado estacionário, assumindo-se que a direção do fluxo subterrâneo seja paralela ao talude, caso a precipitação antecedente tenha intensidade suficientemente baixa e longa duração. Nestas circunstâncias, aplica-se a equação abaixo para a obtenção da carga hidráulica concernente à água subterrânea:

$$\psi(Z,0) = (Z - d) \cdot (\cos \alpha)^2 \quad (2.97)$$

Onde d refere-se à profundidade do lençol freático medida na direção de Z , em condições estacionárias. A equação de conservação de massa para a água subterrânea, conforme Montgomery & Dietrich (1994), fornece:

$$Z_T - d = \frac{(I_z)_{\text{estacionário}}}{K_x} \cdot \frac{A}{b \cdot \sin \alpha \cdot \cos \alpha} \quad (2.98)$$

Na expressão acima, Z_T é a profundidade da camada impermeável, $(I_z)_{\text{estacionário}}$ é a taxa de infiltração na superfície do solo em condições estacionárias e na direção normal ao talude, a qual pode ser considerada como a chuva antecedente, K_x corresponde à condutividade hidráulica na direção paralela ao talude, A é a bacia drenada e b relaciona-se à largura do elemento da encosta ao longo da direção tangente ao contorno topográfico local.

Papa et al. (2011) optaram em adotar o enfoque oferecido por Iverson (2000), que propõe uma resolução analítica da equação de Richards, voltada para o cálculo da pressão resultante do período curto de precipitação.

Tomando-se as hipóteses de infiltração vertical e as condições iniciais de umidade, bem como as condições de contorno definidas por um fluxo vertical subterrâneo nulo para grandes profundidades abaixo do nível freático e a entrada de água na superfície do solo regida pela lei de Darcy, a expressão a seguir traz a solução indicada por Iverson (2000):

$$\psi(Z,T) = \psi(Z,0) + Z \cdot \frac{I_z}{K_z} \cdot [R(T^*)] \quad (2.99)$$

Em que $\psi(Z,0)$ corresponde à carga hidráulica no início do evento pluviométrico, I_z é a razão de infiltração na superfície do solo, em uma direção normal ao talude, relacionando-se com a intensidade da precipitação que deflagra os eventos de corridas de detritos, K_z refere-se a condutividade hidráulica em relação à direção normal da encosta e a parcela $R(T^*)$ é expressa da seguinte forma:

$$R(T^*) = \sqrt{\frac{T^*}{\pi}} \exp\left(\frac{-1}{T^*}\right) - \operatorname{erfc}\left(\frac{1}{\sqrt{T^*}}\right) \quad (2.100)$$

Onde erfc representa a função de erro complementar, assim definida:

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt \quad (2.101)$$

O termo T^* é dado pela equação seguinte:

$$T^* = \left[\frac{T}{\frac{Z^2}{4.D_0(\cos \alpha)^2}} \right] \quad (2.102)$$

Em que T relaciona-se à duração do evento de chuva e D_0 é a difusividade máxima característica, a qual governa a transmissão da carga hidráulica quando o solo está próximo da saturação.

Papa et al. (2011) realizaram uma divisão das variáveis de entrada em duas classes principais, categorizando-as em estáticas e dinâmicas. O modelo matemático proposto tem a capacidade de avaliar um provável caso de instabilidade relacionado a um elemento da bacia com determinadas características (variáveis de entrada estáticas), sujeitas a uma dada precipitação (variáveis de entrada dinâmicas).

Na Tab. 2.7 encontram-se relacionadas à listagem das variáveis, sua classe e as unidades em que devem ser incorporadas ao modelo, bem como uma breve descrição de cada uma.

Tabela 2.7. Listagem das variáveis classificadas em dinâmicas e estáticas (modificado de Papa et al., 2011).

Classificação	nº	Variável	Unidade	Descrição
Dinâmica	1	T	h	Duração do evento de chuva
	2	I_z	mm/h	Taxa de infiltração da chuva na superfície do terreno, na direção normal do talude
	3	$(I_z)_{\text{estacionário}}$	mm/mês	Taxa de infiltração da chuva antecedente na superfície do terreno, na direção normal do talude
Estática	4	A/b	m	Razão entre a bacia drenada e a largura do elemento do talude ao longo da direção tangente ao contorno topográfico local

Classificação	n°	Variável	Unidade	Descrição
Estática	5	Z_T	m	Profundidade total disponível do solo – profundidade da camada rígida impermeável
	6	γ_s	kN/m ³	Peso específico médio do solo
	7	ϕ	Graus (°)	Ângulo de atrito interno
	8	c	kPa	Coesão do solo
	9	α	Graus (°)	Declividade do talude
	10	K_x	mm/s	Condutividade hidráulica na direção paralela ao talude
	11	K_z	mm/s	Condutividade hidráulica na direção na direção normal ao talude
	12	D_0	m ² /s	Difusividade máxima característica que governa a transmissão da carga hidráulica quando o solo está próximo da saturação

A proposta original montada por Papa et al. (2011) contempla apenas uma quantidade determinada de elementos computacionais, que devem ser incorporados ao modelo e que sejam capazes de representar o comportamento completo da bacia analisada. Segundo Papa et al. (2011), torna-se impraticável a análise de estabilidade para todas as células que compõem a bacia pelo sistema de alerta, em virtude do tempo dispensado pelo modelo e da necessidade de informações detalhadas sobre as propriedades dos materiais envolvidos.

Desta forma, a simulação descrita anteriormente é executada para certa quantidade (n) de elementos computacionais que podem representar o comportamento de toda a bacia, no intuito de avaliar se esta bacia específica tem a capacidade de sediar a formação de corridas de detritos. Os elementos computacionais usados condizem a entidades virtuais definidas por um vetor de variáveis, não correspondendo a células reais atreladas a uma determinada posição física composta por coordenadas geográficas.

A organização para os dados de entrada se caracteriza, então, em uma matriz de variáveis estáticas que representam a bacia analisada. O procedimento para estruturação desta matriz corresponde à divisão da bacia em distritos, homogêneos do ponto de vista dos atributos geomorfológicos, possuindo, portanto, os mesmos valores para o conjunto de

variáveis do solo: Z_T , γ_s , ϕ , c , K_x e K_z . A difusividade máxima característica D_0 tem a possibilidade de assumir qualquer valor positivo, dependendo da proximidade em relação à saturação completa, de forma que, baseando-se nesta consideração, D_0/K_z pode ser usado como um parâmetro de calibração.

Nesta metodologia estabelecida por Papa et al. (2011), considera-se a incerteza na avaliação das variáveis do solo por meio da atribuição, a cada variável, do seu valor médio associado a um intervalo de confiança. A definição do intervalo de confiança para cada variável é dependente das metodologias usadas para a avaliação da variável analisada e das incertezas atreladas aos seus dados. Quando se realiza a atribuição de uma variável específica do solo para um determinado número de vetores de entrada, isto ocorre mediante uma distribuição normal de probabilidade para a variável em questão, a qual é definida a partir de sua média e seu intervalo de confiança.

As variáveis topográficas α e A/b são derivadas para cada célula pertencente à região estudada, a partir da aplicação ao modelo numérico de terreno, de ferramentas incorporadas ao SIG, sendo que seus valores são atribuídos a cada distrito, obedecendo à frequência de distribuição dos valores da bacia real.

Papa et al. (2011) definem, também, que a quantidade n de elementos a serem aplicados às análises pode ser estabelecida como um parâmetro de entrada, atuando diretamente, tanto no tempo de processamento, quanto na representatividade da matriz de entrada.

Em se tratando da atribuição para as variáveis de entrada dinâmicas T , I_z e $(I_z)_{\text{estacionário}}$, deve-se estabelecer os limites inferior e superior do intervalo de valores possíveis para as variáveis, assim como a quantidade total de valores para uma dada variável específica, m_i , com i assumindo os valores 1, 2 e 3. Procede-se a simulação para cada linha da matriz de input estática, que representa um distrito homogêneo, em conjunto com cada combinação para as variáveis de entradas dinâmicas, de forma que o número total de simulações executadas (N) pode ser representado por:

$$N = n \cdot \prod_{i=1}^3 m_i \quad (2.103)$$

A saída de dados fornecida a cada simulação corresponde ao número de elementos instáveis e ao volume correspondente de detritos disponíveis. A partir dos dados gerados, por interpolação simples, é possível estabelecer, em relação a cada valor de chuva antecedente

(I_z)_{estacionário}, um gráfico representando curvas em função da intensidade e duração das chuvas que reproduzem um valor fixo de porcentagem de células instáveis ou um valor fixo de volume total de detritos.

2.4.2 TRABALHOS USADOS NA MODELAGEM DE CARACTERÍSTICAS DE INTENSIDADE DA AMEAÇA ASSOCIADA A DESLIZAMENTOS

Destacam-se, agora, as pesquisas que serviram de embasamento teórico no desenvolvimento da metodologia voltada para a simulação de características relacionadas à intensidade da ameaça, como a definição da trajetória e do alcance dos eventos, além das estimativas para a velocidade dos escorregamentos e para as áreas de deposição de detritos.

2.4.2.1 SIMULAÇÃO DAS TRAJETÓRIAS DE DESLIZAMENTO PELO USO DE AUTÔMATOS CELULARES

Apoiou-se no trabalho de Lai & Dragičević (2011) para estabelecer a ferramenta que simula a dinâmica do processo de propagação dos deslizamentos, baseando-se no uso de autômatos celulares capazes de representar componentes espaciais e temporais do problema.

A abordagem proposta tem por objetivo, ser uma alternativa às práticas comuns de pesquisas nesta área, que têm como foco o uso de equações diferenciais para explicar o fluxo dinâmico dos escorregamentos e que, frequentemente, mostra-se trabalhoso, caso não sejam efetuadas as devidas simplificações.

Lai & Dragičević (2011) destacam que o conjunto de insumos iniciais, requeridos por muitos modelos de análise de deslizamentos que utilizam plataformas SIG, devem ser detalhados e conseguidos, preferencialmente, a partir de trabalhos de campo. Além de poder acarretar altos custos em relação à realização da pesquisa, a disponibilidade de dados não minimiza a dificuldade de implementação de métodos de base física voltados para a estabilidade de taludes. Como possível solução para isto, o trabalho propõe uma metodologia com característica de baixo custo para a previsão das áreas de deposição de deslizamentos, sem que haja a necessidade de aquisição de informações refinadas na aplicação do modelo físico determinístico.

O modelo é formado pela conjugação de uma regressão logística e uma teoria pautada em autômatos celulares, incorporando informações de alta resolução espacial e dados digitais de elevação, devidamente armazenados em um ambiente SIG, conforme mostrado pela Fig. 2.18.

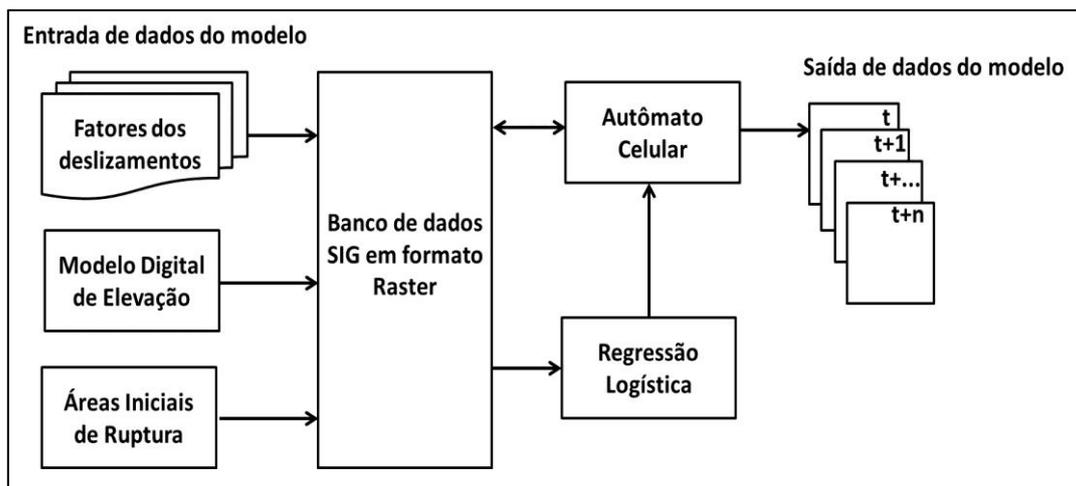


Figura 2.18. Interação entre os componentes do modelo (modificado de Lai & Dragičević, 2011).

Além do modelo numérico de terreno, verifica-se, a partir da Fig. 2.18, que a entrada de dados requer os fatores classificados como detonantes dos escorregamentos, assim como as localizações iniciais das rupturas. As regras de transição presentes nos autômatos celulares são propostas a partir do processo de regressão logística, que determinam os fatores que mais influenciam na ocorrência dos eventos.

A abordagem por intermédio de autômatos celulares é composta por cinco partes principais que formam sua estrutura:

- Grade, definida como a região do local de estudo que limita onde o fenômeno se desenvolve. A resolução espacial conferida a esta matriz dita a distância que o deslizamento percorrerá quando ocorre a propagação do evento para uma célula vazia;
- Vizinhança, a qual foi empregada uma vizinhança de Moore de 3 por 3 células, onde cada autômato celular é cercado por oito vizinhos, conforme a Fig. 2.19. Entretanto, Lai & Dragičević (2011) explicam que alguns vizinhos podem ser desconsiderados, desde que sua influência para o autômato possa ser desconsiderada com base na posição. Exemplo disso é o que ocorre no caso de estudo exposto no trabalho. Uma vez que a área avaliada, situada ao norte de Vancouver, Canadá, possui vertentes na direção sudoeste, os vizinhos que mais interferem no processo são aqueles com maiores altitudes localizados a nordeste. Assim sendo, conforme a Fig. 2.20, os vizinhos mais baixos que a célula central, com as coordenadas $[-1,0]$, $[-1,-1]$ e $[0,-1]$, em preto, não desempenham qualquer importância, podendo ser desconsiderados.

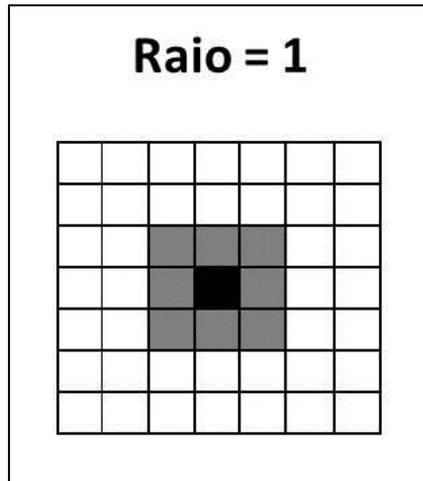


Figura 2.19. Vizinhança de Moore utilizada nos autômatos celulares por Lai & Dragičević (2011).

-1, 1	0, 1	1, 1
	0, 0	1, 0
		1, -1

Figura 2.20. Vizinhança de Moore modificada para uma exposição de vertentes com direção sudoeste (modificado de Lai & Dragičević, 2011).

- Conjunto finito de estados que cada célula do autômato celular pode assumir, que, neste caso, corresponde à presença ou não de escorregamentos de terra;
- Regras de transição, referentes às funções determinísticas que governam a evolução do processo. Estas leis derivam de uma abordagem empírica ou de base física, sendo que a regra principal, no caso dos movimentos de massa, segundo Claessens et al. (2007), indica que a frente de deslizamento se propaga na direção das altitudes mais baixas e com maiores inclinações. A restrição do fluxo baseia-se no fato de que o movimento de subida não ocorre. Assim sendo, o fluxo de escorregamento não será transportado de uma célula de menor elevação para uma com maior altimetria. O critério de parada adotado para o fluxo segue estudos empíricos desenvolvidos por Benda & Cundy (1990), os quais determinam que isto deva ocorrer para declividades inferiores a 5°.

Acessando as informações de elevação da área de estudo por meio da grade que contém o modelo numérico de terreno, o autômato celular verifica para qual célula o deslizamento será propagado.

- Iteração ou intervalo de tempo, representando uma única aplicação das regras de transição com a consequente atualização da área do deslizamento. Em cada iteração, as leis são aplicadas de tal modo que todas as células da malha são atualizadas ao mesmo tempo. Lai & Dragičević (2011) ressaltam que, para o início da simulação, é necessário introduzir, junto ao conjunto de dados geoespaciais, uma configuração inicial contendo os polígonos que servirão de pontos de ruptura dos deslizamentos.

Antes da execução das simulações por meio dos autômatos celulares, faz-se uso da regressão logística, cujo objetivo refere-se ao estabelecimento do melhor ajuste, capaz de descrever a relação entre a presença ou ausência de escorregamentos de terra (variável dependente) e parâmetros geométricos extraídos do modelo numérico de terreno (variáveis independentes), os quais irão compor as regras de transição presentes nos autômatos celulares. O ajuste da variável dependente, por intermédio da regressão logística, emprega a seguinte equação:

$$Y = \log it(p) = \ln\left(\frac{p}{1-p}\right) = B_0 + B_1 \cdot X_1 + B_2 \cdot X_2 + \dots + B_n \cdot X_n \quad (2.104)$$

Onde p é a probabilidade da variável dependente (Y) ser igual a 1, $p/(1-p)$ é a razão de chance, B_0 é o intercepto e B_1, B_2, \dots, B_n correspondem aos coeficientes de regressão, os quais medem, respectivamente, a contribuição dos fatores independentes X_1, X_2, \dots, X_n na variação de Y .

Os pesos dos coeficientes resultantes do ajuste dos parâmetros previsto pela equação acima representam o grau de influência na previsão dos eventos de deslizamentos.

Os parâmetros topográficos derivados do modelo numérico de terreno podem contribuir ou impedir o progresso do deslizamento, de forma que o grau de influência de cada fator no ajuste da própria grade do Modelo Numérico de Terreno (MNT) depende dos pesos atribuídos aos respectivos coeficientes. A Tab. 2.8 apresenta os resultados gerados pela regressão logística empreendida por Lai & Dragičević (2011).

Tabela 2.8. Coeficientes da regressão logística (modificado de Lai & Dragičević, 2011).

Variáveis independentes	Coeficiente
Declividades 0-5°	+0.002
Declividades 6-10°	+0.019
Declividades 11-15°	-0.063
Declividades 16-20°	+0.029
Declividades 21-25°	+0.081
Declividades 26-30°	+0.070
Declividades 31-35°	-0.071
Declividades 36-40°	+0.043
Declividades 41-45°	-0.023
Declividades 46-50°	+0.019
Encosta côncava	+0.204
Encosta convexa	-0.053
Inflexão da vertente	-0.006
Encosta em sela	+0.206
Talvegue	-0.047
Cumeada	+0.121
Intercepto	-0.983

Observou-se que as curvaturas côncavas e encostas sob o formato de selas foram os parâmetros que mais influenciaram positivamente o processo, ao passo que o coeficiente da regressão com tendências mais negativas correspondeu a características de encostas convexas. Quando os parâmetros têm pesos positivos e agem, portanto, contribuindo para a propagação do evento, o MNT será alterado pela diminuição do valor da altimetria das células, onde estas características são encontradas, ampliando, portanto, as chances do fluxo seguir nesta direção. No caso em que os pesos são negativos, os parâmetros correspondentes impedem o fluxo, elevam a altimetria nestas áreas e diminuem a probabilidade de ocorrência do deslizamento.

Desta forma, combinaram-se aspectos das regras de transição, a prevenção quanto à não ocorrência do fluxo encosta acima e critérios de parada com o ajustamento das elevações que atuam no desencadeamento de eventos de deslizamentos, conforme expresso pelo fluxograma da Fig. 2.21.

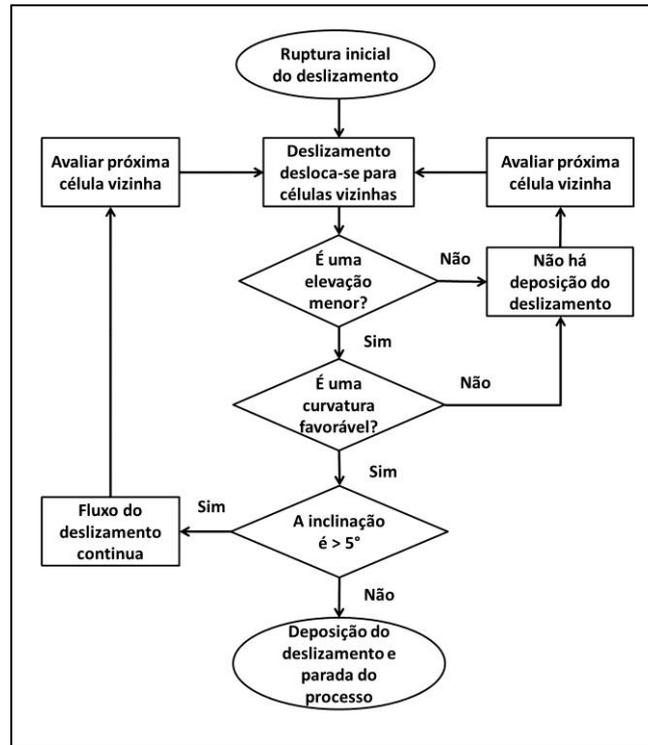


Figura 2.21. Fluxograma da modelagem do processo com base nas características topográficas derivadas do MNT e se a curvatura é favorável ao deslizamento (modificado de Lai & Dragičević, 2011).

2.4.2.2 AVALIAÇÃO DA ESPESSURA DE DETRITOS PELO USO DE ALGORITMOS DE FLUXO PARA MÚLTIPLOS VIZINHOS

Conforme Gruber & Peckham (2009), o efeito de fluxos divergentes, importantes principalmente em encostas, pode ser representado apenas por meio de métodos de fluxo para múltiplos vizinhos. A Fig. 2.22 traz a representação para as direções múltiplas de fluxo correspondente ao espalhamento a partir de uma célula para várias mais baixas.

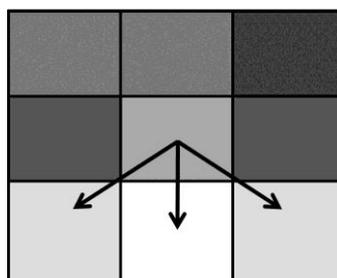


Figura 2.22. Direções múltiplas de fluxo designadas para um pixel central com uma vizinhança 3x3 (modificado de Gruber & Peckham, 2009).

Na Fig. 2.22, a altimetria do terreno está representada em uma escala de nível de cinza, sendo que a tendência crescente da elevação relaciona-se ao tom mais escuro do pixel. Uma fração da massa pertencente ao pixel central é distribuída para cada uma das três células mais baixas, para onde as setas indicam, de forma que a soma de todas as frações distribuídas devem corresponder a um.

Os algoritmos que têm função de controlar o fluxo divergente e sua distribuição por uma vizinhança são frequentemente referenciados por métodos MFD (multiple flow direction), como FD8 (Freeman, 1991) ou TOPMODEL (Quinn et al., 1991). Gruber & Peckham (2009) destacam que dentro de uma formulação genérica, a fração de drenagem d presente em um pixel vizinho NB_i pode ser dada da seguinte forma:

$$d_{NB_i} = \frac{\tan(\beta_{NB_i})^v \cdot L_{NB_i}}{\sum_{j=1}^8 [\tan(\beta_{NB_j})^v \cdot L_{NB_j}]} \quad (2.105)$$

A declividade β (considerada positiva para células mais baixas, e nula nos pixels mais elevados) nas células vizinhas, em diferentes comprimentos de contorno L de drenagem, tem influência na fração d , assim como o expoente v , que controla a dispersão. No sentido de preservar a massa, os potenciais de drenagem em cada vizinho são normalizados para a unidade sobre uma matriz 3x3, podendo-se, ainda, empregar diferentes pesos a células com menores altitudes, entre as quais o fluxo se distribui.

Gruber & Peckham (2009) também citam que o fluxo torna-se concentrado na direção das maiores inclinações para valores altos de v , ao passo que baixas medidas deste parâmetro resultam em uma forte dispersão ($v \geq 0$). Valores de v , situando-se de 4 a 6 com L , igual tanto para as direções cardinais quanto as diagonais, produzem os melhores resultados, segundo Holmgren (1994). O método torna-se flexível pelo emprego deste expoente, trazendo a dificuldade, por outro lado, de determinar as medidas consideradas ideais para o termo em questão.

2.4.2.3 ESTIMATIVA DA VELOCIDADE DO FLUXO DE DESLIZAMENTO PELO USO DE MODELO BASEADO NA TEORIA DO ADENSAMENTO

Como componente integrante na quantificação da vulnerabilidade de estruturas localizadas no interior de escorregamentos de terra pela teoria empregada, encontra-se o dado relativo à velocidade da massa deslizando, no sentido de representar a intensidade da ameaça

nestes casos. Assim, fez uso do trabalho apresentado por Qiao & Clayton (2013), que estabelece expressões para a poropressão, velocidade e distância percorrida para o fluxo deslizando com base em um modelo de deslizamento fundamentado na teoria do adensamento unidimensional de Terzaghi. As quantificações físicas destes termos são codificadas por meio de funções matemáticas com o emprego da linguagem de programação Mathematica, em conjunto com a criação de interfaces gráficas que auxiliam a manipulação do modelo por parte do usuário.

Qiao & Clayton (2013) abordam, inicialmente, a formulação para o cálculo do excesso de poropressão por meio da teoria do adensamento unidimensional de Terzaghi (Terzaghi, 1936) em três casos distintos, correspondentes a seções de um estrato ideal de argila. Tratando-se das condições de contorno, a camada de argila encontra-se apoiada sobre uma base horizontal impermeável, com a drenagem livre no topo. A Fig. 2.23 retrata os casos mencionados.

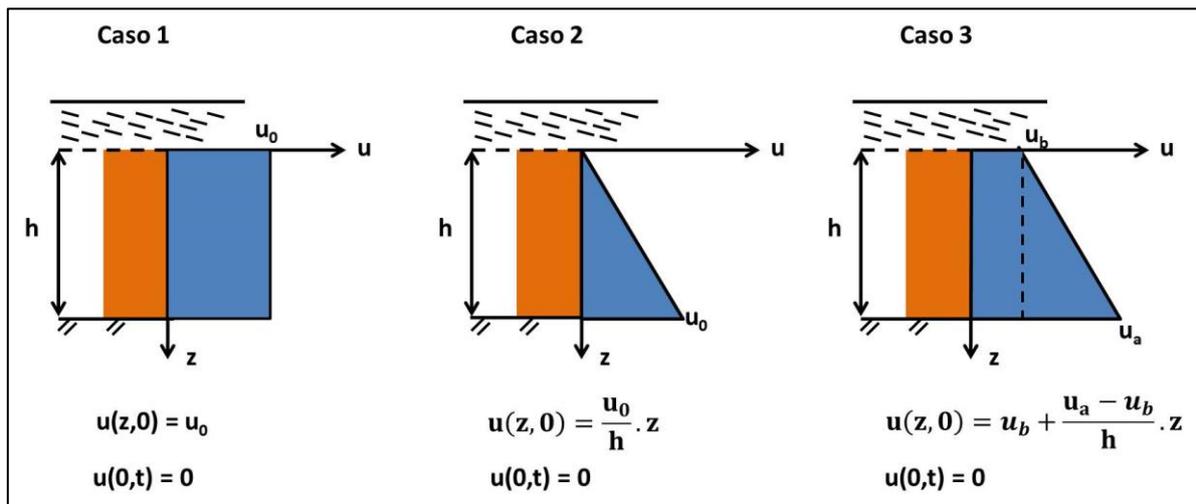


Figura 2.23. Configurações para o excesso de poropressão estudados (modificado de Qiao & Clayton, 2013).

As condições de contorno são as mesmas para os três casos, sendo que a diferença entre eles reside nas condições iniciais impostas. Para o caso 1, a teoria do adensamento de Terzaghi fornece a seguinte fórmula para o cálculo do excesso de poropressão $u_1(z,t)$ em uma determinada profundidade z da camada de argila e no tempo t :

$$u_1(z,t) = \sum_{n=1}^{\infty} \frac{4 \cdot u_0}{(2 \cdot n - 1) \cdot \pi} \sin \frac{(2n-1) \cdot \pi \cdot z}{2 \cdot h} \cdot \exp \left[- (2 \cdot n - 1)^2 \cdot \pi^2 \cdot C_v \cdot t / (4 \cdot h^2) \right] \quad (2.106)$$

Para o caso 2, o excesso de poropressão $u_2(z, t)$:

$$u_2(z,t) = \sum_{n=1}^{\infty} \left[-\frac{8.u_0}{(2.n-1)^2.\pi^2} \cos(n.\pi) \right] \sin \frac{(2.n-1).\pi.z}{2.h} \exp \left[-(2.n-1)^2 \pi^2 C_v.t / (4.h^2) \right] \quad (2.107)$$

Já o terceiro caso emprega a seguinte fórmula para o excesso de poropressão $u_3(z,t)$:

$$u_3(z,t) = \sum_{n=1}^{\infty} \left[\frac{4.u_b}{(2.n-1).\pi} - \frac{8(u_a - u_b)}{(2.n-1)^2.\pi^2} \cos(n.\pi) \right] \sin \frac{(2.n-1).\pi.z}{2.h} \exp \left[-(2.n-1)^2 \pi^2 C_v.t / (4.h^2) \right] \quad (2.108)$$

Nas equações anteriores, C_v corresponde ao coeficiente de adensamento do solo. A Eq. 2.108 pode ser utilizada para computar o excesso de poropressão para os três casos apresentados.

O modelo empregado por Qiao & Clayton (2013) foi mostrado pela primeira vez por Hutchinson (1986), cujo esquema encontra-se representado na Fig. 2.24.

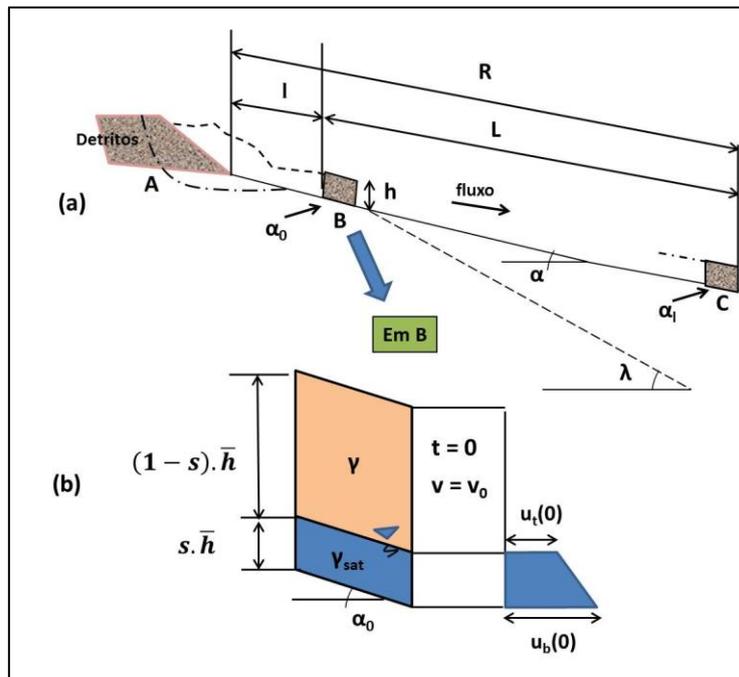


Figura 2.24. (a) Seção esquemática para o fluxo. (b) Configuração assumida para o excesso inicial de poropressão, derivado de Hutchinson (1986) (modificado de Qiao & Clayton, 2013).

A rotina computacional proposta é capaz de calcular a velocidade do deslizamento, o tempo e a poropressão em certa posição, podendo julgar automaticamente a localização e o instante que o fluxo irá parar dentro de uma encosta com diferentes configurações de ângulos de inclinação. Os detritos são compostos por uma massa solta e sem coesão, sendo saturados

na sua parte inferior. Ocorre a formação de uma lâmina de detritos após a ruptura do talude com espessura uniforme e comprimento de transição l . Concentra-se na avaliação do elemento principal da lâmina de detritos, cujo comportamento do movimento representa o fluxo de deslizamento. Destacado na parte (b) da Fig. 2.24, esse elemento possui dimensões expressas por $\bar{h} \times b$ (altura x comprimento), com uma camada basal saturada de altura $s\bar{h}$.

O desequilíbrio de forças que atuam no elemento principal decorrente da súbita perda de resistência ao cisalhamento na superfície de deslizamento, fazendo com que se mova com certa aceleração ao longo da encosta BC, que possui α como ângulo de inclinação principal. A resistência ao longo da superfície BC é dada por:

$$\tau = [m.g.\cos\alpha - u_b(t).b.\sec\alpha] \tan\phi' \quad (2.109)$$

Onde $u_b(t)$ é a poropressão basal no tempo t , ϕ' refere-se ao ângulo de atrito do solo na superfície BC do talude, enquanto o peso do elemento, produto da sua massa m pelo valor da aceleração da gravidade g , pode ser expresso por:

$$m.g = b.\bar{h} \cdot [(1-s)\gamma + s.\gamma_{sat}] \quad (2.110)$$

Nesta última equação, γ e γ_{sat} referem-se, respectivamente, ao peso específico natural e ao peso específico saturado do solo.

Pode-se considerar como condição inicial para o elemento principal B, a liquefação da parte saturada, total ou parcial, em função de uma carga não drenada súbita. O valor máximo para o excesso inicial de poropressão na base da camada impermeável $u_b(0)$ será:

$$(u_{b0})_{\max} = [(1-s)\gamma + s.\gamma_{sat}] \bar{h} \cdot \cos^2 \alpha_0 \quad (2.111)$$

De forma semelhante, a poropressão máxima inicial no topo da camada saturada é escrita como:

$$(u_{t0})_{\max} = (1-s)\gamma.\bar{h} \cdot \cos^2 \alpha_0 \quad (2.112)$$

No entanto, as poropressões iniciais reais não alcançam facilmente os valores máximos previstos, de forma que se utiliza o termo r_0 proposto por Hutchinson (1986) para representar a razão $u_b(0)/(u_{b0})_{\max}$ ou $u_t(0)/(u_{t0})_{\max}$, conforme as equações abaixo:

$$u_b(0) = r_0 \cdot (u_{b0})_{\max} \quad (2.113)$$

$$u_t(0) = r_0 \cdot (u_{t0})_{\max} \quad (2.114)$$

À medida que o elemento principal está em movimento, ocorre a dissipação da poropressão, com o conseqüente aumento da tensão efetiva, resultando, por fim, com que o elemento entre em repouso. A condição inicial para a dissipação da poropressão pode ser representada pelo caso 3, presente na Fig. 2.23, de maneira que, a partir da Eq. 2.108, ao se fazer $u_b = u_t(0)$ e $u_a = u_b(0)$, tem-se:

$$u_b(t) = \sum_{n=1}^{\infty} \frac{\sin \frac{\pi(2n-1)}{2} [4\pi(2n-1)u_b - 8(u_a - u_b)\cos(n\pi)] \exp \left[-\frac{\pi^2 C_v (2n-1)^2 t}{4h_s^2} \right]}{\pi^2 (2n-1)^2} \quad (2.115)$$

Nesta equação $h_s = s\bar{h}$, correspondendo ao comprimento do caminho de percolação da água.

A aceleração $a(t)$ do elemento pode ser obtida pelo desequilíbrio entre as forças que nele atuam:

$$a(t) = \frac{P \cdot \sin \alpha - \tau}{m} \quad (2.116)$$

Onde P refere-se ao peso do elemento, dado pela Eq. 2.110, e τ é a resistência expressa pela Eq. 2.109. Efetuando-se as correspondentes substituições e simplificações, tem-se:

$$a(t) = \frac{dv(t)}{dt} = g \left\{ \sin \alpha - \left(\cos \alpha - \frac{u_b(t)}{h \cdot \cos \alpha \cdot [(1-s)\gamma + s \cdot \gamma_{sat}]} \right) \tan \phi' \right\} \quad (2.117)$$

A velocidade $v(t)$ e a distância $x(t)$ do elemento são dadas por:

$$v(t) = v_0 + \int_0^t a(t) \cdot dt \quad (2.118)$$

$$x(t) = \int_0^t v(t) \cdot dt \quad (2.119)$$

O termo v_0 é a velocidade inicial atribuída ao elemento principal, quando este inicia seu movimento. Agora, fazendo:

$$a_1 = g[\sin \alpha - \cos \alpha \cdot \tan \phi'] \quad (2.120)$$

$$a_2 = \frac{g \cdot \sec \alpha \cdot \tan \phi'}{h \cdot [\gamma(1-s) + \gamma_{sat} \cdot s]} \quad (2.121)$$

$$a_3 = \frac{4 \sin\left[\frac{\pi}{2}(2n-1)\right] [\pi(2n-1)u_b - 2 \cos(\pi.n)(u_a - u_b)]}{\pi^2(2n-1)^2} \quad (2.122)$$

$$a_4 = -\frac{\pi^2 C_v (2n-1)^2}{4.h_s^2} \quad (2.123)$$

Assim:

$$u_b(t) = a_3 \sum_{n=1}^{\infty} \exp(a_4.t) \quad (2.124)$$

$$a(t) = a_2 \sum_{n=1}^{\infty} a_3 \cdot \exp(a_4.t) + a_1 \quad (2.125)$$

$$v(t) = \int_0^t a(t).dt + v_0 = a_2 \sum_{n=1}^{\infty} \frac{a_3 \cdot [\exp(a_4.t) - 1]}{a_4} + a_1.t + v_0 \quad (2.126)$$

$$x(t) = \int_0^t v(t).dt = a_2 \sum_{n=1}^{\infty} \frac{a_3 [-a_4.t + \exp(a_4.t) - 1]}{a_4^2} + \frac{1}{2}t(a_1.t + 2.v_0) \quad (2.127)$$

O somatório das séries infinitas contidas nas equações anteriores pode ser substituído pela soma de uma sequência $\{S_m\}$, onde S representa a expressão envolvida na série infinita. O valor m substitui o símbolo ∞ nas expressões e determina a precisão computacional imposta aos resultados. Qiao & Clayton (2013) ressaltam que o código computacional criado foi concebido para calcular as quantidades físicas de modo a ver a convergência destes resultados para distintos valores de m.

Pode-se separar a encosta em vários segmentos, no caso em que a superfície BC não é reta, sendo que cada divisão pode ser assumida como uma linha reta com sua própria inclinação (Fig. 2.25). Como cada segmento apresenta um comprimento e uma declividade próprios, é necessário verificar se o elemento principal irá parar ao longo do tramo, no qual está se movendo ou não. No caso do movimento não cessar, o tempo decorrente para percorrer o dado segmento, a velocidade e a aceleração no fim do segmento devem ser calculados para servirem de condições iniciais que serão impostas à próxima divisão, até que o movimento cesse em alguma das inclinações.

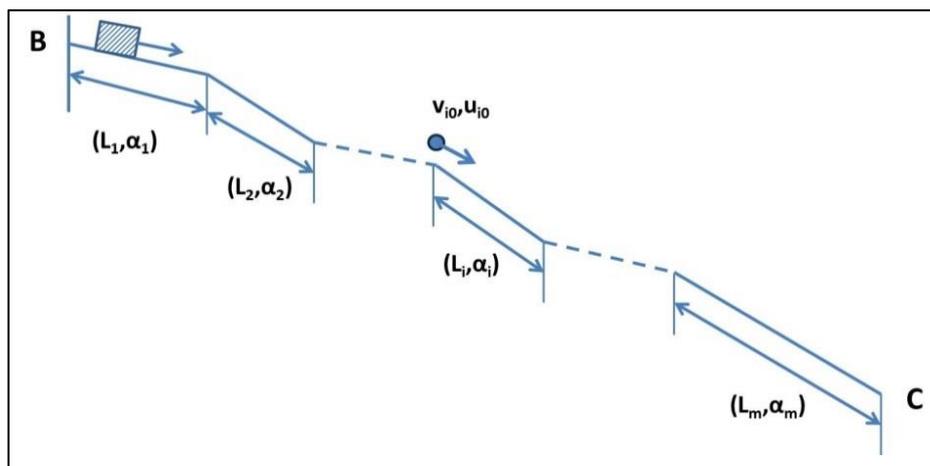


Figura 2.25. Esquema que representa o fluxo em uma seção do talude com diferentes ângulos de declividade (modificado de Qiao & Clayton, 2013).

2.4.3 MODELO PARA QUANTIFICAÇÃO DA VULNERABILIDADE

Basicamente, o método indicado por Li et al. (2010) norteou a análise da vulnerabilidade nesta pesquisa, culminando na implementação computacional do modelo proposto, o qual fornece a estimativa de um valor em termos médios, associado a este componente.

Conforme Fell et al. (2008a), a vulnerabilidade pode ser calculada em relação a qualquer elemento em risco (como população, edifícios, obras de engenharia, atividades econômicas, serviços de utilidades públicas e outras infraestruturas) presente em áreas potencialmente afetadas por ameaça de deslizamentos. Nesta pesquisa, considerou-se apenas a vulnerabilidade associada a edificações, embora o modelo quantitativo definido por Li et al. (2010) também propõe uma abordagem voltada para a definição desta componente associada a pessoas, mas que não foi considerada neste trabalho.

Conforme Li et al. (2010), a vulnerabilidade fica definida pela intensidade da ameaça, estando este vinculado aos elementos em risco expostos, e pela capacidade dos mesmos em resistirem a uma determinada ameaça. Assim, o valor da vulnerabilidade, representada por V , é fornecido pela aplicação da função apresentada por:

$$V = f(I, R) = \begin{cases} \frac{2I^2}{R^2} & \text{se } \frac{I}{R} \leq 0,5 \\ 1 - \frac{2(R-I)^2}{R^2} & \text{se } 0,5 < \frac{I}{R} \leq 1 \\ 1 & \text{se } \frac{I}{R} > 1 \end{cases} \quad (2.128)$$

Na equação anterior, I representa a intensidade da ameaça e R corresponde à resistência atribuída ao elemento em risco, sendo que ambos os fatores são adimensionais.

Por sua vez, os termos que compõe a vulnerabilidade também dependem de determinados fatores, os quais carregam consigo as características correspondentes à ocorrência de episódios de deslizamentos, trazendo, neste caso, informações sobre a ameaça, e aspectos concernentes às estruturas assentes na região analisada, os quais traduzem o desempenho de resistência destes elementos em risco.

Reproduz-se, na Fig. 2.26, a tendência de variação da função de vulnerabilidade com relação ao quociente I/R , enquanto que na Fig. 2.27 esta relação é representada considerando a intensidade, com base em diferentes valores de resistência, ambas, propostas por Li et al. (2010).

Pela Fig. 2.27, Li et al. (2010) chamam atenção que uma mesma intensidade de ameaça pode resultar em graus de danos distintos aos elementos expostos. Assim, enquanto certas estruturas podem ser completamente destruídas por possuírem más características de resistência, aquelas que apresentam melhores valores dos atributos voltados para a resistência da ameaça tem a tendência de sofrer apenas pequenos danos. Considerando uma mesma intensidade, quanto menor a resistência, maior a vulnerabilidade.

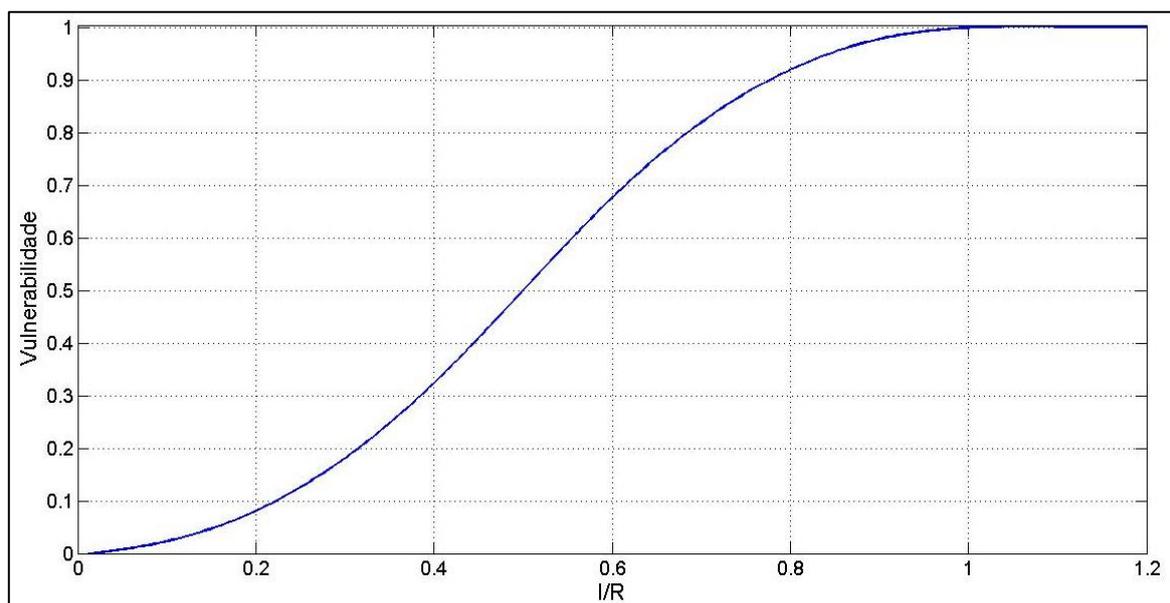


Figura 2.26. Variação teórica de V em relação ao quociente I/R (modificado de Li et al., 2010).

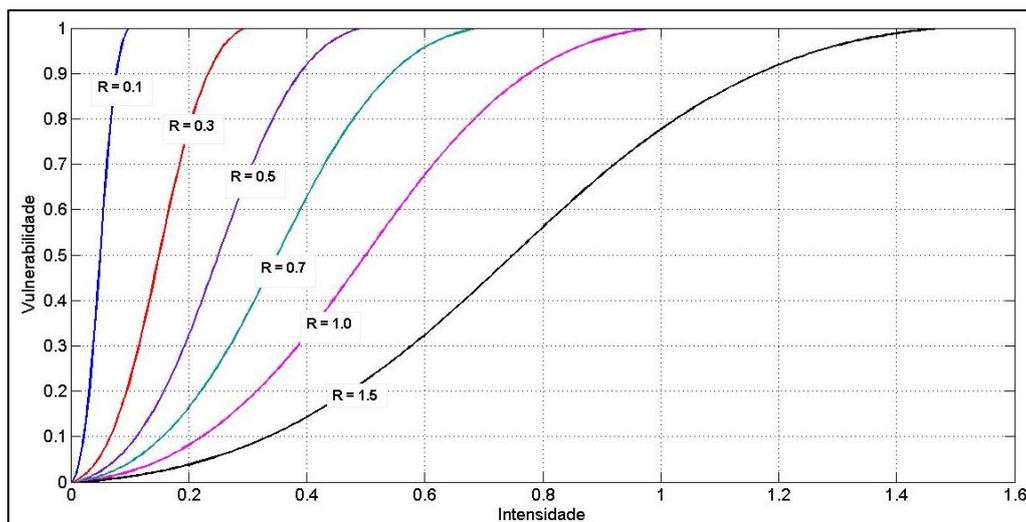


Figura 2.27. Tendência teórica de variação das curvas de vulnerabilidade em função da intensidade da ameaça para distintos valores de R (modificado de Li et al., 2010).

Ao comparar dois elementos com resistências diferentes, observa-se que a vulnerabilidade da construção menos resistente é mais sensível a uma baixa intensidade de ameaça, quando confrontada com outro que tenha resistência superior. Da forma que o modelo foi articulado por Li et al. (2010), no caso em que a intensidade se iguala a metade da resistência conferida ao elemento, o valor esperado para a vulnerabilidade torna-se igual a 0,5. Chama-se a atenção, ainda, para a possibilidade de que tanto a intensidade quanto a resistência atinjam medidas superiores à unidade.

A intensidade da ameaça, no estudo analisado, é definida por um fator de intensidade dinâmica, estabelecido em função da velocidade do deslizamento, e por um fator de intensidade geométrica, responsável por representar as características dos elementos ou deslizamentos existentes na área de estudo, conforme exposto abaixo:

$$I = f(I_{dyn}, I_{gem}) \quad (2.129)$$

onde I_{dyn} e I_{gem} expressam o fator de intensidade dinâmica e o fator de intensidade geométrica, respectivamente. Tais fatores também não possuem dimensões. No caso do fator de intensidade geométrica, este se subdivide, ainda, dependendo da localização das estruturas em relação à área do deslizamento.

Enquanto a definição de I_{dyn} é pautada na velocidade do deslizamento, I_{gem} considera características relacionadas ao tamanho dos elementos expostos ou dos deslizamentos. Dependendo da localização, a intensidade pode ser dada por uma das seguintes equações:

$$I = I_{dyn} \cdot I_{dpt} \quad (2.130)$$

$$I = I_{dyn} \cdot I_{dfm} \quad (2.131)$$

Aplica-se a primeira equação quando as construções estão fora da área do deslizamento. No caso de elementos no interior da área do evento, a segunda fórmula é requerida.

O termo I_{dpt} corresponde ao fator de profundidade de detritos, utilizado na avaliação de elementos situados fora da área de deslizamentos rápidos. I_{dfm} é denominado fator de deformação, empregado na avaliação de estruturas contidas em uma massa deslizante pertencente a um deslizamento com baixa velocidade.

Tanto I_{dpt} quanto I_{dfm} podem ser considerados fatores de intensidade geométrica (I_{gem}), cujo uso varia conforme uma das situações previstas acima. Ambos os fatores serão definidos em detalhes, neste texto, em um momento adiante.

Foram utilizados alguns limiares relacionados à velocidade de deslizamentos para a fixação dos valores do fator de intensidade dinâmica. Estes limites são fundamentados em medidas de velocidade que carregam um significado destrutivo importante, que pode ser causado pelos movimentos de massa envolvidos nas análises.

Para estruturas, foco da abordagem aqui realizada, considera-se a sua destruição ou a perda de funcionalidade ou de manutenção, se a velocidade for superior a 5×10^{-1} mm/s. Nestas circunstâncias, o fator de intensidade dinâmica é considerado igual a 1. O patamar inferior, abaixo do qual as construções não sofrem maiores danos e o fator de intensidade dinâmica é considerado nulo, relaciona-se à velocidade de 5×10^{-7} mm/s. Na equação abaixo, estão expressas as leis que definem o fator supracitado para um valor específico de velocidade de deslizamento.

$$I_{dyn} = \begin{cases} 0 & \text{se } C \leq 5 \times 10^{-7} \text{ mm/s} \\ \frac{1}{36} \cdot (\log_{10} C + 6,3)^2 & \text{se } C > 5 \times 10^{-7} \text{ mm/s} \end{cases} \quad (2.132)$$

Nas equações, C relaciona-se à velocidade média da massa deslizante (em mm/s). Li et al. (2010) mencionam que a estimativa da velocidade pode ser realizada por métodos de simulação de caráter empírico, analítico simplificado ou numérico. No entanto, não foi citado no trabalho de Li et al. (2010) nenhum dos modelos que poderiam fornecer esta avaliação. Pode-se mencionar o trabalho de Serna et al. (2015) como exemplo da aplicação de técnicas

numéricas na estimativa das velocidades dos eventos de deslizamentos, vinculadas a quantificação da vulnerabilidade. Serna et al. (2015) afirmam que a forma como as metodologias atuais de projeto de taludes envolvem a confiabilidade e o fator de vulnerabilidade estrutural, diante da regulamentação em vigor, deve ser revista. Observa-se que, pelo trabalho citado, os métodos de natureza numérica ainda se mostram exigentes do ponto de vista computacional para o alcance das velocidades de escorregamentos, ressaltando que a avaliação de valores plausíveis se mostra como um ponto crucial da metodologia de Li et al. (2010) e que deve ser considerado com atenção. A relação proposta entre a velocidade dos eventos e o fator de intensidade dinâmica encontra-se representada na Fig. 2.28.

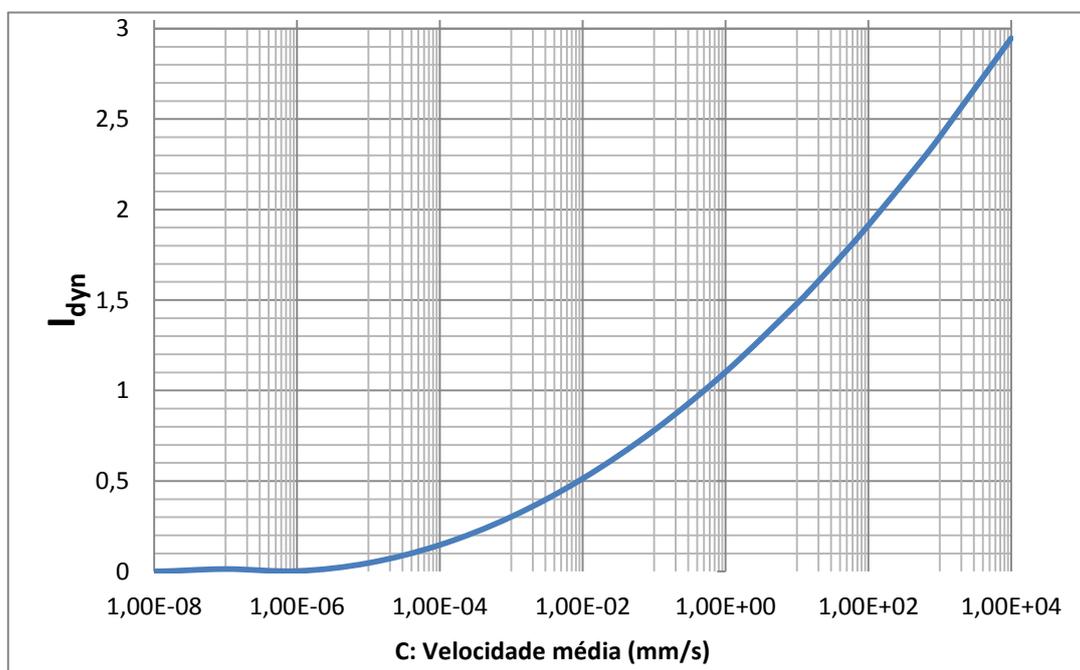


Figura 2.28. Modelo de intensidade dinâmica para estruturas (modificado de Li et al., 2010).

Li et al. (2010) preconizam que a vulnerabilidade é fortemente influenciada a partir do posicionamento apresentado pelos elementos em risco, especialmente para estruturas. Por exemplo, edificações que se encontram exteriores a uma rápida massa instável podem ser atingidas por corridas de detritos com altas velocidades, ou mesmo por blocos de rocha, ao passo que estruturas localizadas no interior de uma massa que se desloca de forma lenta, poderão sofrer com rupturas estruturais ou operacionais decorrentes do recalque diferencial ou devido a deslocamentos absolutos.

Quando se trata dos episódios em que as estruturas estão fora de uma rápida massa instável, Li et al. (2010) citam alguns trabalhos que tratam sobre a vulnerabilidade sob esta

óptica. Conforme Dai et al. (2002), o fator apropriado atrelado à vulnerabilidade pode ser avaliado sistematicamente por meio de julgamentos de especialistas, uma vez estabelecido o tipo particular de instalação e a espessura provável de detritos presentes no local da construção. Já segundo Ragozin & Tikhvinsky (2000), a vulnerabilidade das estruturas encontra-se ligada à profundidade das fundações. Os valores fornecidos à vulnerabilidade, conforme Ragozin & Tikhvinsky (2000), tomando como base diferentes profundidades de fundação e em função da espessura prevista de detritos advindos dos deslizamentos, podem ser observados na Tab. 2.9.

Tabela 2.9. Indicação de vulnerabilidade em diferentes profundidades de fundação (modificado de Ragozin & Tikhvinsky, 2000).

Profundidade da fundação (incluindo estacas), em m	Espessura de detritos dos deslizamentos	Vulnerabilidade
≤ 2	< 2	1
> 2	< 2	0
Menor que a profundidade do deslizamento	2 – 10	1
10 – 13	2 – 10	0,5 – 1
> 13	2 – 10	0 – 0,5
Qualquer	> 10	1

Nota-se que a profundidade de detritos fornece uma ideia da intensidade, enquanto a habilidade de resistência das estruturas é dada pela profundidade das fundações, de forma que edificações que possuem fundações profundas frequentemente têm uma maior capacidade de resistência. O que Ragozin & Tikhvinsky (2000) especificam na Tab. 2.9 como valores de vulnerabilidade, Li et al. (2010) definem como fatores de profundidade de detritos da seguinte forma:

$$I_{dpt} = 0,1.D_{dpt} \quad (2.133)$$

onde I_{dpt} refere-se ao fator de profundidade de detritos e D_{dpt} está vinculado à profundidade de detritos, em metros, provenientes dos deslizamentos, os quais se encontram presentes no local da construção. O fator de profundidade de detritos, em conjunto com o fator de intensidade dinâmica, pode, de certo modo, refletir a energia de impacto ocasionado pela massa deslizante.

A danificação causada a infraestruturas que se encontram presentes no interior de uma massa instável em movimento, geralmente é proporcional à distorção interna de suas fundações. Neste caso, o fator de intensidade geométrica assume a denominação de fator de deformação, sendo definido pela expressão seguinte:

$$I_{dfm} = \frac{D_S}{D_{S,T}} \quad (2.134)$$

onde:

I_{dfm} é o fator de deformação;

D_S relaciona-se ao índice de deslocamento da fundação ou da estrutura;

$D_{S,T}$ refere-se ao valor limite de D_S e com a mesma unidade, acima do qual espera-se um dano estrutural completo ou uma perda de funcionalidade. Pode-se representar o índice de deslocamento de diferentes maneiras, como, por exemplo, sob a forma de recalques absolutos, recalques diferenciais ou inclinação parcial. O aspecto escolhido dependerá do tipo de construção que será alvo da análise.

É possível que estruturas dentro de regiões instáveis apresentem ruptura estrutural, decorrente de movimentos ou recalques diferenciais, ou mesmo rupturas operacionais, observadas quando a medida do recalque ou a inclinação de uma edificação extrapola o valor estabelecido como aceitável, mesmo em ocasiões em que os sinais indicativos deste tipo de falha (como rachaduras) não sejam observados.

Modelos teóricos, baseando-se na experiência ou a partir de dados disponíveis, podem ser aplicados na definição dos limiares críticos. No mesmo trabalho, Li et al. (2010) mostram, como exemplos, alguns limites de deslocamento para fundações e edificações propostos pelo Ministério da Habitação e Desenvolvimento Urbano e Rural da República Democrática da China em 2002, que encontram-se reproduzidos na Tab. 2.10. Conforme será mostrado posteriormente, estes mesmos limiares foram utilizados no sistema computacional proposto como uma opção de entrada de dados, no caso em que não há inicialmente uma estimativa ou um estudo sobre estes valores em uma dada região.

Tabela 2.10. Valores limites exemplificados para deslocamento de fundações, relativos a diferentes estruturas (modificado de Li et al., 2010).

Características de deformação		Solo de fundação	
		Solo com média a baixa compressibilidade	Solo muito compressível
Estruturas de alvenaria	Inclinação parcial da fundação	0,002	0,003
Estrutura de piso único	Recalque (mm)	120 (apenas para solo com média compressibilidade)	200
Edifício arranha-céu com forma simples	Recalque médio da fundação (mm)	200	
Edifícios industriais e civis comuns			
Estrutura em quadros	Recalque diferencial entre estacas adjacentes	0,002 *	0,003 *
Colunas externas com parede de alvenaria		0,0007 *	0,001 *
Edifício arranha-céu			
$H_g \leq 24$	Inclinação integral	0,004	
$24 < H_g \leq 60$		0,003	
$60 < H_g \leq 100$		0,0025	
$H_g > 100$		0,002	

Características de deformação		Solo de fundação	
		Solo com média a baixa compressibilidade	Solo muito compressível
Estrutura em torre			
$H_g \leq 20$	Inclinação da fundação	0,008	
$20 < H_g \leq 50$		0,006	
$50 < H_g \leq 100$		0,005	
$100 < H_g \leq 150$		0,004	
$150 < H_g \leq 200$		0,003	
$200 < H_g \leq 250$		0,002	
Estrutura em torre			
$H_g \leq 100$	Recalque da fundação (mm)	400	
$100 \leq H_g < 200$		300	
$200 \leq H_g < 250$		200	

* (em mm) distância central entre estacas adjacentes; H_g (em m): altura da construção acima do solo.

Parcela imprescindível para a efetivação do cômputo da vulnerabilidade, a componente de resistência reflete a capacidade de elementos vulneráveis em resistirem à ação de um deslizamento que possui um dado grau de severidade, sendo que o modelo aplicado pela pesquisa para o seu cálculo estabelece o emprego de uma série de fatores de resistência, conforme mostrado abaixo:

$$R = \left(\prod_{i=1}^{n_s} \xi_i \right)^{\frac{1}{n_s}} \quad (2.135)$$

Na Eq. 2.135, cada termo ξ_i corresponde a um dos $n_s \geq 1$ fatores de resistência, adimensionais, os quais são responsáveis por representarem diferentes categorias.

Especificamente para estruturas, Ragozin & Tikhvinsky (2000) e Spence et al. (2005) propuseram quatro características consideradas primárias que interferem grandemente na resistência de construções, as quais se encontram citadas abaixo:

- Idade da estrutura, podendo ser representada pelo grau de manutenção;
- Altura da edificação, dada pelo número de pavimentos;
- Profundidade da estrutura de fundação;
- Materiais empregados na construção da estrutura de suporte da carga vertical, correspondendo à tipologia estrutural do edifício.

Assim, a abordagem quantitativa da resistência para estruturas fica determinada pela adequação da equação fornecida anteriormente às quatro características citadas, segundo a fórmula exposta na expressão seguinte:

$$R_{str} = \left(\xi_{sfd} \cdot \xi_{sty} \cdot \xi_{smn} \cdot \xi_{sht} \right)^{\frac{1}{4}} \quad (2.136)$$

onde ξ_{sfd} , ξ_{sty} , ξ_{smn} e ξ_{sht} são, respectivamente, os fatores de resistência atrelados à profundidade da fundação, à tipologia estrutural, ao estado de manutenção e à altura da construção.

Li et al. (2010) explicitam alguns exemplos em relação aos valores que podem ser atribuídos às classes dos fatores de resistência que foram a pouco estipulados. Li et al. (2010) ressaltam que a definição destes fatores pode ser realizada de maneira subjetiva, observando-se as condições e o estado atual das edificações expostas à ameaça. Nas Tab. 2.11, Tab. 2.12 e Tab. 2.13 encontram-se compilados os fatores propostos por Li et al. (2010) para a tipologia estrutural, estado de manutenção e altura, respectivamente, bem como as diferentes categorias conectadas a tais valores.

Tabela 2.11. Valores propostos do fator de resistência para a tipologia estrutural (modificado de Li et al., 2010).

Tipologia Estrutural	Nome de campo	ξ_{sty}
Estrutura muito leve (construção de madeira simples)	LT	0,1
Estrutura leve	LS	0,2
Estrutura mista (concreto e madeira)	MS	0,4
Paredes de tijolos, concreto	BC	0,8
Concreto armado	RC	1,3
Reforçado	RF	1,5

Tabela 2.12. Valores propostos dos fatores de resistência para o estado de manutenção (modificado de Li et al., 2010).

Estado de manutenção	Nome de campo	ξ_{smn}
Muito pobre	VP	0,1
Pobre	P	0,4
Médio	M	0,8
Bom	G	1,2
Muito bom	VG	1,5

Tabela 2.13. Valores propostos dos fatores de resistência para a altura (modificado de Li et al., 2010).

Altura	Nome de campo	Número de pavimentos	ξ_{sht}
Andar único	S	1	0,1
Edifício baixo	L	2	0,4
Edifício médio	M	3,4,5	0,9
Edifício alto	H	6+	1,5

Já o fator de resistência garantido pela profundidade da fundação, pode ser obtido pela aplicação da formulação presente na equação abaixo, assegurada a partir de dados de Ragozin & Tikhvinsky (2000).

$$\xi_{sfd} = \left(\frac{D_{fod}}{10} \right)^{\frac{1}{3}} + 0,05 \quad (2.137)$$

onde o termo D_{fod} é a profundidade da fundação da estrutura, em metros.

O resumo da metodologia aplicada ao cálculo da vulnerabilidade de estruturas pode ser expresso pela Tab. 2.14.

Tabela 2.14. Resumo para os métodos de cálculo de vulnerabilidade para estruturas (modificado de Li et al., 2010).

Velocidade	Localização	Cálculo da vulnerabilidade		
		Resistência	Intensidade	Vulnerabilidade média
Rápido	Dentro	-	-	1
	Fora	$(\xi_{sfd} \cdot \xi_{sty} \cdot \xi_{smn} \cdot \xi_{sht})^{\frac{1}{4}}$	$I_{dyn} \cdot I_{dpt}$	Eq. 2.128
Lento	Dentro	$(\xi_{sfd} \cdot \xi_{sty} \cdot \xi_{smn} \cdot \xi_{sht})^{\frac{1}{4}}$	$I_{dyn} \cdot I_{dpt}$	Eq. 2.128
	Fora	-	-	0

3 METODOLOGIA

A metodologia proposta no desenvolvimento da tese teve como objetivo primordial criar o aparato capaz de fornecer a quantificação do risco com um caráter espacial, encontrando-se retratada nas seções seguintes. Basicamente, a metodologia proposta a ser utilizada na elaboração da pesquisa contemplou as quatro etapas explicitadas na Fig. 3.1.

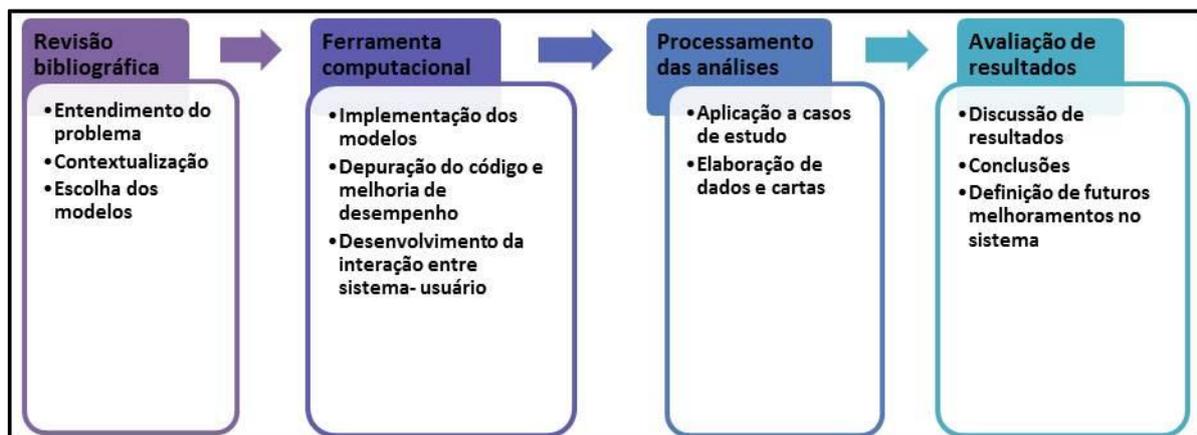


Figura 3.1. Etapas propostas para a metodologia.

Inicialmente, foi realizada uma extensa revisão bibliográfica no intuito de conseguir o entendimento do problema em questão, a qual contemplou os principais pontos pertinentes sobre a relação existente entre eventos de precipitação e movimentos de massa, assim como os modelos existentes de estabilidade de encostas utilizados em estudos de deslizamentos.

A partir de uma pesquisa sistemática na literatura científica especializada, conseguiu-se estabelecer um arcabouço técnico sobre o assunto, composto por conceitos, práticas metodológicas e os principais avanços presentes nesta área do conhecimento. Isto permitiu nortear a pesquisa, auxiliando no desenvolvimento dos passos aplicados ao trabalho.

A execução da primeira fase pertencente à metodologia forneceu subsídios para o estabelecimento da visão global a ser introduzida ao sistema de cálculo de risco devido a deslizamentos, como a escolha dos modelos e suas alterações a serem adotados na composição da ferramenta computacional desenvolvida. Procedeu-se a programação e teste dos códigos responsáveis por determinar as parcelas que definem o conceito de risco associado a escorregamentos de terra. Buscou-se refinar, também, a estrutura dos algoritmos criados quanto ao desempenho, considerando o ganho na eficiência em termos de redução do tempo de processamento computacional.

Esta etapa correspondeu, ainda, ao processo de elaboração de um procedimento que possibilitou a comunicação entre as informações oriundas de uma plataforma de sistemas de

informação geográfica (responsável pela organização do montante de dados ambientais espaciais) e o sistema responsável pelas análises de risco. Em um momento posterior, focou-se na experiência do usuário quanto à manipulação do software, dando-se atenção à elaboração de interfaces gráficas que auxiliassem os comandos de entrada/saída e de processamento das análises.

Uma vez constituída a ferramenta computacional, realizou-se análises com o intuito de comprovar a sua empregabilidade em diferentes casos estudados, gerando uma gama de novas informações, as quais se encontram devidamente discutidas, e proporcionando, desta forma, um melhor entendimento das condições presentes nos exemplos.

Ao final, foram levantadas conclusões conseguidas por intermédio do desenvolvimento desta pesquisa, chamando-se atenção aos aspectos que ainda poderão ser explorados dentro do seu escopo.

Já foram apresentados, no capítulo 2, os modelos teóricos que serviram de base para a criação do sistema, oriundos da pesquisa bibliográfica realizada. Agora, as especificidades, de cunho computacional, ligadas à implementação das rotinas para análise, e dos recursos de utilização disponíveis aos usuários são explicados em detalhes.

3.1 ANÁLISE QUANTITATIVA DE RISCO DEVIDO A DESLIZAMENTO

Apresenta-se, nesta seção, uma descrição das características e recursos associados ao aparato computacional desenvolvido na presente pesquisa, tanto em seus aspectos gerais, quanto em pontos específicos de implementação e estruturação. Ainda aqui, tem-se por objetivo, expor a sequência de etapas do processo de análise por meio do sistema criado, assim como definir os conjuntos de informações necessários para cada uma das opções de modelos disponíveis. Em suma, esta parte do documento se propõe a ser uma espécie de manual de utilização do programa computacional com detalhes da sua concepção, bem como do seu funcionamento e manuseio.

3.1.1 ESTRUTURA GERAL DO SISTEMA

A ferramenta proposta congrega as diversas etapas necessárias para o cálculo do risco atrelado à ocorrência de deslizamentos, fornecendo resultados em termos quantitativos. Ao entender que a estrutura aqui concebida é composta por componentes interconectados que atuam de maneira integrada, formando um todo organizado, pode-se considerar que a

denominação de sistema é cabível ao programa computacional, fruto desta pesquisa, o qual, nesta perspectiva, foi denominado por RiskLab (Risk–Landslide–Laboratory).

O cerne da estruturação aqui aplicada, apresenta-se como o estabelecimento de um procedimento capaz de calcular o risco vinculado a eventos de escorregamentos rasos em uma determinada área, baseando-se em informações relevantes do meio físico, devidamente estruturadas de forma espacial, as quais devem representar adequadamente a região em estudo, compatíveis com a escala de trabalho requerida.

Os modelos que se encontram incorporados à presente pesquisa correspondem a uma compilação de inúmeros trabalhos, os quais foram selecionados criteriosamente, obedecendo aos princípios de relevância técnica e de concordância sinérgica entre as partes do problema abordado. A reunião destes estudos proveio do advento de uma pesquisa sistemática aos principais meios da literatura científica especializada.

Apoiando-se no conceito exposto por Fell et al. (2005), o qual define risco (R) como o produto entre o valor da ameaça (Pr) e as consequências, estas últimas traduzidas em termos de perdas, caso uma ruptura ocorra, arquitetou-se as fases da determinação quantitativa do risco. O termo associado às consequências pode ser desmembrado no produto entre os fatores da vulnerabilidade (V) e os custos impostos aos elementos em risco (C). A obtenção dos valores do risco foi realizada mediante a equação e o modelo expressos na Fig. 3.2.

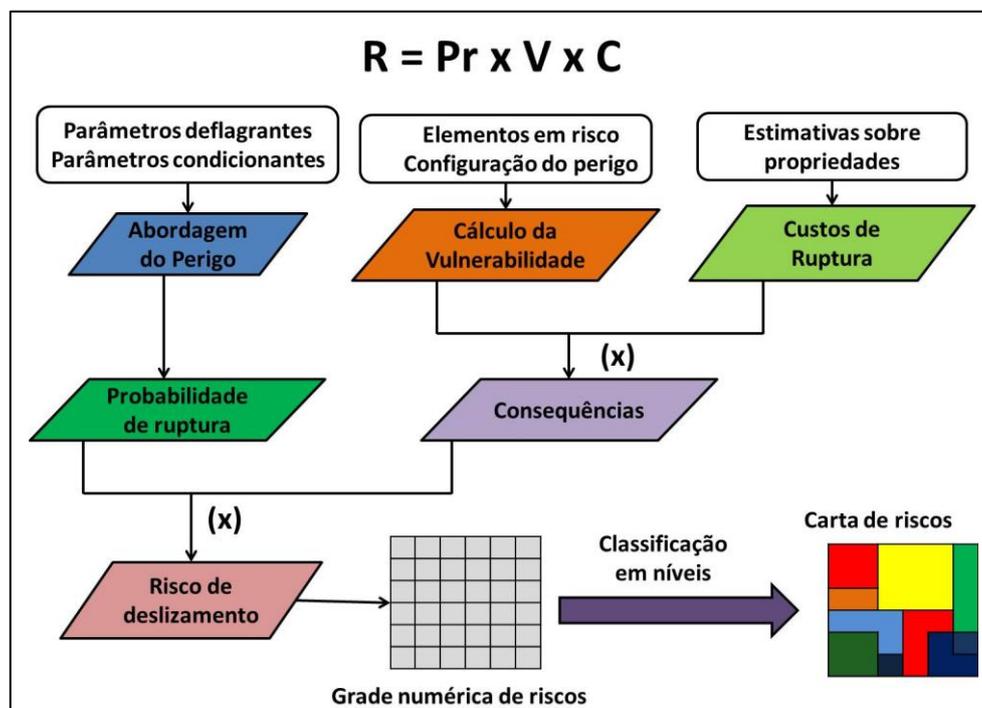


Figura 3.2. Equação e diagrama com a abordagem do modelo de risco utilizado.

Para cada etapa que representa uma determinada parcela da equação de risco, foram criados módulos computacionais específicos referentes a cada uma destas tarefas, capazes de executá-las individualmente. Conjuntamente, elaborou-se, também, um procedimento que procedesse a integração entre os diferentes blocos de cálculo, fornecendo, ao final, o produto esperado, configurado pelo risco dos deslizamentos.

Ressalta-se que, para a implementação do referido sistema, utilizaram-se códigos desenvolvidos na linguagem de programação concernente ao software Matlab. Os módulos relativos às etapas do processamento do risco, entre elas, a análise da ameaça, da vulnerabilidade e dos valores monetários atrelados aos elementos em risco, possuem variadas opções quanto à entrada de dados e aos modelos que podem ser empregados pelos potenciais usuários do sistema.

A concepção do programa computacional criado contemplou sua comunicação com uma plataforma pautada em sistemas de informação geográfica, fato que proporcionou a possibilidade da extensão das análises a um contexto com características espaciais. A Fig. 3.3 traz o esquema da etapa de pré-processamento dos dados, relativo à organização dos insumos na plataforma SIG, com a consequente importação destas informações para o sistema de análise de riscos.

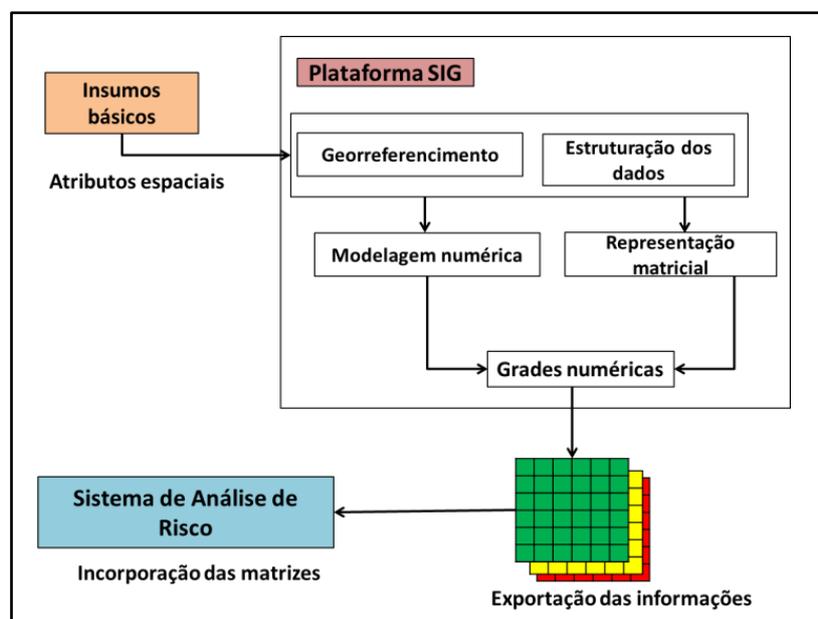


Figura 3.3. Organização da entrada de dados para o cálculo do risco decorrente de deslizamentos.

Em virtude da capacidade presente nas ferramentas de SIG referente à manipulação e transformação de dados geográficos, o seu uso tornou plausível a adequação do arranjo das

informações requeridas pelo sistema de análise de riscos com razoável facilidade. Uma parcela dos diversos atributos do meio físico, que são requeridos à avaliação do risco, apresenta a necessidade de uma representação sob o formato de modelos matriciais para a sua inserção ao programa computacional realizados no Matlab.

Desta forma, cada célula da grade gerada, correspondente a uma determinada porção do terreno e identificada por um par de coordenadas, associa-se, por sua vez, a certo valor da característica que está sendo avaliada. Assim sendo, a base da organização dos dados se fundamenta na representação da região destinada à avaliação sob a forma de matrizes. Para insumos com estas características, deve-se realizar, previamente, sua seleção e ordenação sob o formato raster.

O procedimento de preparação dos elementos demandados nas análises inicia-se com a criação de um banco de dados e um projeto na plataforma SIG, que será utilizada pelo usuário, específicos para região que será alvo de investigação, com a finalidade de organização dos dados geográficos em escala e projeção definidas. A recomendação anterior é pertinente, uma vez que o sistema que calcula o risco, apesar de fornecer a opção de visualização das grades geradas, não permite maiores manipulações nos dados como um sistema de informação geográfica, devendo-se incorporar ao SIG os resultados obtidos para uma melhor avaliação dos cenários alcançados frente aos elementos geográficos da região analisada, com o conseqüente estabelecimento de conclusões pertinentes. A partir disto, pode-se listar uma seqüência de passos encadeados, os quais são requeridos para a geração do conjunto necessário de dados de entrada. Tal seqüência é listada a seguir:

- Uma vez escolhidas as formas de cálculo a serem utilizadas no risco devido a deslizamentos, devem ser observadas as informações exigidas para a sua efetivação. Para cada modelo presente no programa computacional, foi realizada a listagem dos arquivos que deverão ser gerados e tomados como input para o perfeito funcionamento da ferramenta, sendo que cada um deles se relaciona a um determinado fator ambiental da região analisada. Deste modo, isto permitirá nortear a ação do usuário, quanto à demanda de dados necessária para aplicação de alguma opção presente no sistema, possibilitando-o, ainda, antever possíveis problemas quanto à viabilidade concernente à existência ou à disponibilidade das informações solicitadas. Isto proporcionará que, caso na impossibilidade de aquisição de determinado dado, o usuário possa optar por um modelo alternativo;

- De posse da lista de insumos básicos, deve-se proceder a uma busca por fontes voltadas para aquisição destes fatores. Contatos acionados junto a entidades idôneas, como instituições de ensino superior, empresas privadas e os devidos órgãos da esfera pública do Estado, correspondem a uma etapa imprescindível para ter o acesso às potenciais fontes primárias de alcance aos dados iniciais. Certamente, a natureza das informações conseguidas, tanto com relação ao conteúdo quanto à forma, será variada. Assim, faz-se necessário uma triagem, verificando-se quais dos elementos adquiridos podem ser imediatamente inseridos na plataforma SIG e quais deverão passar por algum tratamento antes da sua incorporação definitiva ao escopo dos dados de entrada. Feito isto, executa-se a importação, para o software de sistemas de informação geográfica, dos elementos com atributos espaciais que compõe o meio físico da área objeto de estudo, identificados como importantes no passo anterior, procedendo-se ao respectivo georreferenciamento dos mesmos, quando necessário. Dentro do projeto, sugere-se, no intuito de prover um arranjo adequado, a criação de categorias com denominações representativas para grupos particulares de atributos espaciais, os quais precisam ser agrupados em distintos planos de informações, associando-se a um determinado modelo de dados. Inicialmente, não deve haver uma grande preocupação quanto ao formato dos subsídios inseridos, quer sejam matriciais, quer sejam vetoriais, sendo que esta conformação será citada em um passo posterior;
- Uso dos recursos presentes na plataforma SIG para a geração de novos dados a partir do grupo primário de informações, como a execução da modelagem digital do terreno, que permitirá, também, a derivação de produtos oriundos desta representação, os quais expressam parte do input requerido. Dentre os produtos que podem ser obtidos, encontram-se as grades de declividade, exposição e aquelas relacionadas a processos hidrológicos, como as grades de fluxo acumulado e direção de fluxo. Atenta-se que a realização das demais interpolações, caso haja necessidade, precisam ser efetuadas nesta etapa, bem como a atribuição das características vinculadas às estruturas construídas, que se encontram presentes na área de estudo;
- Verificar se todos os planos de informação que serão exportados para o cálculo do risco encontram-se sob o formato de estrutura matricial e com resolução adequada. Se, porventura, algum layer ainda estiver com uma estrutura vetorial, sua transformação deverá ser efetivada, observando-se que as matrizes relacionadas às representações em formato raster possuem as mesmas dimensões;

- Por fim, realiza-se a exportação do conteúdo que se encontra presente nos planos de informação sob a forma de entidades representadas pelas matrizes regulares.

Com a concretização destes passos, retira-se o cabeçalho específico presente nos arquivos exportados, que traz as informações de controle dos planos de informação precedentes da plataforma SIG (como resolução da matriz, projeção cartográfica, datum, entre outros), salvando-se somente os dados pertinentes à grade numérica em arquivos com extensões de texto (".txt") e com denominações próprias, às quais já foram estabelecidas na mesma listagem de arquivos previamente citada. A nomenclatura deve ser estritamente obedecida, evitando, desta forma, erros no momento da importação dos dados pelo sistema desenvolvido no Matlab.

É recomendado, para efeitos de organização, que a totalidade dos arquivos de entrada deve, então, estar situados em um diretório específico, facilitando o acesso e, conseqüentemente, o carregamento destas informações. No entanto, nada impede que os diferentes arquivos estejam em diretórios distintos, já que o programa permite a busca das entradas em qualquer diretório ou pasta presente no computador onde se realiza a análise.

Para facilitar a interação usuário-sistema, foram criadas várias interfaces gráficas do usuário (do inglês Graphical User Interface - GUI) para cada passo de cálculo, promovendo um manuseio mais familiar do aplicativo, já que os componentes gráficos usados, como botões e menus, assemelham-se àqueles existentes em programas comerciais populares e bastante difundidos. O uso de interfaces gráficas elimina a necessidade dos usuários aprenderem a linguagem de programação ou comandos típicos para executarem uma dada aplicação.

O software Matlab possui uma função própria para elaboração de interfaces gráficas, denominada GUIDE (GUI Development Environment). Por meio do editor de layout procedente desta função, foi possível projetar, graficamente, a interface pelo preenchimento de uma figura com os distintos componentes responsáveis pelas operações de escolha, entrada e saída de dados e processamento das análises nos diversos níveis de operação do sistema.

Ao salvar essa figura, a função GUIDE gera, automaticamente, um arquivo de programa em Matlab com o código-fonte, contendo as funções que controlam como a interface opera. Este arquivo gerado fornece os códigos responsáveis pela iniciação da GUI, bem como contém o quadro para as chamadas das rotinas que deverão ser executadas quando um usuário interage com a interface, dentre as quais se encontram tanto as responsáveis

efetivamente pelo cálculo das parcelas definidoras do risco, quanto àquelas voltadas para alteração de opções de entrada de dados.

Destaca-se que o processo de elaboração das interfaces gráficas ocorreu como último estágio de elaboração do sistema proposto. Os esforços iniciais foram canalizados para a implementação das rotinas computacionais associadas aos modelos de cálculo. Em um primeiro protótipo, a entrada de dados era realizada quase exclusivamente por intermédio da leitura direta de arquivos presentes em um diretório específico (o mesmo onde se encontrava o programa executável), onde os menus presentes eram aqueles pré-definidos pelo Matlab, possuindo opções restritas, quanto ao tipo de análise e aos valores de coeficiente de variação (CV) atrelados aos métodos estatísticos. A Fig. 3.4 traz como exemplo o menu principal da primeira versão do sistema. Na oportunidade, ainda era denominado SARD (Sistema de Análise de Risco de Deslizamentos).

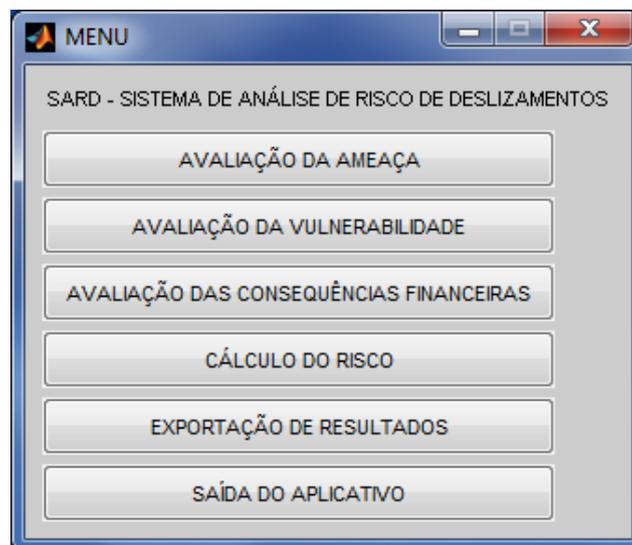


Figura 3.4. Tela do menu principal da versão inicial do sistema.

Diferentemente desta primeira versão do sistema, a incorporação das GUIs mostra-se com o papel fundamental, no sentido de permitir que o usuário possua um maior contato com o programa computacional, obrigando-o a ter um controle e um entendimento maior de todo o processo arquitetado, assim como do funcionamento da organização dos dados de entrada. Isto é imprescindível para que o usuário não seja um mero operador da plataforma, mas que consiga apreender um arcabouço teórico mínimo, acerca do tema de risco e, mais especificamente, aquele atrelado ao risco trazido pelos movimentos de massa. Ao serem concebidos os projetos das GUIs, procedeu-se uma integração entre os elementos gráficos responsáveis pelo input e as chamadas dos códigos de cálculo. A transferência de dados entre

as diferentes partes da GUI foi concretizada pelo uso de variáveis globais, possibilitando que as ações solicitadas pelos usuários em cada nível interajam de forma sincronizada, e que o fluxo de informações ocorra satisfatoriamente e com sincronia entre as janelas do programa.

Ao executar a plataforma, inicialmente o usuário irá encontrar a tela principal do sistema, na qual, além de estarem listadas todas as opções básicas de análise disponíveis, correspondentes às parcelas do risco, opções complementares também se encontram presentes. Além das seleções padrões de ajuda e saída do sistema, há a possibilidade de visualizar os resultados obtidos e, caso interesse ao usuário, a grade pode ser salva para posterior uso. A visão geral da tela inicial do programa computacional encontra-se representada na Fig. 3.5.



Figura 3.5. Interface gráfica inicial com as principais opções do sistema.

Para efetivar a obtenção do risco, as grades de ameaça, vulnerabilidade e custos trazidos pelas rupturas, sob a forma de valor monetário das estruturas, devem ser definidas. Assim, cada um destes modos deve ser acionado pelo menos uma vez. Caso não seja criada uma matriz advinda da aplicação de um modelo específico, o usuário tem a possibilidade de importar uma representação já pronta para cada uma das parcelas que compõem o risco, oriunda de um trabalho realizado anteriormente.

A seleção de uma determinada opção leva ao redirecionamento a uma nova interface, específica daquele modelo ou recurso. Cada tipo de análise possui um ícone criado especificamente para representar e indicar em que passo se encontra a execução do sistema. Estes ícones, que se situam no canto superior esquerdo de cada GUI, são mostrados juntamente com a respectiva etapa que representam, na Fig. 3.6.

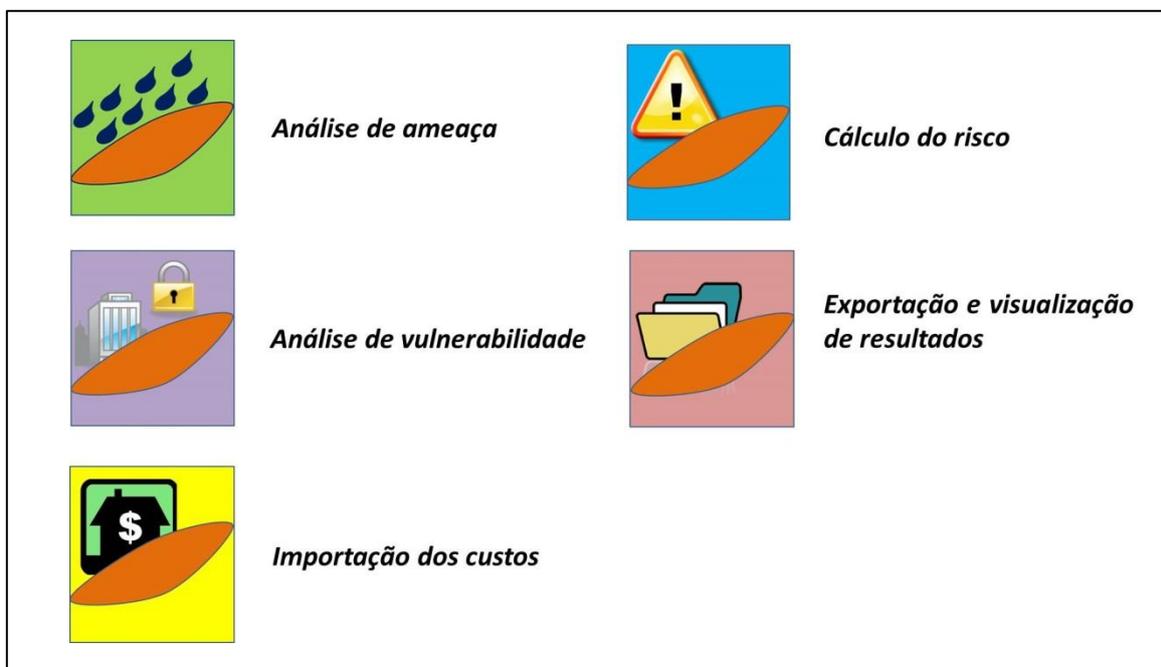


Figura 3.6. Ícones usados para identificar as análises.

As etapas disponíveis serão abordadas, para cada modo, em mais detalhes nas seções que se seguem.

3.1.2 MODOS DE ANÁLISE DE AMEAÇA

O primeiro passo na construção do cenário de risco corresponde ao estabelecimento da condição de instabilidade, que possui a capacidade de trazer uma consequência indesejável perante a distribuição de infraestrutura de uma região. Esta condição de instabilidade é

denominada como ameaça, sendo determinada por um agente natural que detona esta situação e atua no terreno, no sentido de modificar suas características de resistência.

A representação e quantificação da ameaça, ao fazer uso de alguma das opções disponíveis, estão sob a forma de probabilidade de ruptura, com valores que variam de zero (ausência de ameaça e maior grau de estabilidade) a um (instabilidade máxima com grande chance de ruptura). A medida da probabilidade de ruptura está associada, numericamente, à área abaixo da curva de densidade de probabilidade, que se encontra à esquerda do limite crítico do fator de segurança fixado como indicativo de estabilidade.

O módulo responsável pela avaliação da ameaça tem como desígnio principal fornecer, mediante a aplicação da ação de uma precipitação que atua como deflagrador, a matriz de valores associados à probabilidade de ruptura de deslizamentos para a área de estudo, como produto final. Utilizaram-se quatro modelos fundamentais para este fim.

Além das alternativas voltadas para análise e opções padrões de saída e ajuda, o usuário irá encontrar o esquema teórico, ilustrando o procedimento da avaliação da ameaça proposto por Van Westen et al. (2008). Este diagrama visa expor os insumos ambientais básicos, necessários ao estudo de forma ampla, em conjunto com os processos impostos a estes dados para se alcançar a quantificação do panorama de ameaça, situando o usuário da cadeia dos passos envolvidos. A interface correspondente aos módulos encontra-se na Fig. 3.7.

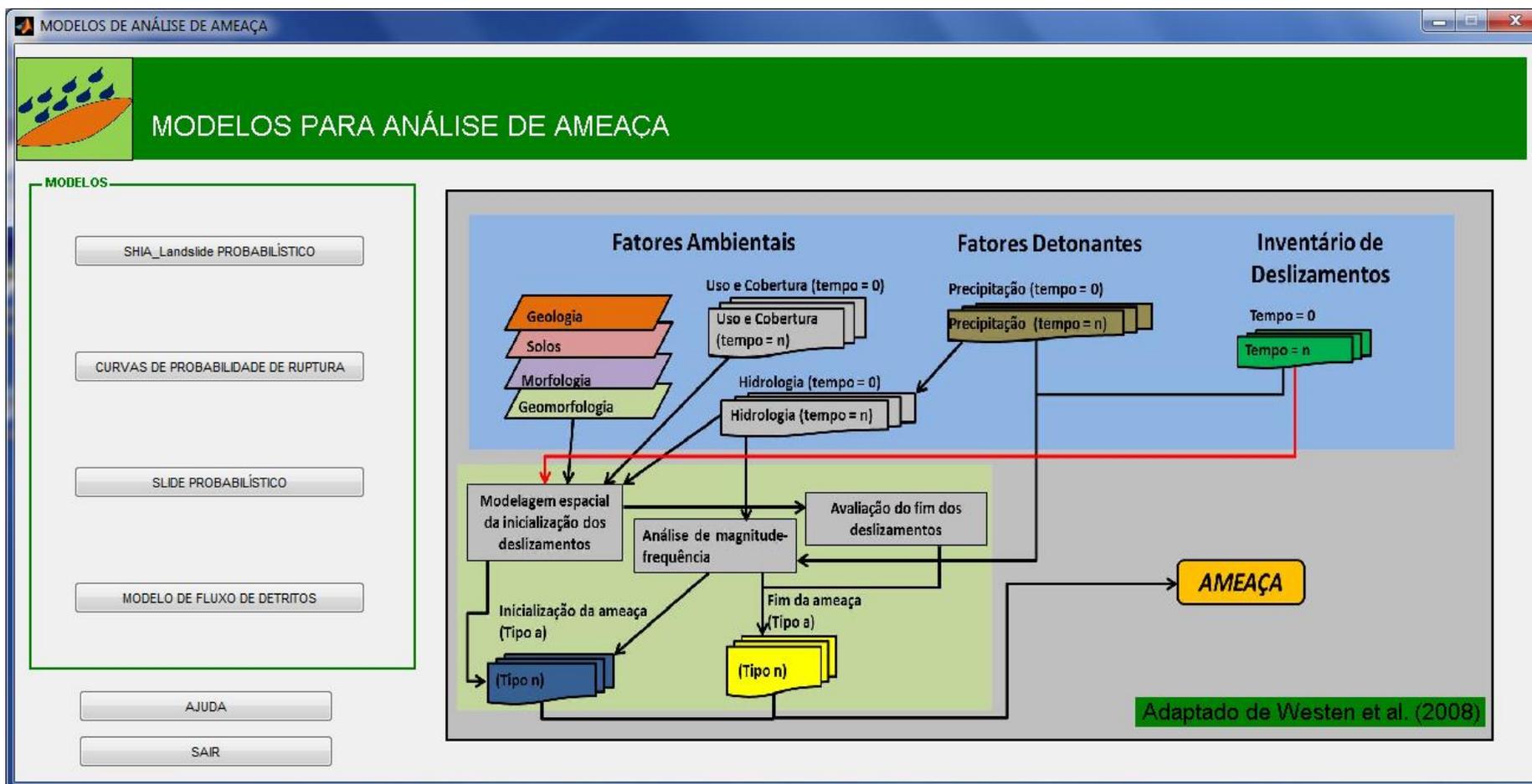


Figura 3.7. Interface gráfica contendo as opções de ameaça.

Em termos gerais, a atuação da abordagem dos modelos que retratam a ameaça carrega um funcionamento comum, independentemente dos cálculos que utilizam, descrito na Fig. 3.8.

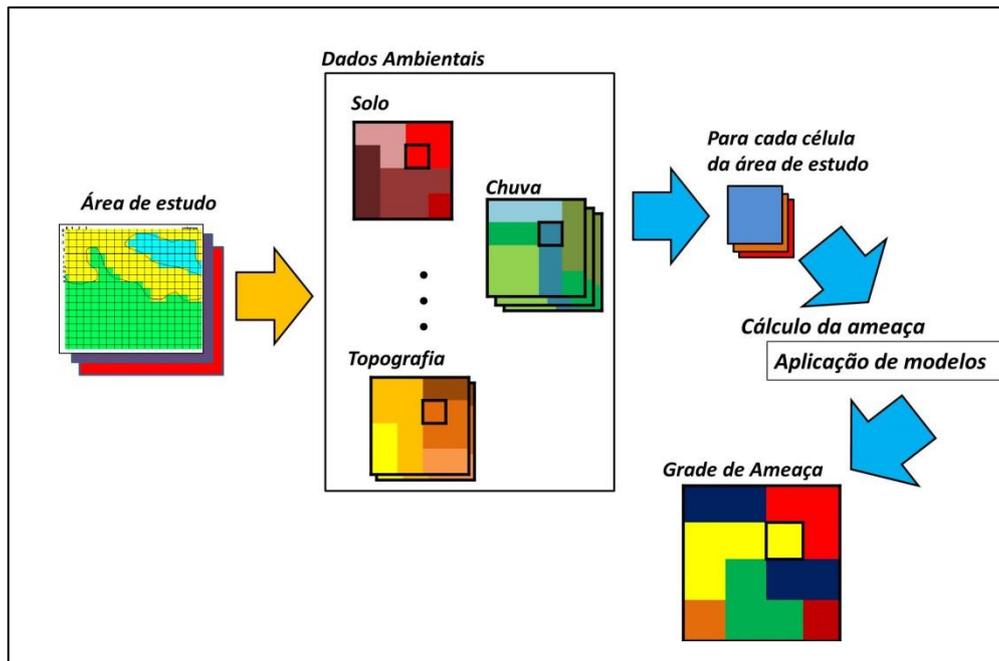


Figura 3.8. Mecanismo geral para cálculo da ameaça.

A Fig. 3.8 representa bem a atuação genérica do código responsável pela quantificação da parcela de ameaça, mediante os inputs impostos. O processo básico se desenvolve de maneira que cada célula pertencente à matriz correspondente à representação de certo retângulo analisado terá seus fatores ambientais selecionados a partir da sua respectiva posição e fornecidos ao conjunto de instruções específicas, concernentes a um dos modelos que se encontram disponibilizados.

Esta ação permite que a quantificação da ameaça para cada um dos pixels seja realizada e, ao final do processo iterativo que varre toda a área de estudo, a matriz para esta primeira etapa da análise do risco esteja completa. Cada um dos modelos incorporados está devidamente abordado e documentado em detalhes, em termos de funcionamento, nas seções seguintes.

3.1.2.1 MODELOS DE CURVAS DE PROBABILIDADE DE RUPTURA EM FUNÇÃO DA CHUVA

Nesta interface, o usuário poderá optar entre três tipos de análises, cujo foco está embasado no uso de funções que fornecem os valores de probabilidade de ruptura em função

da precipitação acumulada em um dado intervalo de tempo. Há, também, uma quarta opção que se refere à importação direta da matriz de ameaça, a qual pode ser proveniente de uma avaliação ou estudo já concretizado em uma pesquisa anterior.

Em função da escolha empreendida no menu de “OPÇÕES DE ANÁLISE”, diferentes botões de seleção para entrada de dados serão liberados para que se possa buscar o arquivo, em um dado diretório, contendo as informações necessárias para uma avaliação específica. Ressalta-se que, ao eleger um tipo de análise, os demais botões para as outras opções ficam desabilitados automaticamente, norteando, de forma intuitiva, a aplicação do programa, reduzindo, assim, a incidência de possíveis erros de input.

Após a escolha de uma determinada opção relacionada ao input das análises que melhor se adapte à intenção e aos recursos daquele que usa o sistema, com a consequente definição do arquivo correto para um dado específico, o caminho retratado pelo diretório desta informação é mostrado no campo de texto situado ao lado do respectivo botão de seleção.

Esta identificação clara do local no qual determinado arquivo encontra-se armazenado, apresenta-se como mais um artifício de garantia contra possíveis erros que possam acontecer na seleção de uma informação, sendo que esta especificação dos diretórios estará presente nas interfaces gráficas dos demais modelos, sempre que ocorrer dados desta natureza.

A adoção das técnicas que utilizam as funções discretas e contínuas parte do pressuposto que o usuário já haveria executado o procedimento proposto por Azevedo (2011), conforme já descrito na seção sobre os fundamentos dos modelos utilizados, para a região objeto de estudo e que, portanto, já se teriam definido, tanto as variadas funções gaussianas de probabilidade de ruptura em função da precipitação acumulada, caracterizadas pelos seus respectivos coeficientes, quanto à distribuição espacial da atuação destas leis, pixel a pixel, por meio da delimitação das suas áreas de influência, sendo estas, parte das informações requeridas para a opção em questão.

O que difere estas duas abordagens corresponde à forma como é feita a atribuição de dados em cada célula de trabalho. No caso das curvas discretas, a região de estudo encontra-se compartimentada em um determinado número de áreas de influência sob a ação de uma determinada estação pluviométrica, onde o comportamento de cada elemento responde a uma dada função gaussiana. A Fig. 3.9 indica a interface correspondente às análises para as curvas de probabilidade de ruptura.

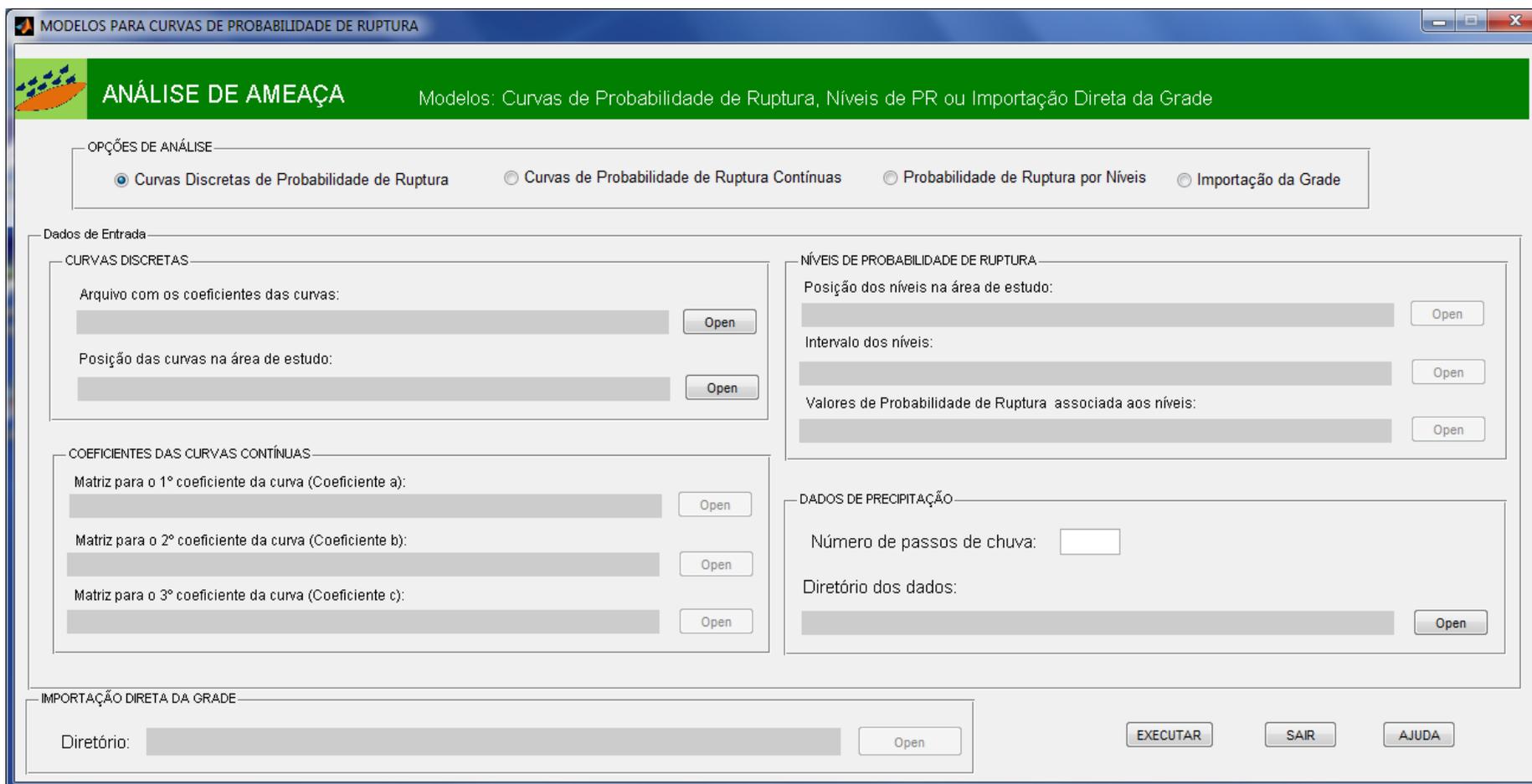


Figura 3.9. Visão da interface gráfica com opções que usam curvas de probabilidade de ruptura em função da chuva e importação direta da grade.

O que se faz é propor uma numeração, de maneira a associar cada pixel que representa uma dada porção do terreno em estudo a um conjunto de coeficientes que definem uma determinada função de probabilidade de ruptura. Na aplicação desta técnica, fica evidente uma descontinuidade entre os resultados gerados na fronteira entre duas áreas de influência contíguas, que são representadas por curvas de parâmetros distintos.

A incorporação da possibilidade em usar curvas contínuas decorre da necessidade em transpor a existência da quebra de continuidade na probabilidade de ruptura promovida pela representação compartimentada do comportamento do terreno, em termos de polígonos de Thiessen, para cada estação pluviométrica. Isso pôde ser superado pela espacialização dos dados dos três coeficientes que definem a curva gaussiana.

As plataformas SIG possuem recursos que possibilitam a criação de planos de informações em formato raster, a partir do processo de interpolação de informações pontuais, caracterizadas pelos valores dos parâmetros que compõem as funções, e que se encontram situados espacialmente na localização das estações pluviométricas.

Agora, cada célula da região estará vinculada a uma curva gaussiana distinta, resultante da interpolação dos parâmetros, sendo que a variação das aplicações gaussianas na extensão do terreno se dará no sentido de não fornecer descontinuidades nesta distribuição.

Para a opção de curvas discretas, o usuário deverá fornecer os seguintes arquivos como dados de entrada:

- Matriz com os coeficientes das curvas gaussianas, a qual deverá estar organizada de forma que cada linha corresponderá a uma determinada função. Assim, a matriz terá tantas linhas quanto o número de funções de probabilidade de ruptura necessário para a descrição da área de estudo. As colunas da matriz representam os coeficientes a, b e c, respectivamente, definidores das funções, conforme Eq. 2.54;
- Representação da área de estudo sob a forma matricial e com resolução apropriada, onde o número inteiro positivo presente em cada elemento faz menção a uma das linhas que constituem a matriz de coeficientes. Desta forma, cada elemento da referida matriz, que está ligado a uma porção do terreno, irá assumir um valor que varia da unidade ao número de linhas da matriz definidora das funções gaussianas citada acima, ou seja, toda região estudada estará atrelada a uma curva de probabilidade de ruptura. Na Fig. 3.10, encontra-se o detalhe para entrada destes dados.

CURVAS DISCRETAS

Arquivo com os coeficientes das curvas:

Posição das curvas na área de estudo:

Figura 3.10. Inputs para a opção de curvas discretas.

Já a utilização das curvas contínuas, implica na aquisição das informações listadas abaixo:

- Matrizes com as dimensões da área de estudo portando os valores interpolados dos coeficientes a, b e c, que definem a curva gaussiana de probabilidade de ruptura em função da precipitação acumulada em um dado intervalo de tempo.

Os formatos suportados para os arquivos podem ser do tipo documento de texto (“.txt”) ou formato ASCII (“.asc”). Destaca-se o painel ativo correspondente a esta alternativa na Fig. 3.11.

COEFICIENTES DAS CURVAS CONTÍNUAS

Matriz para o 1º coeficiente da curva (Coeficiente a):

Matriz para o 2º coeficiente da curva (Coeficiente b):

Matriz para o 3º coeficiente da curva (Coeficiente c):

Figura 3.11. Inputs para análise utilizando curvas contínuas.

Para as duas opções mostradas, são necessárias, ainda, os planos de informações contendo os registros pluviométricos para efetivar as avaliações, conforme especificação:

- Matrizes com as medidas de precipitação atuantes como deflagradoras dos escorregamentos para diferentes momentos. Isto abre a possibilidade de simular a evolução da atuação de um evento de chuva. Estas matrizes devem ter as mesmas dimensões da matriz de representação da área de estudo, proporcionando que todas apresentem a mesma resolução. Atenta-se que as unidades dos dados de acumulados pluviométricos e de tempo devem ser compatíveis com aquelas previstas pelas funções gaussianas incorporadas nos demais arquivos de entrada.

O painel específico da entrada de dados para as informações de chuva requer que o usuário forneça o número de passos (matrizes) de chuva que serão aplicados ao terreno e especifique em qual diretório os arquivos estão armazenados, conforme Fig. 3.12.



DADOS DE PRECIPITAÇÃO

Número de passos de chuva:

Diretório dos dados:

Open

Figura 3.12. Dados necessários para inclusão da precipitação nas análises.

Obrigatoriamente, os arquivos devem estar no formato texto e devem ser nomeados, sequencialmente, como “input_chuva i .txt”, onde i representa o número do respectivo passo de chuva a ser simulado. Assim, por exemplo, para três passos de chuva, os arquivos que deverão estar no diretório selecionado serão “input_chuva1.txt”, “input_chuva2.txt” e “input_chuva3.txt”, sendo que a aplicação destas chuvas às curvas gaussianas irá obedecer esta sequência. Em todos os casos em que as informações de chuva são requisitadas nestes moldes de organização, a nomenclatura dos diversos planos de informações deve seguir a regra estabelecida acima.

Tratando-se dos aspectos de implementação, para uma dada configuração de chuva, a rotina do algoritmo atua varrendo toda a matriz da área de estudo com a identificação das curvas, verificando e carregando as funções correspondentes a cada elemento, com a subsequente aplicação da referida equação, proporcionando, desta forma, o alcance da probabilidade de ruptura definida pela medida de chuva acumulada para aquela célula específica.

Para as curvas discretas, cada número lido em um dado pixel da área de trabalho leva ao conjunto de três valores dos parâmetros a , b e c , correspondente a certa linha da matriz de coeficientes. Estes valores são substituídos na equação da função gaussiana e o valor lido da precipitação é substituído como a variável independente. No caso contínuo, os coeficientes são retirados em cada posição, diretamente das três matrizes importadas pelo sistema.

Este procedimento é realizado de maneira iterativa para todas as matrizes inseridas com informações pluviométricas, resultando, ao final, na geração de igual número de grades de probabilidades de ruptura, proporcionando averiguar a influência da chuva na estabilidade das encostas para toda a região definida.

Desenvolveu-se, também, uma variação deste modelo em que, ao contrário da utilização de funções gaussianas contínuas, empregam-se, agora, funções definidas por várias sentenças abertas correspondentes a um conjunto de funções constantes para cada nível de probabilidade de ruptura, assumindo uma configuração escalonada. Assim, definem-se patamares específicos através dos intervalos de precipitação e seus consequentes valores de probabilidade de ruptura. Como dados de entrada, além das matrizes contendo as informações de precipitação nos moldes explicados acima e do estabelecimento da área de estudo com organização idêntica àquela explanada para a opção das curvas discretas, somam-se os arquivos nos quais contém a definição das funções provedoras dos níveis de probabilidade de ruptura:

- Matriz que traz a divisão dos intervalos de precipitação acumulada, em determinada unidade de medida, de forma que cada linha define a configuração dos patamares para fornecimento da probabilidade de ruptura. Novamente, o arquivo terá tantas linhas quanto o número necessário de funções concebidas para modelar o comportamento da área de estudo. As colunas trarão os limites extremos de cada trecho, que correspondem às abscissas dos intervalos pluviométricos definidores dos níveis de probabilidade de ruptura. É requerido que, ao final de cada linha, após o último valor válido, seja incluído o número -1 como procedimento de controle. O usuário também poderá empregar funções que tenham um número diferenciado de intervalos de precipitação entre si. Neste caso, necessita-se que as linhas com um número menor de elementos válidos sejam complementadas com valores -1, de forma que todas fiquem com o mesmo número de colunas, conforme o arquivo representado pela Fig. 3.13;

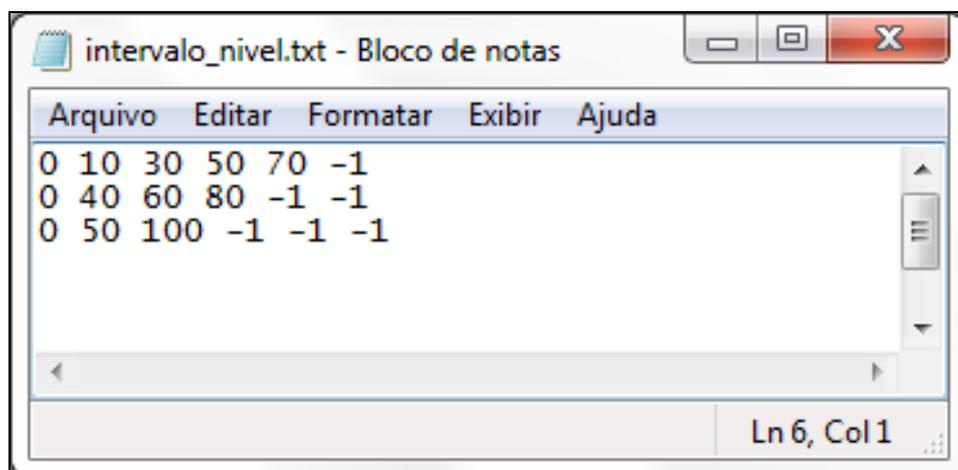


Figura 3.13. Exemplo de arquivo empregado na opção que usa níveis de probabilidade de ruptura.

- Arquivo com os valores de probabilidade de ruptura que se associam a cada intervalo de chuva para as funções definidas. A disposição dos valores presentes nas linhas desta matriz deve ter perfeita concordância com os intervalos definidos na matriz anterior, atentando-se que a sequência das linhas toma significativa importância, uma vez que tanto a base da enunciação de toda a região analisada, quanto à escolha correta da função aplicada ao estabelecimento da probabilidade de ruptura, pautam-se na configuração destas linhas. Aqui também se devem completar aquelas linhas com um número menor de elementos com o valor -1. Não há obrigação da incorporação do número de controle após o último valor válido de probabilidade de ruptura na linha com o maior número de elementos, mas todas as linhas do arquivo deverão ter a mesma quantidade de colunas, caracterizando, de fato, uma matriz.

O processo aplicado pelo programa, nesta opção, é semelhante àquele comentado para as curvas gaussianas discretas, diferindo-se apenas pelo fato de que as funções são dadas por dois arquivos diferentes. Nos casos que os dados de precipitação considerados para a previsão encontram-se abaixo do limite inferior do primeiro intervalo, atribui-se a esta medida de chuva a probabilidade de ruptura definida por este tramo (menor valor de probabilidade de ruptura). De maneira similar, se a precipitação exceder o máximo valor estipulado para intervalo final confere-se a probabilidade de ruptura prevista pelo intervalo, correspondente ao limite superior.

Há, ainda, como quarta opção, a possibilidade de incorporar diretamente a camada de ameaça pela importação de uma grade referente a uma análise realizada anteriormente no próprio sistema ou em outra ferramenta específica, bastando, para tanto, selecionar o arquivo correspondente.

Para efetivar qualquer uma das análises mostradas, basta o usuário pressionar o botão “EXECUTAR” na parte inferior da interface. Com isso, uma barra de progresso surgirá, indicando a evolução do processo em porcentagem, assim como indicado na Fig. 3.14. Destaca-se que este indicador do transcurso da tarefa também estará presente para os demais modelos.

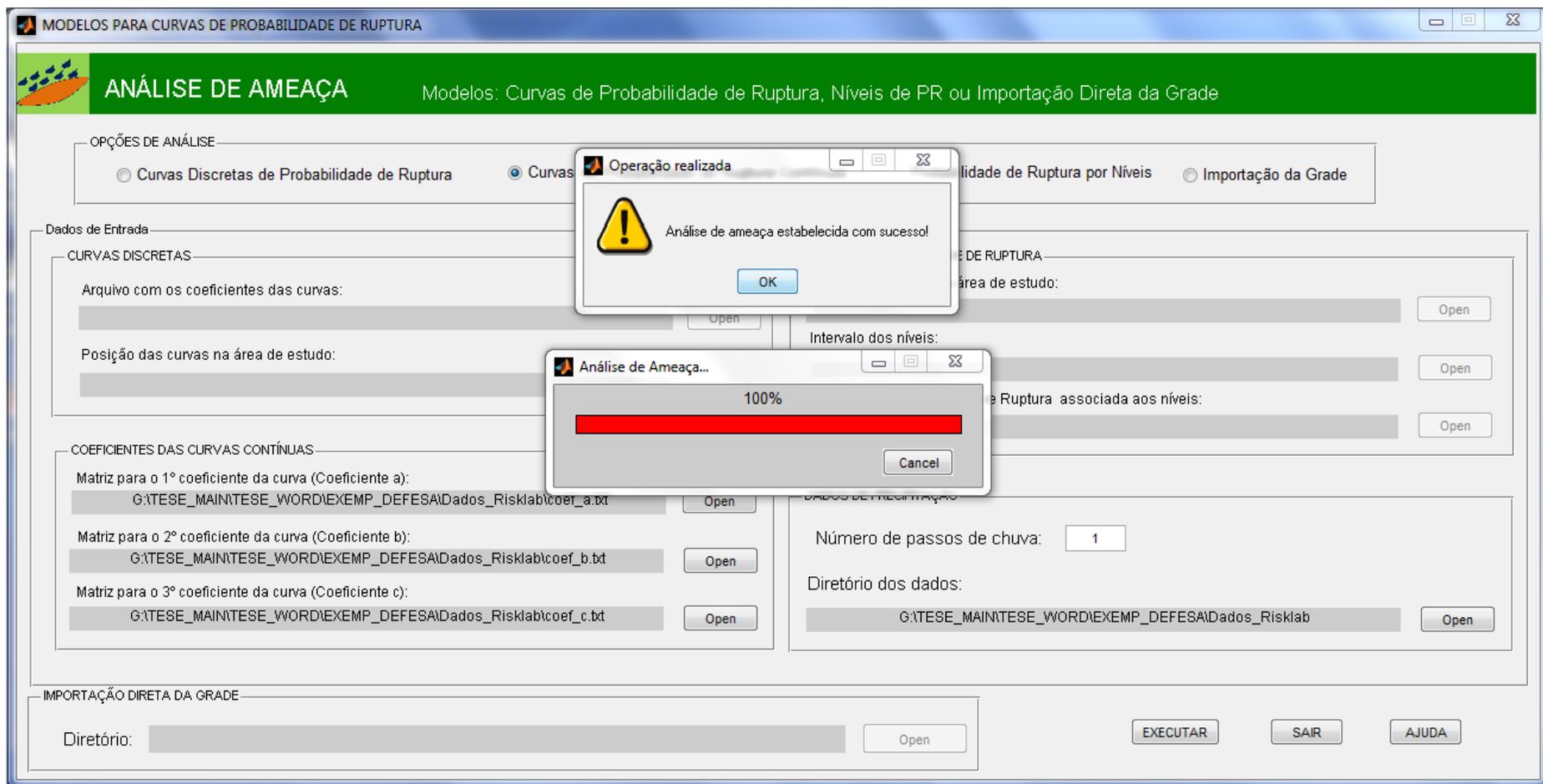


Figura 3.14. Indicação da barra de progresso para o processo de ameaça.

3.1.2.2 MODELO SHIA_LANDSLIDE PROBABILÍSTICO

A estrutura elaborada para o acesso e interação ao modelo SHIA_Landslide segue os moldes propostos por Aristizábal (2014). Por este painel, são introduzidos todos os parâmetros de natureza geotécnica e hidrológica, fatores de calibração e condições iniciais demandados pelo modelo. A janela encontra-se dividida em grupos principais de painéis, correspondentes às informações dos mapas da área de estudo, dados do evento, limites da área, condições iniciais, pontos de controle e fatores de correção.

Todos os elementos associados ao campo “MAPAS E DADOS DA ÁREA” deverão ser congregados ao sistema sob o formato de arquivos, necessitando-se, portanto, da respectiva indicação do diretório onde cada um encontra-se salvo. Todos os arquivos desta ala terão formato matricial, representando espacialmente as informações da região avaliada, pixel a pixel. Todos os layers, nesta situação, deverão ter resoluções iguais, resultando em grades com as mesmas dimensões.

A única exceção nesse primeiro conjunto de dados citados da área avaliada, quanto ao tipo de organização apresentado, condiz ao último arquivo solicitado desta lista, denominado “Valores de dx, x_lim, y_lim e no_data”. Ao contrário dos mapas expressos em matrizes, este arquivo será organizado em um vetor linha com as informações, nesta ordem, de:

- Dimensão do pixel (resolução espacial das células de trabalho), em metros;
- Latitude, expressa em coordenada plana (em metros) do canto inferior esquerdo do retângulo que define a área de estudo;
- Longitude, expressa em coordenada plana (em metros) do canto inferior esquerdo do retângulo que define a área de estudo;
- Valor que deverá ser considerado como inválido e indicativo da ausência de dados pertinentes e representativos.

Na abordagem original de Aristizábal (2014), todos esses dados estavam presentes em um único arquivo. Para fins de implementação, foi necessária a presença dos elementos acima mostrados, sendo inseridos no presente sistema, no entanto, separadamente, por questões de conveniência e adequação à estrutura geral de organização imposta ao código computacional. Os demais inputs para limite de área, condições iniciais, fatores de correção e pontos de controle deverão ser inseridos diretamente nos campos de edição correspondentes, já que cada valor servirá como um fator para toda a bacia. A Fig. 3.15 mostra a GUI designada para a alternativa do SHIA_Landslide.

MODELO SHIA_Landslide PROBABILÍSTICO

ANÁLISE DE AMEAÇA

Modelo: SHIA_Landslide Probabilístico

MAPAS E DADOS DA ÁREA

	Unidades	Diretório	
Direção de fluxo	1-9		Open
Declividade	rad		Open
Área Acumulada	m ²		Open
Capacidade de Infiltração (Ks)	cm/h		Open
Capacidade de Percolação (Kp)	cm/h		Open
Cap. de Armaz. Max (S1max)	mm		Open
Cap. Gravit. Max. (S3max)	mm		Open
Prof. da Superfície de Rupt. (Zs)	m		Open
Peso Específico do Solo (Y)	kN/m ³		Open
Ângulo de Atrito (phi)	rad		Open
Coesão (C)	kN/m ²		Open
Evapotranspiração (EVP)	mm		Open
Valores de dx, x_lim, y_lim e no_data			

LIMITE DA ÁREA

Parâmetro	Unidade	Valor
Encosta	m ²	
Canal	m ²	

FATORES DE CORREÇÃO

Parâmetro	Intervalo	Valor
S1max	min: 0.1; max: 1.5	
EVP	min: 0.5; max: 2	
Ks	min: 0.0; max: 1	
Kp	min: 0.0; max: 1	
Kpp	min: 0.1; max: 10	
S3max	min: 0.1; max: 2	
V2	min: 0.1; max: 2	
V3	min: 1; max: 1000	
V4	min: 1; max: 1000	
V5	min: 0.5; max: 1.5	
C	min:--;max:--	
Phi	min:--;max:--	
Zs	min:--;max:--	

CONDIÇÕES INICIAIS

Parâmetro	Unidade	Valor
Vel. superficial (V2)	m/s	
Vel. subsuperficial (V3)	m/s	
Vel. fluxo de base (V4)	m/s	
Vel. corrente de fluxo (V5)	m/s	
Armazenagem estática (S1)	mm	
Armazenagem superf. (S2)	mm	
Armazenagem gravit. (S3)	mm	
Aquífero (S4)	mm	
Corrente (S5)	m ²	

MÉTODO FOSM

DEFAULT

Parâmetro	C	phi	Ys
CV (%)	40	10	3

ANÁLISES DE PR

Padrão (Susceptibilidade e tempo final da chuva) Tempos definidos

TEMPOS PARA PR

Open

DADOS DO EVENTO E FS CRÍTICO

Dados Chuva Open

Coord. Postos Open

FS crítico Open

Dados Open

PONTOS DE CONTROLE

	Linha	Coluna
Mais baixo	<input type="text"/>	<input type="text"/>
Controle	<input type="text"/>	<input type="text"/>

EXECUTAR SAIR AJUDA

Figura 3.15. Interface gráfica do usuário referente ao modelo SHIA_Landslide.

A entrada de dados para a precipitação segue o mesmo padrão apresentado no trabalho de Aristizábal (2014), diferindo-se da proposta que necessita de planos de informação prontos, representando a área total do terreno, para cada passo de tempo. Três arquivos deverão ser incorporados por meio do painel ligado aos dados do evento, mostrado em detalhes na Fig. 3.16.

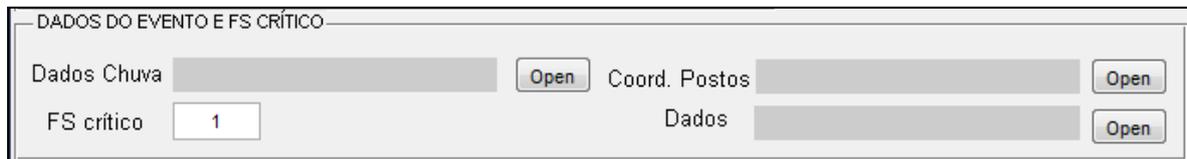


Figura 3.16. Painel específico para incorporação de dados de chuva.

O campo “Dados Chuva” refere-se ao arquivo composto pelas medidas de precipitação que serão adicionadas, as quais estarão dispostas em tantos vetores quanto o número de registros a serem simulados. Assim sendo, o arquivo tomará a feição de uma matriz, a qual está organizada de tal maneira que cada linha trará os registros pluviométricos para um dado intervalo de tempo e cada coluna representará certa estação onde aquela aferição de pluviosidade foi observada.

Em adição, deve-se estabelecer um arquivo que contenha um único vetor, definindo, sequencialmente, as seguintes informações:

- Intervalo de tempo correspondente entre os registros de chuva, igual para todos e dado em segundos;
- Número total de dados pluviométricos que serão impostos ao modelo e que representa a quantidade de linhas que a matriz do arquivo anterior deverá ter;
- Número de postos pluviométricos presentes na área de estudo, de onde se vincularão as medidas de chuva.

O terceiro arquivo deverá conter as coordenadas, em sequência, das estações pluviométricas. Nesta grade, a primeira coluna representará a latitude, enquanto a segunda, a longitude. Conseqüentemente, o número de linhas estará condicionado ao número de estações.

Posteriormente, foi introduzida a este painel, uma opção adicional, quanto ao limiar crítico que define a condição de estabilidade de uma determinada encosta.

Até então, o fator de segurança adotado como limite para distinguir a qualidade de estável ou instável de uma célula de trabalho, do ponto de vista determinístico, referia-se ao valor unitário habitual, correspondendo a condição de equilíbrio limite, sendo que tal procedimento encontrava-se embutido de forma fixa no código. No entanto, na perspectiva de

que se tenha uma flexibilidade da margem de tolerância frente à ameaça de um cenário de escorregamentos, propôs-se a entrada de mais um campo na interface gráfica, denominado “FS crítico”, voltado para que o fator de segurança extremo mais adequado possa ser indicado, passando a representar, neste caso, uma medida de alerta e não mais a condição crítica de instabilidade do talude. Isso permite que se possa controlar a amplitude do intervalo de segurança, que será utilizado nas análises, conforme algum critério formal ou de entendimento ou experiência do usuário que melhor atenda a uma situação específica. De forma padrão, o valor limite do fator de segurança é inicialmente considerado como a unidade, com a possibilidade de modificação.

Chama-se a atenção para o fato de que o parâmetro de correção relativo ao fator de segurança presente na abordagem original, e que tinha por função definir o valor mínimo aceito como estável, é substituído pela entrada “FS crítico” neste novo enfoque. Assim, o campo correspondente a este fator foi retirado do painel de fatores de correção, para evitar eventuais problemas de duplicidade entre tais variáveis.

Painéis complementares também foram adicionados nessa GUI, trazendo os elementos responsáveis pela natureza probabilística, agora incorporada ao modelo. O método probabilístico que se optou para análise corresponde ao FOSM, onde os parâmetros disponíveis para a inserção da aleatoriedade referem-se à coesão, ao ângulo de atrito efetivo e ao peso específico do solo.

A entrada de dados, no tocante à variabilidade para essas variáveis, ocorre sob a forma dos seus correspondentes coeficientes de variação, em porcentagem. Na quantificação dos referidos coeficientes, pode-se impor um conjunto de valores-padrão (default), oriundos de trabalhos técnicos focados, especificamente, no estudo da dispersão destes dados, ou permitir que o usuário imponha as medidas que melhor retratam a condição de aleatoriedade em uma determinada condição de aplicação do modelo. O produto entre a medida de CV para uma dada variável independente probabilística e sua respectiva grade de valores médios irá fornecer a matriz de desvio padrão para toda área de estudo. Para cada variável probabilística, este procedimento será executado, de forma que as matrizes de desvios padrões calculadas servirão na execução dos métodos probabilísticos (diretamente, no caso do método das estimativas pontuais, ou possibilitando o cálculo das variâncias para o método FOSM).

Apoiando-se no trabalho de Assis et al. (2012), os valores relativos aos coeficientes de variação, que são adotados como padrões para a coesão efetiva, ângulo de atrito efetivo e peso específico do solo, correspondem a 40%, 10% e 3%, respectivamente.

Há, também, a possibilidade de escolha quanto à discriminação dos passos de probabilidade de ruptura, que serão gerados e armazenados, possibilitando a sua visualização e exportação, assim como seu emprego nos demais estágios do cálculo do risco. Esta opção funciona de maneira que o usuário tenha que optar por criar, como resultado, apenas duas matrizes de ameaça ou estabelecer uma sequência com a numeração de interesse dos intervalos de tempo relacionados a determinadas grades de chuva.

No primeiro caso, estarão disponíveis para uso, as matrizes de susceptibilidade (caracterizada pela condição de estabilidade do terreno, antes de ser submetido à ação da primeira chuva) e a configuração final de PR após a aplicação do último cenário de chuva, enquanto que, no segundo, define-se, ao final, uma matriz de ameaça para cada passo de chuva presente na sequência e que corresponde à condição de estabilidade do terreno, ao final da aplicação de cada uma das precipitações.

Caso se escolha por definir quais tempos o programa deverá calcular a ameaça, na alternativa de “Tempos definidos”, automaticamente será habilitado o botão no painel “TEMPOS PARA PR”, de modo que se indique o respectivo arquivo contendo estes intervalos, conforme detalhado na Fig. 3.17.

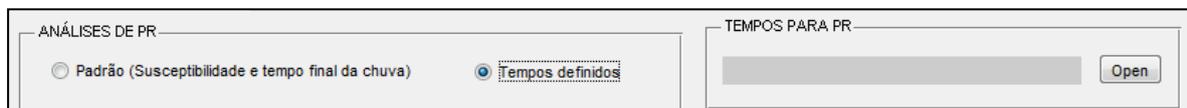


Figura 3.17. Painel retratando as opções de análise da PR.

O arquivo a ser adicionado nesta opção deverá ter a forma de um vetor, cujos elementos que o compõem representarão a numeração dos passos de tempo de interesse para a definição da ameaça, onde cada um deles deverá pertencer ao intervalo que varia desde a unidade até o número total de intervalos de chuvas usados na modelagem. A quantidade presente neste vetor fica a critério do usuário, desde que obedeça a regra estabelecida acima e não extrapole a própria quantidade de chuvas da simulação (o que caracterizaria a referência, portanto, a um dado inexistente).

Quanto aos aspectos computacionais, a base do código fundamentou-se na listagem das rotinas propostas por Aristizábal (2014) que constituem o modelo SHIA_Landslide. Esta listagem está disponível no anexo do referido trabalho e encontra-se desenvolvida na linguagem de programação FORTRAN, conforme já mencionado. Desta maneira, houve a necessidade da conversão do código disponível para a linguagem Matlab.

Nesta adaptação, não se utilizou o código original em sua totalidade. No que se refere às rotinas voltadas para entrada de dados, foi necessário alterá-las, muito embora a quantidade de variáveis iniciais não fosse modificada, conforme observado a partir da explicação acima, concernente à disposição dos componentes da interface gráfica deste modelo. As modificações empreendidas dizem respeito ao emprego da GUI e à leitura dos arquivos por meio do Matlab, processos que se executam diferentemente entre linguagens computacionais.

A organização do código computacional se deu fazendo-se a separação entre a etapa de input, composta pela respectiva importação das grades e parâmetros envolvidos e processada inicialmente, assim que se aciona o botão responsável pela execução do modelo, e a imposição das rotinas de análise, estas salvas em um script próprio, que é chamado por meio de uma única linha de comando após todos os insumos estarem disponíveis no sistema.

Grande parte das funções pertencentes ao modelo SHIA_Landslide não sofreram modificações substanciais, não tendo, inclusive, troca de nomes de variáveis já existentes. Toda a implementação relativa à metodologia de balanço hídrico proposta pelo modelo permaneceu sem mudanças, tanto em relação à calibração do procedimento, quanto às interações de distribuição existentes entre os tanques. As alterações significativas, que na verdade são formadas, na sua maioria, por inclusão de novos códigos, concentram-se na porção voltada para as análises de estabilidade de encostas.

Foram definidos os passos para efetivação do método probabilístico FOSM, correspondente às etapas ligadas à susceptibilidade, realizada em um momento anterior à imposição do primeiro dado de chuva, e para os instantes escolhidos pelo usuário (tempos intermediários ou somente aquele ao final da última precipitação).

As instruções propostas pelo método FOSM compreendem aos cálculos do valor do fator de segurança médio e sua da variância, conforme, respectivamente, a Eq. 2.20 e a Eq. 2.19, expressas no capítulo que apresenta os fundamentos teóricos dos métodos probabilísticos, para cada uma das células que fazem parte da constituição da região estudada, tomando como variáveis independentes probabilísticas o peso específico do solo, a coesão e o ângulo de atrito. O desvio padrão da variável dependente foi definido pela extração da raiz quadrada da variância (para cada célula da matriz).

Pelo fato de utilizar uma função analítica para o cálculo da estabilidade de taludes, as derivadas de FS, requeridas pelo método FOSM, em relação a cada variável independente probabilística, foram estabelecidas diretamente, sem a necessidade de se recorrer à derivada numérica. Adotou-se tal procedimento nos demais modelos que empregam o método FOSM.

A proposta do modelo geotécnico adotada por Aristizábal (2014), de natureza determinística, foi parcialmente alterada neste ponto da implementação. Trabalhou-se, aqui, com a Eq. 2.83 do enfoque original proposto por Aristizábal (2014) para fornecer o valor médio do fator de segurança dentro da abordagem dos métodos probabilísticos, mas sem que houvesse a necessidade de obtenção das espessuras máximas e mínimas que representam as condições extremas de estabilidade, nem do valor de Z_{wcrit} . Consequentemente, o resultado referente à medida de ameaça pôde ser mensurado, não se manifestando mais, como no modelo original, em termos qualitativos, com células classificadas em “estáveis” ou “instáveis”.

De posse das matrizes dos fatores de segurança médios (μ_{FS}) e dos respectivos desvios padrões (σ_{FS}), partiu-se para a execução do procedimento que garantisse, agora, o cômputo da probabilidade de ruptura por meio da adoção, como hipótese, de uma distribuição normal para função de probabilidade atrelada ao fator de segurança. Ou seja, cada pixel terá associado à sua posição, uma distribuição normal do fator de segurança, caracterizada pelo par de fatores μ_{FS} e σ_{FS} , o que permite a definição da probabilidade de ruptura, mediante o estabelecimento do valor crítico de FS (FS_{crit}). Como citado anteriormente, a PR é numericamente representada pela área da curva de probabilidade definida para valores de fator de segurança menores que o limite sugerido para indicar a estabilidade.

Por questões de facilidade e desempenho computacional, decidiu-se fazer uso da normalização de variáveis, transformando-se uma, até então, variável aleatória normal em uma variável normal padronizada, expressa, de maneira genérica, conforme a seguinte fórmula:

$$Z = \frac{FS - \mu_{FS}}{\sigma_{FS}} \quad (3.1)$$

O uso desta formulação se traduz no fato de que a equação da função de distribuição de probabilidade para a variável padronizada pode ser descrita sem o uso direto dos valores da média e do desvio padrão do FS, sendo escrita apenas por meio da variável independente Z, conforme mostrado na equação abaixo:

$$FS(Z) = \frac{1}{\sqrt{2\pi}} e^{\left[-\frac{1}{2}Z^2\right]} \quad (3.2)$$

Esta variável normal padronizada é representada por uma distribuição que sempre terá média igual a zero e desvio padrão unitário. Especificamente para o caso de FS_{crit} , é definido o respectivo valor em termos da variável padronizada conforme a seguinte expressão:

$$Z_{crit} = \frac{FS_{crit} - \mu_{FS}}{\sigma_{FS}} \quad (3.3)$$

O método de obtenção da probabilidade de ruptura de cada pixel pode ser retratado pelo cômputo inicial da variável padronizada Z , correspondente ao valor crítico de estabilidade indicado pelo usuário. Com isso, criam-se dois vetores, o primeiro contendo um conjunto de valores da variável Z pertencente ao intervalo de -10 ao Z_{crit} , separados entre si por uma medida de 0,01, enquanto o segundo armazena as imagens da função $FS(Z)$, oriundas da aplicação de cada elemento do primeiro vetor à Eq. 3.2.

Do ponto de vista de implementação, existe uma facilidade quanto à questão da programação da maneira como foi realizada, já que ao equacionar a função matemática, o código já deverá ter calculado, a priori, a variável Z_{crit} , não necessitando introduzir, portanto, valores da média e desvio padrão nesta formulação.

A Fig. 3.18 traz de forma esquemática o procedimento para o alcance numérico da probabilidade de ruptura, a partir dos dados que definem a distribuição normal para o fator de segurança.

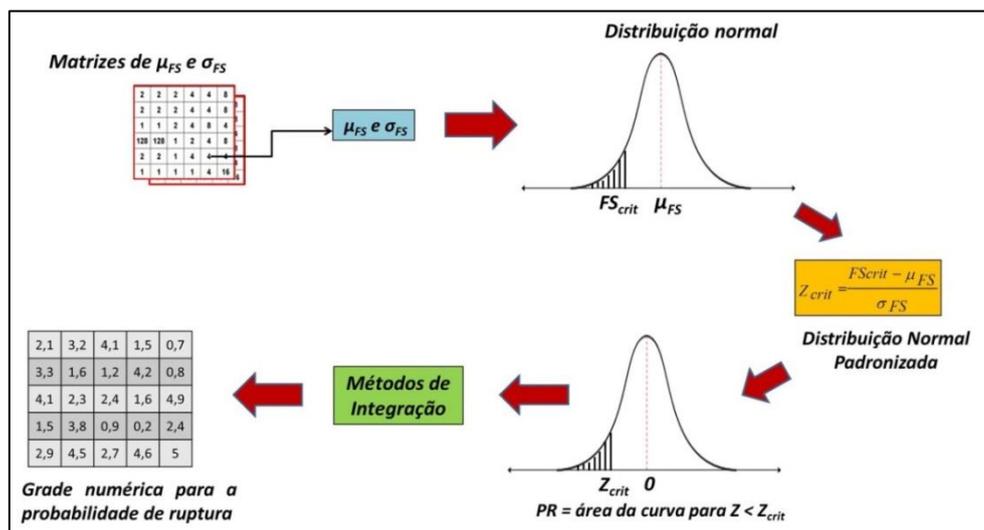


Figura 3.18. Obtenção da PR a partir da função de distribuição de probabilidade.

Uma vez disponíveis os vetores com os valores espaçados da variável independente Z e os pontos da função a integrar, as duas estruturas de dados servirão de argumentos para a função “trapz”, própria do Matlab, a qual retorna, aproximadamente, a integral numérica, via

método trapezoidal. A parte específica desta implementação em Matlab, relacionada ao que foi explicado, encontra-se destacada na Fig. 3.19.

```
398  
399 -         obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.desvio_FS(:,:,k);  
400  
401 -         for i = 1:obj.tam(1)  
402 -             for j = 1:obj.tam(2)  
403  
404 -                 if obj.FS_med(i,j,k) ~= Inf  
405  
406 -                     x = -10:0.01:obj.Zcrit(i,j);  
407  
408 -                     Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);  
409  
410 -                     obj.PR(i,j,obj.cont) = trapz(x,Y);  
411  
412 -                 end % end if  
413  
414 -             end % end for  
415 -         end % end for  
416
```

Figura 3.19. Implementação da obtenção da probabilidade de ruptura.

O uso da função “trapz” representou um ponto de melhoramento do código computacional, em termos de desempenho. Como se verificou que a etapa de integração consumia uma parcela importante do tempo de processamento, tornou-se necessária a verificação da função responsável pela integração que mostrasse uma velocidade aceitável de cômputo frente ao processo previsto. A alternativa de cálculo nas primeiras versões do programa valia-se da função “quad”, também disponível no Matlab, fundamentada na quadratura de Simpson adaptativa.

Ao serem confrontados estes artifícios, o resultado mais satisfatório, quanto à rapidez de processamento, foi promovido pelo método trapezoidal, sendo, a partir disto, adotado em todos os demais modelos que utilizam este tipo de esquema para a quantificação da ameaça por meio da integração da distribuição normal prevista para FS.

3.1.2.3 MODELO SLIDE PROBABILÍSTICO

Seguindo a listagem conjugada com a consequente explanação das funcionalidades do sistema, passa-se ao próximo modelo disponibilizado para uso na avaliação da ameaça de deslizamentos. A metodologia do modelo SLIDE, apresentado por Liao et al. (2010), foi automatizada e modificada para contemplar diferentes modalidades de análises probabilísticas, aumentando o leque de possibilidades quanto ao tratamento dos dados.

A janela voltada para o controle e manipulação das informações desta opção segue a organização nos moldes daquela estabelecida para a alternativa anterior do modelo SHIA_Landslide, possuindo um layout que agrupa separadamente inputs com características similares. Neste sentido, pode-se afirmar que a interface gráfica elaborada é constituída por cinco painéis principais que congregam os seguintes insumos:

- Mapas de dados da área de estudo, onde a mesma é representada matricialmente em certa resolução;
- Informações pluviométricas do evento simulado e valor do fator de segurança aceito como crítico;
- Tipo de análise probabilística que será utilizada e os coeficientes adicionais requeridos por ela;
- Definição dos intervalos de aplicação da chuva, para os quais a probabilidade de ruptura será salva e armazenada;
- Condições tidas como iniciais para o nível freático.

A interface gráfica em questão está especifica na Fig. 3.20.

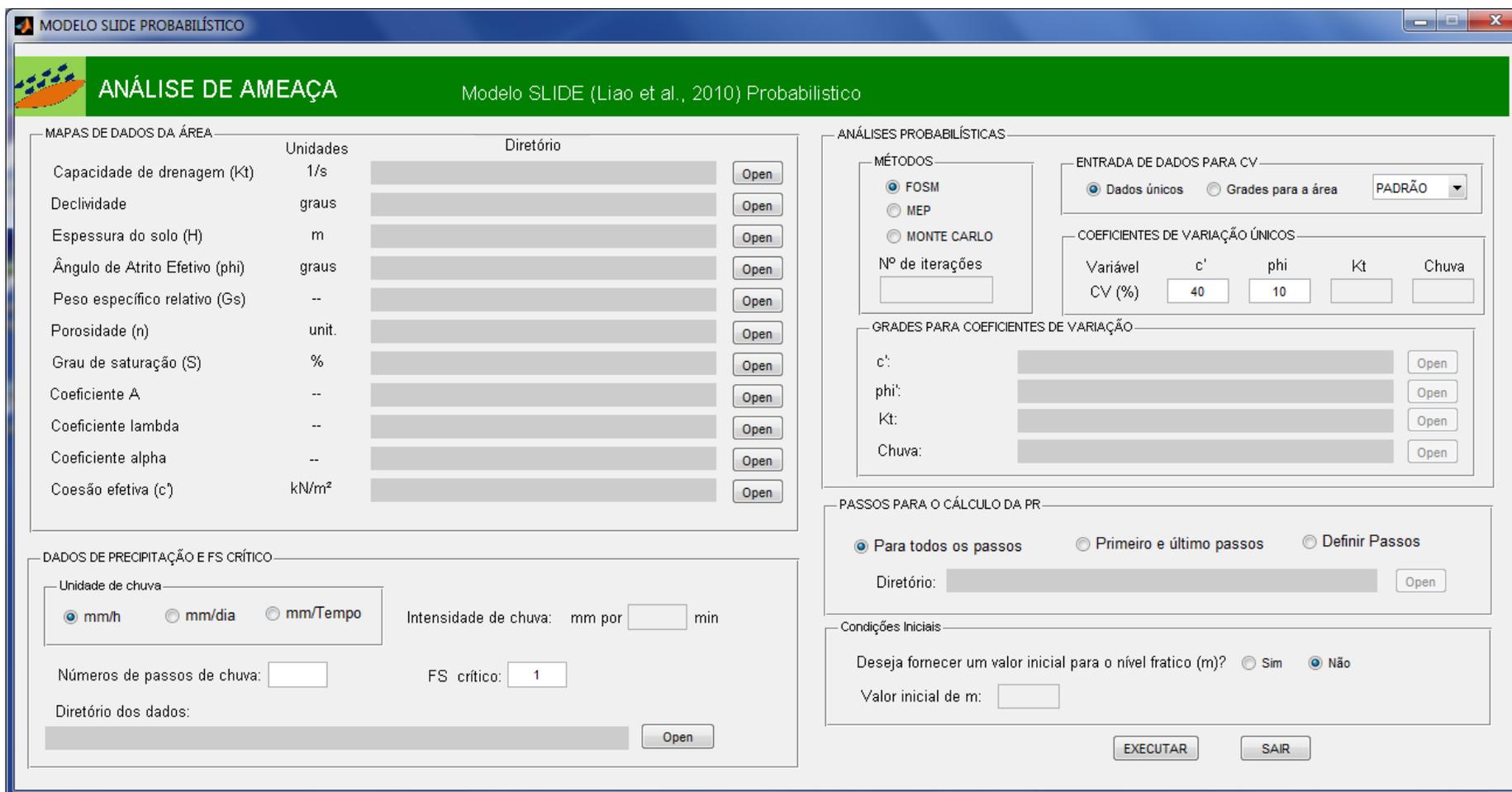


Figura 3.20. Recursos vinculados ao modelo SLIDE agrupadas na interface correspondente.

Os mapas requisitados em forma de grades numéricas estão ligados a parâmetros de caracterização e resistência do solo, assim como a representação da declividade das vertentes oriunda do modelo numérico de terreno. Os parâmetros desta natureza correspondem à lista abaixo:

- Capacidade de drenagem, em 1/s;
- Declividade, em graus;
- Espessura do solo, em m;
- Ângulo de atrito efetivo, em graus;
- Peso específico relativo do solo;
- Porosidade;
- Grau de saturação, em porcentagem;
- Coeficientes A , λ e α , característicos do solo;
- Coesão efetiva, em kPa.

A entrada de dados de precipitação assemelha-se, em parte, àquela proposta no caso das curvas de probabilidade de ruptura. Adicionado ao número de matrizes de chuvas e à indicação do respectivo diretório onde estes dados são armazenados, deve-se estabelecer, agora, a unidade em que a precipitação está representada, escolhendo-se entre as opções de mm/h, mm/dia ou mm/Tempo, onde esta última trata-se de uma alternativa na qual o usuário pode definir a inserção da chuva de forma mais flexível. Para tanto, ao selecionar esta unidade de chuva, automaticamente será liberado o campo para inclusão do tempo, em minutos, fixando, assim, a intensidade de precipitação para os arranjos matriciais impostos à área de estudo. O valor para o fator de segurança crítico também está presente neste painel.

Dentre as análises probabilísticas, encontram-se disponíveis o método das estimativas pontuais, FOSM e de Monte Carlo para serem usados. Associada a estes métodos, apresenta-se a necessidade de fornecer valores dos coeficientes de variação das variáveis responsáveis por impor um cenário de variabilidade às análises. As variáveis que compõem o modelo, escolhidas para figurar com natureza probabilística, correspondem à coesão efetiva, ao ângulo de atrito efetivo, à capacidade de drenagem e à precipitação. Na versão inicial do sistema, estavam disponíveis apenas o método de Monte Carlo e FOSM, sendo que a programação do método das estimativas pontuais foi incorporada posteriormente.

O funcionamento da parte reservada para a seleção e inclusão das informações que compreendem os aspectos probabilísticos ocorre da seguinte forma:

- Inicialmente, escolhe-se um dos três métodos disponíveis, já citados. No caso do método de Monte Carlo, é requerido, ainda, que o número de interações seja proposto;
- A entrada dos coeficientes de variação pode ser por meio de dados únicos ou sob a forma de matrizes com as mesmas dimensões (e, conseqüentemente, a mesma resolução) das demais grades empregadas no estudo. No primeiro caso, os campos para a introdução dos valores dos coeficientes de variação serão liberados, sendo estes representados em porcentagem. Se for escolhido o input sob a forma de grades, os botões correspondentes para a seleção dos arquivos serão disponibilizados;
- Deve-se definir quais e a maneira como serão inseridos os coeficientes de variação para as variáveis consideradas com natureza aleatória no processo. Isso pode ser realizado por dois caminhos: ou pela opção padrão, na qual somente a coesão efetiva do solo e o ângulo de atrito irão compor a análise probabilística, ou pela avaliação personalizada, na qual as quatro variáveis mencionadas anteriormente serão usadas. Para a combinação das seleções de “Dados únicos” e “PADRÃO”, já se encontram definidos os valores de coeficientes de variação em 40% e 10% para a coesão efetiva e o ângulo de atrito, respectivamente. Para “Dados únicos” e “INSERIR”, o usuário deverá entrar com os valores do CV para as quatro variáveis, os quais serão considerados os mesmos para toda a região. A variação espacial dos coeficientes de variação é apreciada quando se escolhe a indicação das grades. Associada à opção padrão, devem-se carregar somente as matrizes de coesão e ângulo de atrito. Já no caso de “INSERIR”, os arquivos de todas as variáveis randômicas se fazem necessárias.

A escolha dos intervalos para os quais as configurações de ameaça, sob a forma de probabilidade de ruptura, serão salvas e passíveis de serem empregadas nas demais etapas do processo, pode ser realizada para todas as configurações de chuvas impostas ao terreno, além das alternativas para o primeiro e último passos ou para intervalos definidos por meio de arquivo específico, ambas já apresentadas no caso do SHIA_Landslide.

Antes que se possa executar o modelo SLIDE, ainda é necessário responder se deseja informar as condições iniciais do nível freático na área de estudo. Em caso positivo, indica-se, no campo apropriado, o valor da variável m presente na formulação proposta, sendo que todo o terreno avaliado estará sob esta condição, antes da execução do procedimento. Caso contrário, o nível freático terá sua posição mais baixa, sendo adotado situando-se na interface solo-rocha (valor de m nulo).

A arquitetura computacional está baseada na programação orientada a objetos. Assim, os códigos e rotinas específicas que são requeridos à efetivação do modelo, bem como as variáveis associadas a estes blocos de comando, estão agrupados em diversas classes, cada uma com uma finalidade definida dentro do esquema global proposto.

Os diversos componentes da GUI são os responsáveis por captar as preferências e os caminhos das pastas que contêm os arquivos requisitados. Uma rotina principal, acionada pelo botão de execução, atua no sentido de importar os dados, adequando suas unidades, direcionando quais objetos serão criados e quais métodos serão ativados com base nas classes estabelecidas, mediante as opções definidas pelo usuário.

Os métodos probabilísticos FOSM e das estimativas pontuais foram representados no algoritmo dentro de uma classe comum, mas por métodos distintos. São estes métodos, que, em última instância, fazem o papel das funções em uma programação orientada a procedimentos, que alteram o estado do objeto mediante o requerimento de aplicação de alguma ação correspondente às análises vinculadas ao sistema. O método de Monte Carlo, por apresentar uma abordagem diferenciada, necessitando de um conjunto de variáveis distinto daquele proposto para os métodos mencionados há pouco e uma implementação que envolve a geração de números aleatórios em sua formulação, ficou programado em uma classe própria. Os atributos, correspondentes às variáveis de cada classe, refletem os dados necessários à efetivação da matriz de ameaça, levando-se em consideração as mais variadas configurações de precipitação que podem ser impostas pelo usuário.

A metodologia do modelo SLIDE, na qual se encontra a atualização do nível freático em virtude do balanço hídrico entre a entrada e a saída de água de cada pixel para os intervalos de tempo, em conjunto com o cálculo dos valores médios do fator de segurança e seus desvios padrões, que definem as distribuições normais de probabilidade advindas dos métodos probabilísticos, estão determinados por métodos que englobam estes dois processos simultaneamente.

Para o método FOSM, executa-se o cálculo do fator de segurança médio, utilizando os valores médios de todas as variáveis independentes randômicas, assim como o desvio padrão a partir das variâncias dos parâmetros de entrada e das derivadas da função que define o FS em relação a cada variável independente, com características aleatórias, para cada uma das células que constituem o terreno.

Em relação ao método das estimativas pontuais, a ação para cada pixel consiste na composição de uma amostra formada por fatores de segurança obtidos da combinação das

estimativas pontuais de cada variável independente aleatória, avaliada na média mais um desvio padrão e na média menos um desvio padrão. Com esta amostra, pode-se calcular o valor médio de FS e seu desvio padrão, que definirão a função de distribuição de probabilidade.

A quantidade de análises executada por célula dependerá do método probabilístico em questão e do número de variáveis aleatórias envolvidas. No caso padrão, onde apenas o ângulo de atrito efetivo e a coesão efetiva do solo são usados, o método FOSM concretizará três análises, enquanto o método de Rosenblueth (1975) contará com quatro cálculos. Quando se introduzem as quatro variáveis independentes, o método FOSM necessitará de cinco execuções, e o método das estimativas pontuais executará dezesseis análises para compor a amostra do FS.

As probabilidades de ruptura são geradas por métodos de classe específicos para este fim, separados das instruções do modelo em si. Então, uma vez definidas as grades de fator de segurança e desvio padrão, estas servirão de insumos para o cálculo da PR, seguindo os mesmos passos já citados anteriormente na seção do modelo SHIA_Landslide probabilístico.

A formulação do método de Monte Carlo conta com o uso da função “normrnd”, presente no software Matlab. Esta função tem por objetivo gerar números aleatórios a partir de uma distribuição normal. Os inputs desta função correspondem exatamente aos valores da média e desvio padrão desta distribuição, além das dimensões da matriz de dados randômicos que será gerada. Deixa-se claro que apenas a distribuição estatística do tipo normal foi utilizada na representação das variáveis independentes com natureza randômica no método de Monte Carlo, resultando, conseqüentemente, na geração de uma distribuição de probabilidade para a variável resposta também normal. A inserção de outras formas de distribuições estatísticas relacionadas às variáveis de entrada corresponde a melhoramentos futuros a serem incorporados ao sistema.

A sistemática da ação prevista pela programação deste método determina que, ao varrer cada posição da matriz da área de estudo, serão sorteados vetores distintos com números aleatórios para cada variável independente definida com um papel no enfoque probabilístico. O tamanho de cada vetor (vinculado à quantidade de valores aleatórios gerados) é determinado pelo número de iterações definidas inicialmente pelo usuário na interface gráfica do modelo.

Assim, para cada posição dos vetores destas variáveis, é tomada a combinação dos valores ali presentes e utilizada na equação do fator de segurança proposta pelo modelo

SLIDE, resultando em um novo grupo de elementos que define uma distribuição normal para a variável dependente.

Uma vez estabelecido o vetor dos fatores de segurança resultantes das combinações das diferentes medidas aleatórias impostas às variáveis independentes, pode-se extrair destes valores de FS, sua média e seu desvio padrão em cada pixel da grade, sendo que, ao fim da execução, estarão criadas duas matrizes nas mesmas dimensões das demais que retratam a área de estudo, armazenando estes dados. Estas estruturas de dados servirão de entradas para o método de classe correspondente à definição das probabilidades de ruptura, cuja programação e operação são iguais àquelas já descritas. O mecanismo de implementação para o método de Monte Carlo está representado na Fig. 3.21.

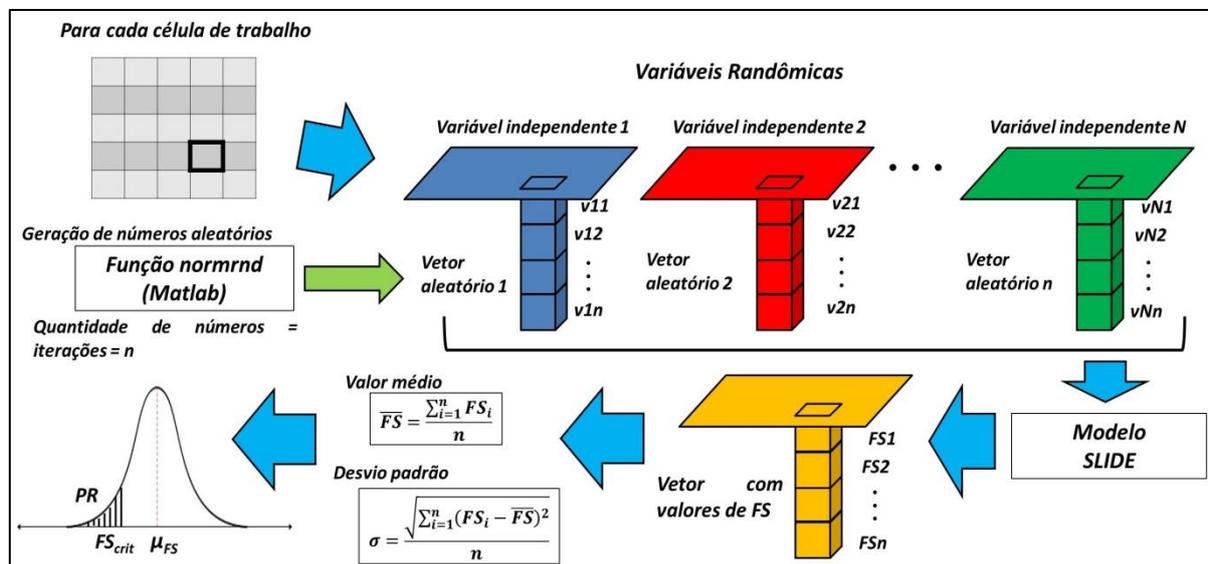


Figura 3.21. Esquematização da implementação do método de Monte Carlo usada para o cálculo do valor médio e desvio padrão de cada célula de trabalho.

3.1.2.4 MODELO DETERMINÍSTICO E PROBABILÍSTICO PARA ALERTA DE INICIALIZAÇÃO DE DESLIZAMENTOS E FLUXO DE DETRITOS

O modelo cuja referência tem o trabalho de Papa et al. (2011) responde ao último modelo a ser explanado dentro da perspectiva de determinação da parcela de ameaça. A visão geral da disposição gráfica e funcional da GUI correspondente possui, em parte, componentes e ações bem próximas daquelas já apresentadas e explicadas em casos anteriormente citados. A GUI relacionada com os componentes responsáveis pelos pontos listados acima pode ser vista na Fig. 3.22.

Modelo para Fluxo de Detritos

ANÁLISE DE AMEAÇA

Modelo para Limiares de Alerta para Inicialização de Fluxo de Detritos (Iverson, 2000; Papa et al., 2011)

TIPOS DE ANÁLISE

Método Probabilístico (FOSM)
 Determinístico (Calibração usando toda área)
 Determinístico (Calibração com número definido de células)
 Nº de células:

VARIÁVEIS ESTÁTICAS E DINÂMICAS

	Unidades	Diretório	
Distritos homogêneos da bacia	--	<input type="text"/>	<input type="button" value="Open"/>
Declividade	graus	<input type="text"/>	<input type="button" value="Open"/>
Área de contrib. pelo compr. (A/b)	m	<input type="text"/>	<input type="button" value="Open"/>
Variáveis Estáticas dos Distritos	--	<input type="text"/>	<input type="button" value="Open"/>
Valores de T, Iz e (Iz)estacionário	--	<input type="text"/>	<input type="button" value="Open"/>
Iz(steady) probabilístico	mm/mês	<input type="text"/>	

DADOS DE PRECIPITAÇÃO E FS CRÍTICO

Unidade de chuva:

mm/h
 mm/dia
 mm/Tempo
 Intensidade de chuva: mm por min

Números de passos de chuva: FS crítico: Valor Dummy:

Diretório dos dados:

PASSOS PARA O CÁLCULO DA PR

Para todos os passos
 Primeiro e último passos
 Definir passos

	Min.	Max.	Interv.
(Iz)estacionário	<input type="text"/>	<input type="text"/>	<input type="text"/>
(Iz)estacionário para o cálculo da PR:	<input type="text"/>		

ANÁLISES PROBABILÍSTICAS

ENTRADA DE DADOS PARA CV

Dados únicos
 Grades para a área
 DEFAULT ▾

COEFICIENTES DE VARIAÇÃO ÚNICOS

Variável	c'	phi	Gama	H	Kz
CV (%)	<input type="text" value="40"/>	<input type="text" value="10"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

GRADES PARA COEFICIENTES DE VARIAÇÃO

c':
 phi':
 Gama:
 H:
 Kz:

Figura 3.22. Elemento gráfico voltado para o uso da simulação de alerta de inicialização de fluxos de detritos.

A interface gráfica encontra-se particionada entre as seguintes parcelas voltadas para input:

- Opções de análise;
- Dados das variáveis estáticas e dinâmicas;
- Informações pluviométricas;
- Definição das grades de PR a serem geradas pelas análises;
- Escolha dos coeficientes de variação a serem empregados no caso da utilização do método probabilístico.

O sistema aqui proposto acompanha, em parte, a organização conferida ao modelo original. A configuração geral do programa permite que o usuário escolha entre três formas iniciais de análise. E é nisto em que consiste o primeiro passo da avaliação por este procedimento. Além da alternativa original que contempla um número específico de células na elaboração das análises, houve a incorporação de duas novas possibilidades de abordagens: uma voltada para a avaliação de todos os pixels da bacia estudada e uma terceira opção, que utiliza o método probabilístico FOSM como núcleo de cálculo.

De forma padrão, ao ser acionado o modelo de Papa et al. (2011), a opção determinística com o uso de todos os pixels da bacia para calibração do modelo é aquela que encontra-se assinalada inicialmente. Automaticamente, à medida que se alterna as opções de análises no painel principal superior, são habilitados ou desabilitados campos e botões específicos para um determinado tipo de alternativa.

Para os modos determinísticos de análise, que podem empregar, tanto um número definido de células, quanto toda a extensão da bacia, a concepção do programa requer que o usuário execute, inicialmente, a ação de fixação dos limiares críticos de precipitação que definem as probabilidades de ruptura a partir de variáveis estáticas e dinâmicas da região em estudo. Este processo é denominado de calibração do modelo e, nas suas condições iniciais, o único botão de ação habilitado refere-se a este procedimento. Chama-se a atenção para que, ao se optar pela utilização de um número definido de células de calibração, este montante deverá ser indicado em um campo correspondente, apontado à direita do painel indicativo das análises.

Uma vez efetivada a etapa de calibração do método, pode-se, então, proceder à previsão da ameaça por meio do emprego dos arquivos com os dados pluviométricos. Esta divisão de operações permite que a primeira parte do processo seja efetuada uma única vez, possibilitando um ganho no tempo de processamento, necessitando apenas da execução do

cálculo da ameaça, quantas vezes fossem necessárias. Caso haja necessidade e seja da vontade do usuário promover uma nova calibração, também existe esta possibilidade.

Os dados de entrada comuns requeridos às três situações de análise estão presentes no painel denominado "VARIÁVEIS ESTÁTICAS E DINÂMICAS" e consistem, basicamente, em configurações sob a feição de grades numéricas, de maneira semelhante às aquelas exemplificadas nos modelos anteriores, sendo que algumas retratam todas as células da área de estudo (região completa), enquanto outras têm a função de arquivar as informações específicas de determinadas variáveis ordenadamente (matriz de dados). Estes arquivos são citados a seguir:

- Matriz representativa da área de estudo, trazendo as informações dos distritos homogêneos sob a forma de numeração sequenciada, onde cada número refere-se a um conjunto de valores de variáveis estáticas que definem um distrito específico;
- Grade de declividade dos taludes para todas as células da bacia analisada (em graus), derivada a partir do modelo numérico de terreno;
- Grade numérica portando os valores para a relação A/b . O procedimento para o alcance desta matriz é composto, inicialmente, pela geração da grade de fluxo acumulado. Esta pode ser obtida por meio dos recursos específicos voltados para processos hidrológicos, que se encontram disponíveis em sistemas de informação geográfica. A matriz A corresponde a grade de fluxo acumulado, multiplicada pela área da célula de trabalho. Uma vez definida esta matriz, ao dividi-la pela dimensão do pixel utilizado (b), esta operação irá fornecer o dado em questão (em termos práticos, para encontrar A/b , basta multiplicar a grade de fluxo acumulado pela dimensão do pixel, considerando a célula quadrada);
- Matriz com os valores das variáveis estáticas, onde a ordem de cada linha se relaciona a um número do encadeamento estabelecido de identificação dos distritos homogêneos, do ponto de vista das características do solo. Nas colunas desta matriz, estão contidas as variáveis estáticas com a seguinte sequência: Z_T , γ_s , ϕ , c , K_x , K_z , D_0 . A quantidade de linhas da referida matriz será equivalente ao número de combinações distintas para os valores das variáveis estáticas mencionadas anteriormente;
- Arquivo portando as informações das variáveis ditas dinâmicas, formado por uma matriz quadrada de ordem 3. As informações das variáveis T , I_z e $(I_z)_{\text{estacionário}}$ estão dispostas na primeira, segunda e terceira linhas, respectivamente. Nas colunas, definem-se, nesta ordem, os valores mínimo e máximo atribuídos para estas variáveis

e a quantidade em que este intervalo será particionado para fins de análise. O programa computacional se encarrega, com base nas informações citadas, de gerar vetores, para cada variável dinâmica, com o número correspondente de elementos determinado pela partição expressa no input, nos quais estarão inseridos os valores igualmente distribuídos, dentro dos limites estabelecidos. Este arquivo é dispensado quando a análise envolve o método FOSM.

Dentro ainda deste painel, existe um campo denominado “ $(I_z)_{\text{estacionário}}$ probabilístico”, o qual se refere a um insumo necessário à análise probabilística, devendo-se definir, neste campo, o valor da chuva antecedente, único, que entrará em conjunto com os registros de precipitações reais no cálculo das probabilidades de ruptura.

Outro conjunto de dados que se faz comum aos três métodos de análises presentes diz respeito às grades, representando a bacia estudada, contendo os dados pluviométricos, voltados para a previsão e definição da feição concernente à ameaça. No entanto, quando se trabalha com as análises determinísticas, a inserção destas informações estará liberada apenas após o procedimento de calibração do modelo. Tanto a estrutura quanto a nomenclatura dos arquivos destas matrizes, seguem a arquitetura antes explicada, presente nas demais opções para o cálculo da ameaça.

Assim como na GUI do modelo SLIDE, as mesmas opções de unidades de chuva podem ser escolhidas, bem como há a necessidade de se indicar a quantidade de configurações de precipitação a serem inseridas e o diretório onde estes dados estão salvos. O fator de segurança crítico poderá ser modificado do valor padrão unitário para uma medida específica, neste ponto da análise.

Seguindo a mesma tendência dos dados de chuva, a estrutura do input para o método probabilístico e para a definição dos intervalos para os quais será calculada a PR é, em síntese, idêntica à apresentada para o método de Liao et al. (2010). Pode-se aplicar, tanto dados únicos dos coeficientes de variação para a região inteira, quanto à distribuição espacial destes elementos em grades armazenadas em arquivos.

Agora, o conjunto de variáveis independentes com natureza probabilística foi alterado, em parte, em relação ao modelo anterior, referindo-se à coesão efetiva, ao ângulo de atrito efetivo, ao peso específico do solo, à espessura do solo e à condutividade hidráulica na direção normal à encosta. Novamente, o usuário terá a opção de considerar apenas a coesão efetiva e o ângulo de atrito do solo como variáveis que possuem variabilidade, em uma abordagem dita padrão pelo sistema (que, no caso de valores únicos, terão CV fixados em

40% e 10%, respectivamente), ou realizar uma análise personalizada que melhor se adapte à realidade do problema, englobando todos os elementos randômicos disponíveis.

Dentro da componente gráfica proposta para a especificação das configurações de PR a serem salvas, faz-se necessário, para o cálculo, que o usuário imponha um $(I_z)_{\text{estacionário}}$ pertencente ao conjunto de valores deste parâmetro, inicialmente proposto no arquivo de variáveis dinâmicas. Para auxiliar na escolha do valor de $(I_z)_{\text{estacionário}}$, o painel traz como novidade a exposição dos limites mínimo e máximo que esta variável pode assumir, assim como a medida do incremento em que este intervalo será repartido, o que permite que se tenha todos os valores possíveis.

Com relação à calibração dos limiares, a partir da leitura dos insumos básicos, o programa desempenha a simulação do modelo, como já dito, para a totalidade das células presentes na área de estudo ou para certo número de elementos definido pelo usuário, levando-se em consideração cada combinação dos dados de entrada das três variáveis dinâmicas, promovendo o cômputo dos seus fatores de segurança.

Para certa quantidade de células, a seleção dos pixels segue de maneira randômica, por meio da geração de pares ordenados de números aleatórios que se referem às posições dos elementos solicitados, obedecendo a uma distribuição uniforme discreta, conseguida pela aplicação da função “randi” do Matlab. Para cada iteração, na qual se fixa os valores de $(I_z)_{\text{estacionário}}$, I_z e T , é verificada a quantidade de elementos instáveis, que apresentam fator de segurança menor ou igual ao valor crítico. A razão entre a quantidade de células instáveis e o número total avaliado fornece a probabilidade de ruptura para uma dada combinação das três variáveis dinâmicas. Similarmente ao que foi mostrado na seção da abordagem original dos modelos, o número total de análises, definido por N_T , conforme Papa et al. (2011), será dado pela equação abaixo:

$$N_T = n \cdot \prod_{i=1}^3 m_i \quad (3.4)$$

onde n relaciona-se ao número de células analisadas e m_i é número de valores a ser utilizado para uma dada variável dinâmica específica.

Para o armazenamento das probabilidades de ruptura resultantes, lançou-se mão da alocação de uma matriz tridimensional, na qual cada linha e coluna representam um determinado valor das variáveis I_z e T , respectivamente, para um valor definido de $(I_z)_{\text{estacionário}}$. Os distintos valores de $(I_z)_{\text{estacionário}}$ mostram-se, de maneira encadeada, como a

terceira dimensão da referida matriz. A coordenação do processo de calibração encontra-se proposto na Fig. 3.23.

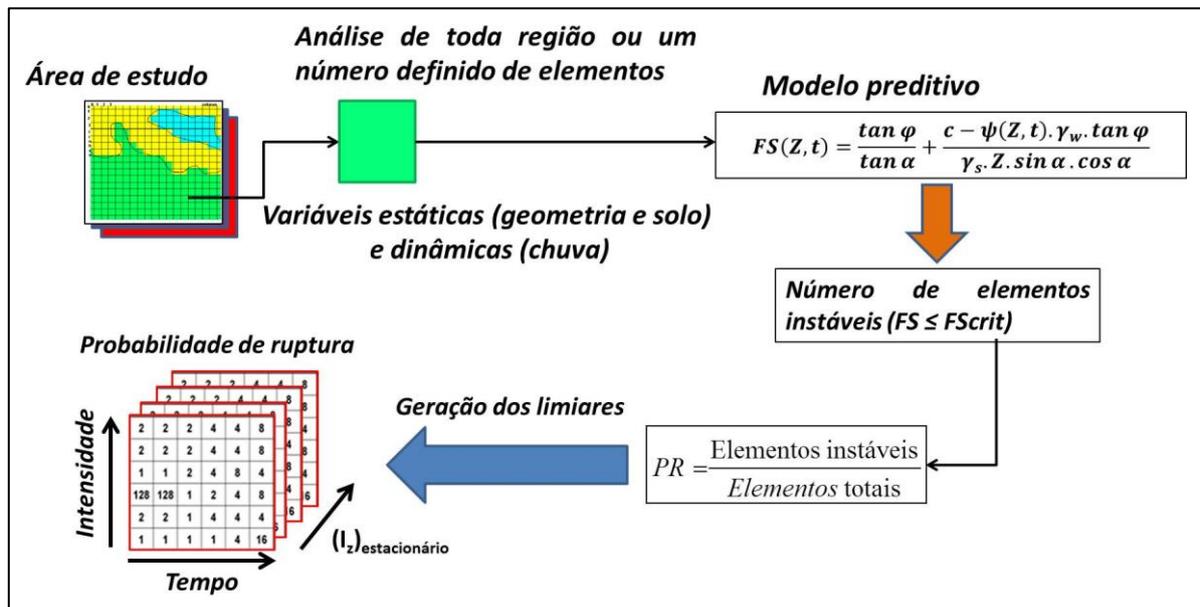


Figura 3.23. Forma de calibração do método de Papa et al. (2011).

A componente de previsão da ameaça opera, para certo valor de $(I_z)_{\text{estacionário}}$, varrendo as matrizes com os dados pluviométricos da extensão avaliada para um dado instante e convertendo estas intensidades de chuvas em probabilidades de ruptura. Tratando-se a probabilidade de ruptura como uma função das variáveis T e I_z , para um $(I_z)_{\text{estacionário}}$ fixo, elabora-se uma interpolação bidimensional na grade que contém os resultados alcançados na etapa anterior, com base nos valores de chuva em uma dada célula e no intervalo de tempo correspondente.

Ao promover esta ação para o valor de $(I_z)_{\text{estacionário}}$ proposto na entrada de dados, o resultado final traz um vetor, cujos elementos são compostos por matrizes com as probabilidades de ruptura previstas para todas as posições da área de estudo e que refletem a ação do progresso das chuvas impostas ao longo de um determinado período.

O número de elementos que forma esse vetor está ligado à quantidade de planos de informação conferidos ao evento pluviométrico, mantendo uma relação direta com a seleção realizada do número de passos da PR a serem concretizados no cálculo da ameaça. Após as matrizes de PR serem estabelecidas, elas estarão disponíveis para visualização/exportação e utilização no arranjo da quantificação do risco. A Fig. 3.24 traz a aplicação do resultado da calibração às informações de chuva para a obtenção da probabilidade de ruptura final.

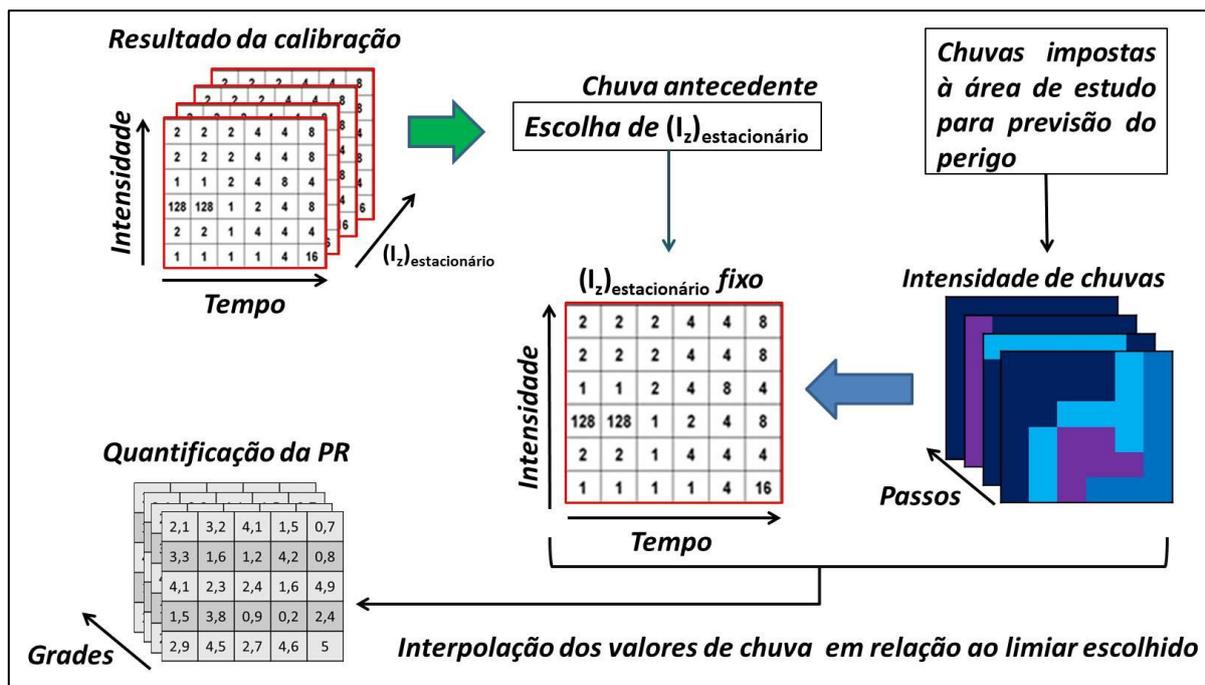


Figura 3.24. Geração dos planos de probabilidade de ruptura para as opções determinísticas.

Em uma implementação anterior, constante na primeira versão do sistema, se procedia a efetivação da análise para todos os valores de $(I_z)_{\text{estacionário}}$ previstos no arquivo das variáveis dinâmicas dos distritos homogêneos, o que resultava em um vetor onde cada posição se referia a um valor de chuva antecedente e armazenava uma matriz tridimensional com as probabilidades de ruptura de cada um dos planos de chuva submetidos ao terreno.

O usuário poderia escolher qual configuração de ameaça seria, então, inserida ao cálculo do risco do deslizamento, pela determinação dos valores de chuva antecedente e do tempo empregados ou apenas da chuva antecedente. Neste último caso, seria considerada a sequência de grades de probabilidade de ruptura resultantes de diferentes momentos e intensidades de precipitações.

Esta abordagem foi modificada para a atual, em virtude de que o procedimento, da maneira como estava montado, gerava um alto custo computacional, dispensando um intervalo de tempo extremamente elevado, dependendo do tamanho da área e da quantidade de informações pluviométricas. Tal fato, em se tratando de um sistema de alerta de deslizamentos, onde o foco corresponde ao fornecimento de uma resposta rápida quanto à previsão dos eventos, não se mostrava tolerável.

A última opção explanada acerca do funcionamento do programa está ligada à utilização do método probabilístico analítico para a averiguação da ameaça, o qual é configurado pela aplicação do método FOSM. Novamente, ressalta-se que existem algumas

particularidades quanto ao input associado a esta seleção, havendo a disponibilidade para a inserção dos coeficientes de variação, em porcentagem, das seguintes variáveis estáticas já citadas: ϕ , c , γ_s , Z_T e K_z . Outra diferença desta alternativa para as demais é que, inicialmente, a variável dinâmica $(I_z)_{\text{estacionário}}$ deverá ser definida, de modo que os processamentos dos cálculos serão efetivados somente para este valor fixo. Uma vez impostas as entradas iniciais, o programa executa a formulação analítica proposta por Papa et al. (2011), agregando a influência da variabilidade dos fatores citados, o que permite a representação da ameaça por meio das probabilidades de ruptura para cada célula da bacia sob estudo, em outputs formados por uma matriz tridimensional (vetor de matrizes).

Similarmente à estrutura obtida para o caso determinístico, estas matrizes tridimensionais podem ser vistas como um vetor cujos elementos são matrizes com as dimensões da área de estudo e que trazem, em suas células, os valores de probabilidade de ruptura. Assim sendo, cada elemento desse vetor é a imagem da consequência, do ponto de vista da ameaça, que um dado episódio imediato de chuva impõe ao terreno. Dentro da configuração composta pelas matrizes de probabilidade de ruptura escolhidas para serem geradas como resultado, reflexo da atuação dos dados pluviométricos, esta cadeia também estará à disposição para a escolha junto à análise de risco.

Com isso, finaliza-se a exposição dos instrumentos empregados na avaliação da ameaça e passa-se a explicar sobre a componente seguinte, que produz o risco devido a deslizamentos.

3.1.3 MODELO DE ANÁLISE DE VULNERABILIDADE

A quantificação da vulnerabilidade proposta por Li et al. (2010), que se fundamenta na dependência entre a intensidade da ameaça e a capacidade de resistência das estruturas envolvidas, apresenta distintas formas para a entrada de dados concernentes a estes dois aspectos básicos do modelo, no sentido de proporcionar um procedimento que mais se adapte às necessidades e recursos do usuário, quanto ao tratamento dos subsídios requisitados.

Nesse sentido, esta seção do programa conta com a possibilidade inicial de escolha entre algumas maneiras principais de input, conforme mostrado na Fig. 3.25.

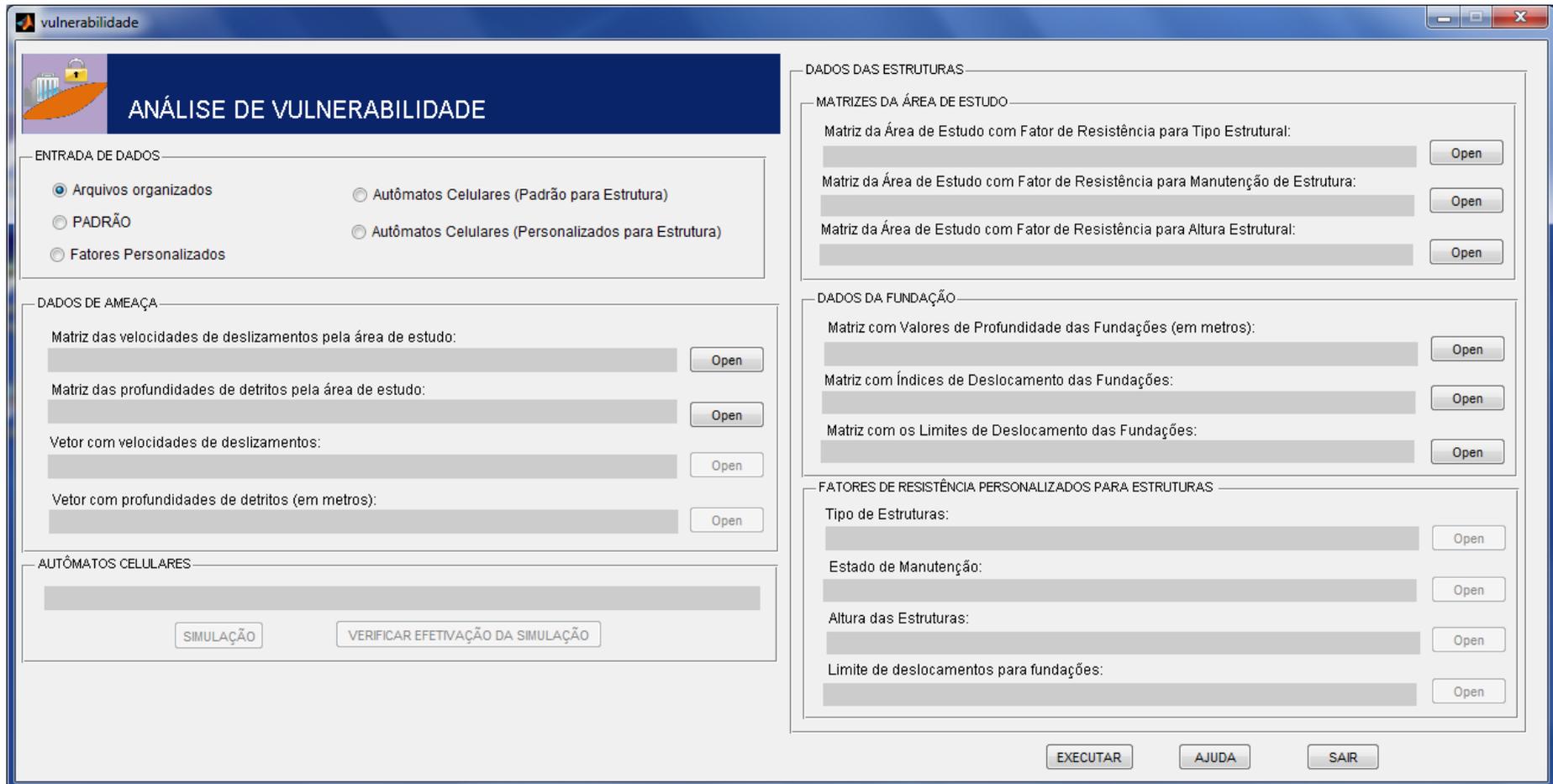


Figura 3.25. Interface gráfica para execução das análises de vulnerabilidade.

Como o modelo se desenvolve em função dos episódios de deslizamentos previstos para ocorrerem devido a um determinado cenário de ameaça e às características das propriedades presentes na trajetória destes eventos, as alternativas implementadas refletem esta abordagem. De forma geral, são requeridas, tanto as grades que trazem o delineamento dos eventos de escorregamentos em conjunto com os valores das respectivas velocidades e espessuras de acúmulo de material, quanto as posições e os atributos dos elementos estruturais contidos na área de estudo.

Enfatiza-se, novamente, que o tratamento da vulnerabilidade dentro da constituição do sistema foi estabelecido idealizando-se variadas formas de entrada de dados, permitindo uma melhor adequação e gerenciamento por parte daqueles que empregam o sistema, no que se refere à demanda e disponibilidade das informações necessárias.

Dentre as cinco alternativas que estão disponíveis no painel superior da interface do cálculo da vulnerabilidade de estruturas, sobre as quais pesa, sobretudo, a estruturação dos dados como fator de destaque, há opções que abrangem desde os casos onde os insumos estão completamente estabelecidos, permitindo a sua incorporação imediata ao sistema, até situações em que a oferta de dados para a região é mínima ou não existe um panorama aceitável de propagação de deslizamentos, sendo necessário empregar valores pré-estabelecidos pela literatura técnica para os parâmetros envolvidos e modelo que simulem o cenário de progressão das ocorrências.

Dentro da representação da superfície sobre avaliação, o fundamento principal consiste no preenchimento apenas das posições das matrizes onde há incidência de elementos em risco e de aspectos relacionados à ameaça. Nos arranjos matriciais empregados, a ausência de informações, as quais se configuram desnecessárias, em um dado local é caracterizada pela inserção de valores nulos nos devidos elementos.

A primeira opção denominada “Arquivos organizados” contempla a posse das grades requeridas para o procedimento pertinente às análises completamente organizadas. Isto significa que os valores de todos os fatores envolvidos no método já devem se encontrar devidamente estruturados e locados espacialmente em matrizes que reproduzam a área de estudo. Assim, os locais com a presença de uma ameaça estabelecida e de estruturas fixadas deverão ter suas características traduzidas conforme solicitado pela metodologia expressa. Listam-se, inicialmente, os arquivos necessários que estão associados à condição de ameaça de movimentos de massa:

- Matriz com o cenário de deslizamentos para a região analisada, o qual estabelece o quadro da ameaça para esta extensão, com a incorporação da velocidade média, em mm/s, dos possíveis eventos. As células que representam a presença de movimentos de massa nesta matriz deverão trazer os valores da velocidade correspondente a estes episódios;
- Grade com a distribuição da localização das áreas de impacto referentes à deposição dos detritos procedentes dos escorregamentos, externas à região da instabilidade. O dado inserido em cada elemento da matriz corresponderá ao próprio valor da espessura da zona de detritos, expresso em metros.

Atreladas às matrizes citadas, adicionam-se aquelas voltadas para a caracterização das estruturas, compostas pelas seguintes informações:

- Índices de deslocamentos conferidos às fundações ou às estruturas situadas na dimensão de estudo, em decorrência da ocorrência de um evento de deslizamento (D_s);
- Valores limites dos índices de deslocamentos que são suportados pelas estruturas ou fundações ($D_{s,T}$);
- Disposição espacial dos valores de profundidade de fundação, em metros, vinculada às estruturas (D_{fod});
- Matrizes associadas à distribuição espacial dos fatores de resistência para o tipo de estrutura (ξ_{sty}), estado de manutenção (ξ_{smn}) e altura (ξ_{shl}) das estruturas presentes.

Quando não há, ainda, avaliações e estudos previamente fixados dos parâmetros solicitados, concernentes aos fatores de resistência das estruturas e aos limites de deformação aceitáveis pelas suas fundações, o usuário poderá recorrer à segunda alternativa sugerida para input, nomeada como “PADRÃO”, na qual estão inseridos os valores recomendados no trabalho de Li et al. (2010), e que funcionam para nortear a definição destes dados. Pautando-se nas classes propostas no artigo técnico para cada uma das distintas particularidades das construções, deve-se atribuir o valor correspondente a estas características mediante a averiguação da composição e da situação estruturais das edificações que formam a região.

Houve alteração, em relação à primeira opção definida, no que diz respeito ao conteúdo das grades matriciais que delimitam o terreno em avaliação. Ao contrário de trazer explicitamente nos seus elementos a grandeza da componente com a qual se está trabalhando, faz-se menção, agora, aos condicionantes da vulnerabilidade de forma indireta, de modo que aquilo expresso nas células é, na realidade, a numeração condizente à codificação utilizada

que permite realizar a concatenação dos elementos destas matrizes com os dados propriamente ditos, dispostos em arquivos próprios. Esta maneira de tratamento das informações encontra-se expressa na Fig. 3.26 trazendo, como exemplo, a medida da profundidade de fundação das estruturas.

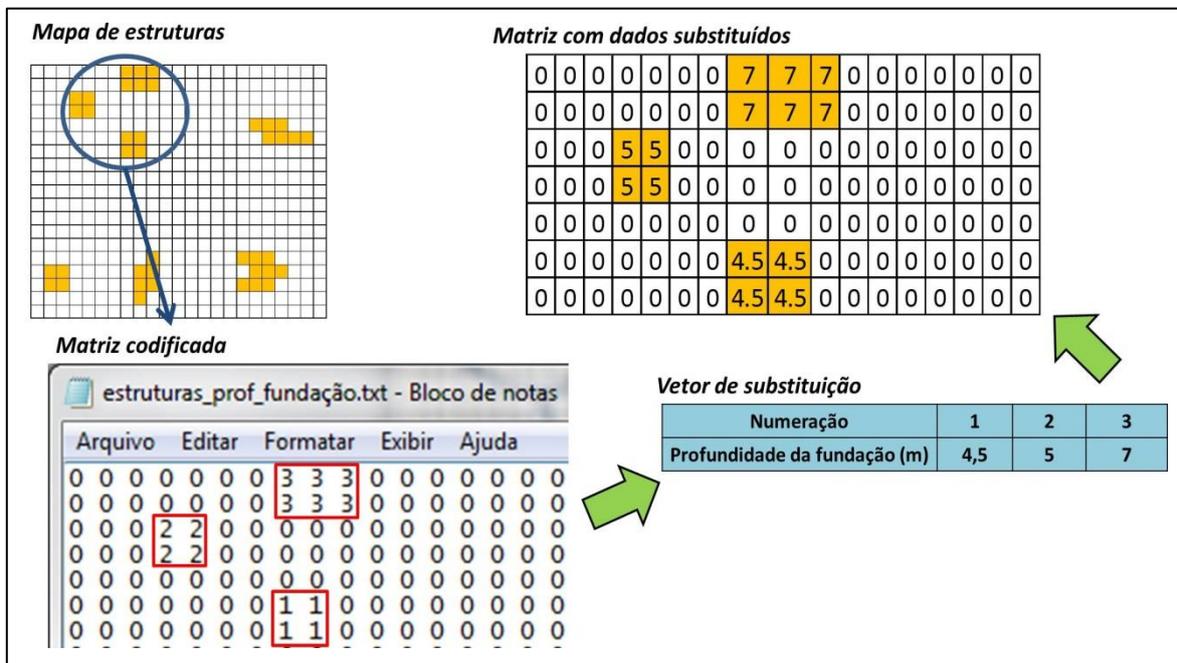


Figura 3.26. Estrutura de input e processo de substituição de dados nas opções que utilizam grades numéricas codificadas.

Assim, as localizações específicas da área avaliada, onde existe a presença de estruturas ou de eventos de escorregamentos, são preenchidas por um sistema numérico sequencial de classes, sendo que estas são redirecionadas aos seus reais valores. Optou-se por esta estruturação, focando-se no aproveitamento da capacidade por parte das plataformas SIG em produzir mapas temáticos com um número arbitrário de classes e com a facilidade de conversão entre os formatos vetorial e matricial, o que contribuiria para a manipulação das informações espaciais e para a sua consequente exportação. As seguintes informações seguem esta linha de arquitetura:

- Matriz portando as classes de velocidades dos deslizamentos que formam o panorama de ameaça;
- Matriz portando as classes de profundidades de detritos para as zonas com possibilidade de serem impactadas pelos eventos de escorregamentos;
- Matriz portando as classes dos valores limites aceitos para os deslocamentos das fundações;

- Matriz portando as classes da numeração dos fatores de resistência concernentes ao tipo de estrutura;
- Matriz portando as classes da numeração dos fatores de resistência referentes ao estado de manutenção das estruturas;
- Matriz portando as classes da numeração dos fatores de resistência para altura das estruturas.

Para os dois primeiros termos mencionados, arquivos suplementares são exigidos contendo os dados das velocidades de deslizamentos e das profundidades de detritos para a substituição das classes criadas em mm/s e metros, respectivamente. Os arquivos são formados por vetores, onde cada elemento constituinte destas estruturas de dados relaciona-se, sequencialmente, às classes descritas.

As grades que comportam a distribuição dos valores para os quatro últimos fatores devem obedecer estritamente à numeração proposta pelas seguintes tabelas: Tab. 3.1, Tab. 3.2, Tab. 3.3 e Tab. 3.4.

Tabela 3.1. Numeração proposta para o fator de resistência da tipologia estrutural.

Tipologia Estrutural	ξ_{sty}	Numeração na matriz
Estrutura muito leve (construção de madeira simples)	0,1	1
Estrutura leve	0,2	2
Estrutura mista (concreto e madeira)	0,4	3
Paredes de tijolos, concreto	0,8	4
Concreto armado	1,3	5
Reforçado	1,5	6

Tabela 3.2. Numeração proposta para o fator de resistência do estado de manutenção.

Estado de manutenção	ξ_{smn}	Numeração na matriz
Muito pobre	0,1	1
Pobre	0,4	2
Médio	0,8	3
Bom	1,2	4
Muito bom	1,5	5

Tabela 3.3. Numeração proposta para o fator de resistência referente à altura.

Altura	ξ_{sht}	Numeração na matriz
Andar único	0,1	1
Edifício baixo	0,4	2
Edifício médio	0,9	3
Edifício alto	1,5	4

Tabela 3.4. Numeração proposta para os limites de deslocamento das fundações (modificado de Li et al., 2010).

Características de deformação	Solo de fundação	Valor limite	Numeração adotada
Estrutura de alvenaria – inclinação parcial da fundação	Compressibilidade média e baixa	0,002	1
	Alta compressibilidade	0,003	2
Estrutura de piso único – recalque (mm)	Compressibilidade média e baixa	120 (apenas para solo com compressibilidade média)	3
	Alta compressibilidade	200	4
Edifício arranha-céu com forma simples – recalque médio da fundação (mm)	Compressibilidade média e baixa	200	5
Edifícios industriais e civis comuns – estrutura em quadros – recalque diferencial entre estacas adjacentes	Compressibilidade média e baixa	0,002 *	6
	Alta compressibilidade	0,003 *	7

Características de deformação	Solo de fundação	Valor limite	Numeração adotada
Edifícios industriais e civis comuns – colunas externas com parede de alvenaria - recalque diferencial entre estacas adjacentes	Compressibilidade média e baixa	0,0007 *	8
	Alta compressibilidade	0,001 *	9
Edifício arranha-céu – $H_g \leq 24$ – inclinação integral	Compressibilidade média e baixa	0,004	10
Edifício arranha-céu – $24 < H_g \leq 60$ – inclinação integral		0,003	11
Edifício arranha-céu – $60 < H_g \leq 100$ – inclinação integral		0,0025	12
Edifício arranha-céu – $H_g > 100$ – inclinação integral		0,002	13
Estrutura em torre - $H_g \leq 20$ – inclinação da fundação	Compressibilidade média e baixa	0,008	14
Estrutura em torre - $20 < H_g \leq 50$ – inclinação da fundação		0,006	15
Estrutura em torre - $50 < H_g \leq 100$ – inclinação da fundação		0,005	16

Características de deformação	Solo de fundação	Valor limite	Numeração adotada
Estrutura em torre - $100 < H_g \leq 150$ – inclinação da fundação	Compressibilidade média e baixa	0,004	17
Estrutura em torre - $150 < H_g \leq 200$ – inclinação da fundação		0,003	18
Estrutura em torre - $200 < H_g \leq 250$ – inclinação da fundação		0,002	19
Estrutura em torre - $H_g \leq 100$ – recalque da fundação (mm)	Compressibilidade média e baixa	400	20
Estrutura em torre - $100 \leq H_g < 200$ – recalque da fundação (mm)		300	21
Estrutura em torre - $200 \leq H_g < 250$ – recalque da fundação (mm)		200	22

* (em mm) distância central entre estacas adjacentes; H_g (em m) altura da construção acima do solo.

Da mesma maneira usada na primeira configuração de entrada de dados, alguns arquivos que retratam a dimensão completa da superfície estudada contêm, diretamente, as próprias informações inseridas nos elementos matriciais. Duas matrizes da entrada de dados mantêm o formato primitivo e carregam as verdadeiras informações associadas a uma localização do terreno. São elas:

- Distribuição espacial da profundidade das fundações (D_{fod}), em metros;

- Índices de deslocamentos ou distorções apresentadas pelas fundações das estruturas da área de estudo (D_s), em conformidade com os critérios dos limites propostos por Li et al. (2010), em termos de características do solo e da deformação da infraestrutura, assim como de suas respectivas unidades de medida.

Há, também, uma alternativa denominada “Fatores Personalizados”, que contempla as situações em que, a partir de pesquisas próprias voltadas para a caracterização das edificações inclusas no estudo, já se encontram catalogados os fatores de resistência das estruturas e os limiares fixados sobre as máximas distorções suportadas pelas fundações, não necessitando, portanto, empregar os valores citados no trabalho de Li et al. (2010).

Nestas condições, a arquitetura do input acompanha o mesmo esquema conferido à segunda opção explicitada acima, com a permanência, inclusive, de todos os arquivos listados como necessários para a referida escolha. É preciso, ainda, incorporar quatro novos vetores como dados de entrada para comportar os registros numéricos correspondentes às características de tipologia estrutural, altura, estado de manutenção e limites das deformações toleradas pelas fundações.

A quarta e a quinta possibilidades, expressas por “Autômatos Celulares (Padrão para Estrutura)” e “Autômatos Celulares (Personalizados para Estrutura)”, têm ambas, como foco, o uso de modelagem para proporcionar a simulação do desenvolvimento de eventos de deslizamentos, partindo de uma configuração inicial da ameaça que venha afetar a região avaliada.

Isso significa que este processo irá resultar na geração automática das matrizes, contendo a delimitação da trajetória das ocorrências de escorregamentos rasos em conjunto com os valores de velocidade da massa deslizante e, nas áreas de acúmulo de material, de espessura de detritos. O que as diferencia é quanto aos dados das estruturas que deverão ser inseridos, os quais se assemelham às opções “PADRÃO” e “Fatores Personalizados”, explicados acima.

O modelo projetado para esta finalidade foi idealizado pela combinação de três pesquisas, cuja base corresponde à aplicação de autômatos celulares para o delineamento do alcance dos fenômenos de movimentos de massa.

Quando se seleciona uma destas opções mencionadas, o painel intitulado “AUTÔMATOS CELULARES” habilita o botão previsto para a simulação. Ao pressioná-lo, uma nova interface surge com os campos necessários para preenchimento, a fim de que a

execução desta tarefa seja realizada. Os detalhes de implementação e funcionamento, tanto do modelo, quanto desta nova GUI, estão explicitados em uma seção específica mais adiante.

Adicionado a isso, uma mensagem é informada, em um campo específico presente no painel da GUI principal de vulnerabilidade, anunciando que a etapa de avaliação pelos autômatos celulares foi acionada e solicitando que o usuário verifique se a simulação que prevê a velocidade e altura de detritos foi realmente praticada.

Se a operação transcorrer com sucesso, uma nova mensagem anunciará a disponibilidade da avaliação completa da vulnerabilidade, com a consequente liberação do botão “EXECUTAR”, na parte inferior direita da interface. Caso contrário, será informado que a execução não foi bem sucedida, solicitando que aquele que utiliza o sistema refaça a análise pelo modelo.

Com relação ao algoritmo desenvolvido, uma vez que todos os dados de entrada estejam carregados, o código computacional gera, inicialmente, uma matriz nula, onde serão salvos os resultados da análise de vulnerabilidade, com as mesmas dimensões e resolução das grades que representam a área de estudo.

Segue-se a averiguação, a partir da varredura linha a linha das matrizes onde estão armazenados os fatores de resistência, de quais dos seus elementos compreendem edificações.

Ao encontrar uma posição não nula, que indica a instalação de uma construção, o algoritmo realiza a distinção quanto ao método de cálculo aplicado na determinação da vulnerabilidade, ou seja, verifica se a edificação encontra-se no interior da massa instável ou nas zonas sujeitas à deposição de detritos provenientes dos eventos de movimentos de massa, preenchendo, desta forma, a posição relativa da matriz criada com o valor computado de vulnerabilidade.

Caso a estrutura situe-se exterior à influência de qualquer ação nociva de ameaça, a vulnerabilidade atribuída é considerada nula. Ao fim do procedimento, a matriz de vulnerabilidade apresentará, na sua composição, a distribuição espacial da quantificação pela região analisada da vulnerabilidade concernente aos elementos em risco.

Na Tab. 3.5 encontra-se o resumo dos insumos requisitados em cada enfoque. Na próxima seção, passa-se a explicar as especificidades existentes na programação realizada para a estimativa da velocidade dos deslizamentos e da espessura de detritos.

Tabela 3.5. Lista de informações correspondente às alternativas de input para vulnerabilidade.

Opções de input	Informações		Descrição
	Ameaça	Estruturas	
Arquivos organizados	<ul style="list-style-type: none"> • Matriz com as velocidades dos deslizamentos (mm/s); • Espessura de detritos acumulado (m); 	<ul style="list-style-type: none"> • Matrizes com os índices de deslocamento das fundações (D_S) e seus limites ($D_{S,T}$); • Matriz com a profundidade das fundações (m); • Matrizes com a numeração das classes para os fatores de resistência de altura, manutenção e tipo estrutural; 	Grades totalmente organizadas;
PADRÃO	<ul style="list-style-type: none"> • Matriz com a numeração das classes de velocidades dos deslizamentos; • Vetor com valores de velocidade de deslizamento (mm/s) para substituição; • Matriz com a numeração das classes de espessura de detritos acumulado; • Vetor com valores de espessura de detritos (m) para substituição; 	<ul style="list-style-type: none"> • Matriz com os índices de deslocamento das fundações (D_S); • Matriz com a profundidade das fundações (m); • Matriz com a numeração das classes para limites pré-estabelecidos de deslocamento; • Matrizes com a numeração das classes para os fatores de resistência pré-estabelecidos de altura, manutenção e tipo estrutural; 	Uso de parâmetros para estruturas propostos por Li et al. (2010);
Fatores Personalizados	<ul style="list-style-type: none"> • Matriz com a numeração das classes de velocidades dos deslizamentos; • Vetor com valores de velocidade de deslizamento (mm/s) para substituição; • Matriz com a numeração das classes de espessura de detritos acumulado; 	<ul style="list-style-type: none"> • Matriz com os índices de deslocamento das fundações (D_S); • Matriz com a profundidade das fundações (m); • Matriz com a numeração das classes para limites de deslocamento; 	Parâmetros introduzidos pelo usuário;

Opções de input	Informações		Descrição
	Ameaça	Estruturas	
Fatores Personalizados	<ul style="list-style-type: none"> • Vetor com valores de espessura de detritos (m) para substituição; 	<ul style="list-style-type: none"> • Matrizes com a numeração das classes para os fatores de resistência de altura, manutenção e tipo estrutural; • Vetores com valores dos fatores de altura, manutenção e tipologia estrutural para substituição propostos pelo usuário; • Vetor com limites de deslocamento propostos pelo usuário; 	Parâmetros introduzidos pelo usuário;
Autômatos Celulares (Padrão para Estrutura)	Dados de entrada definidos pelo tipo de simulação escolhida	Igual para opção “PADRÃO”	Análise de ameaça pela aplicação de autômatos celulares e parâmetros das estruturas pré-estabelecidos;
Autômatos Celulares (Personalizados para Estrutura)	Dados de entrada definidos pelo tipo de simulação escolhida	Igual para opção “Fatores Personalizados”	Análise de ameaça pela aplicação de autômatos celulares e parâmetros das estruturas estabelecidos pelo usuário;

3.1.3.1 GERAÇÃO AUTOMÁTICA DAS TRAJETÓRIAS, VELOCIDADE E ÁREA DE DEPOSIÇÃO DE ESCORREGAMENTOS

O emprego da metodologia para o cálculo da vulnerabilidade proposta por Li et al. (2010), admite a necessidade da listagem e existência das grades contendo a quantificação das velocidades dos escorregamentos dentro da sua trajetória, assim como a localização das zonas a serem impactadas em conjunto com a espessura dos detritos nestes pontos.

Como mostrado até então, o usuário deverá incorporar essas informações totalmente organizadas ao sistema para, que haja o processamento desta etapa da avaliação do risco. Isso faz com que aquele que utilize o sistema necessite, com base em pesquisas anteriores e registros históricos, realizar um estudo detalhado no sentido de estimar estes dados. No entanto, esse tipo de ponderação requer certa experiência por parte de quem a executa, sendo dependente do julgamento do especialista envolvido no processo e, de certa forma, apresentando-se como um método de caráter subjetivo.

Na tentativa de transpor esta dificuldade, tornando o procedimento automatizado e com elementos mais racionais, desenvolveu-se uma rotina computacional capaz de modelar os três tópicos principais envolvidos neste problema. Para tanto, abordagens presentes em distintos trabalhos pesquisados, até então atuando independentemente e não estando relacionados inicialmente, foram combinadas em um ambiente de análise bidimensional, permitindo-se verificar a evolução dos cenários vinculados aos eventos de deslizamentos.

As pesquisas estabelecidas como ponto de partida para a criação do programa foram trabalhos envolvendo simulação de escorregamentos com o auxílio de autômatos celulares relativos à Lai & Dragičević (2011), cômputo da velocidade do fluxo por meio de um modelo de adensamento aplicado em processos de deslizamentos descrito por Qiao & Clayton (2013) e pesquisa de Gruber & Peckham (2009), que descrevem o método de direções múltiplas de fluxo aplicado na transferência da altura de detritos entre células. Os detalhes dos métodos originais foram expostos no capítulo de revisão bibliográfica, juntamente com os demais modelos. Serão mostrados, a seguir, os artifícios de estruturação e da lógica que estão embutidos na rotina criada, visando a um melhor entendimento de como o programa atua.

Antes, no entanto, torna-se necessário fazer uma observação no sentido de explicar que, apesar de situar-se vinculada à definição da vulnerabilidade, a modelagem dos autômatos celulares poderia, também, estar perfeitamente, ou até melhor, encaixada junto ao cálculo da ameaça. Mas, em função da dificuldade encontrada em termos de se definir a trajetória e

características dos eventos de deslizamento, cruciais para o caso da vulnerabilidade, optou-se por associar este processo à presente etapa.

Inicialmente, explica-se como a trajetória das ocorrências é estabelecida. A região finita de aplicação do autômato celular é definida por uma matriz de mesmas dimensões da área de trabalho. Assim, uma das entradas básicas requeridas pelo programa trata-se de uma grade regular nestes moldes, onde estão assinaladas as células que serão consideradas fontes dos deslizamentos rasos a serem simulados.

A ideia principal do modelo concentra-se em verificar quais células, dentre aquelas vizinhas de um dado pixel que contém um escorregamento raso, satisfazem as condições para que o evento possa ser propagado. Para tanto, a própria vizinhança de cada elemento potencial a receber o deslizamento é verificada quanto a certas exigências. Esta verificação é efetivada para cada passo de tempo, correspondendo a uma iteração, referente à aplicação destes princípios de transição e, por conseguinte, à atualização da nova área do deslizamento. A vizinhança adotada para análise refere-se ao modelo de Moore com raio unitário. Os requisitos observados, podendo ser ponderados como as leis de transição de estado para o autômato celular, são os seguintes:

- Altimetria: compara-se a altitude do elemento central com a altimetria de cada vizinho, sendo que, para que exista a transferência para o pixel avaliado, o valor do seu MNT deve ser menor que aquele da vizinhança;
- Presença de escorregamento: o critério da altimetria é necessário, mas não suficiente para a ocorrência do processo. Além de o vizinho referir-se a um local mais alto que o candidato central, aquele deve estar assinalado por conter um deslizamento. O fenômeno só pode ser propagado a partir de uma célula onde já exista;
- Curvatura das vertentes: a curvatura do elemento analisado deve ser satisfatória, no que se refere à predisposição ao estabelecimento dos movimentos de massa. A definição de quais configurações são aquelas mais propensas a deflagrarem ocorrências podem ser indicadas pelo usuário, definidas por meio de um estudo específico para a área em questão, ou podem-se utilizar os dados citados no trabalho de Lai & Dragičević (2011), considerados como opções padrões;
- Declividade: é estabelecido um limite para diferenciar células que, cumprindo todas as requisições anteriores, serão classificadas como escorregamentos (aquelas com inclinações maiores que este valor) ou como áreas de deposição de detritos (declividade inferior ao valor imposto). Como no caso da curvatura, este índice poderá

ser definido como um dado de entrada ou se utilizará o mesmo proposto por Lai & Dragičević (2011).

A partir do exposto, os elementos da grade que armazena a trajetória dos deslizamentos, correspondente ao autômato celular, podem assumir quatro valores válidos para os estados admissíveis das células entre as situações inicial e final:

- Para deslizamentos atuais, passíveis de serem transferidos à células vizinhas, atribui-se valor 1;
- Deslizamentos já propagados em iterações anteriores, impõe-se o valor 2;
- Aos pixels classificados como movimentos de massa e com declividades abaixo do limite estabelecido, é atribuído o valor 3 (áreas de deposição que não propagam deslizamentos);
- Células livres de escorregamentos carregam o valor nulo.

A Fig. 3.27 traz um exemplo de simulação do processo executado pelo autômato celular, com base em um conjunto de células iniciais classificadas como fonte dos episódios de movimento de massa.

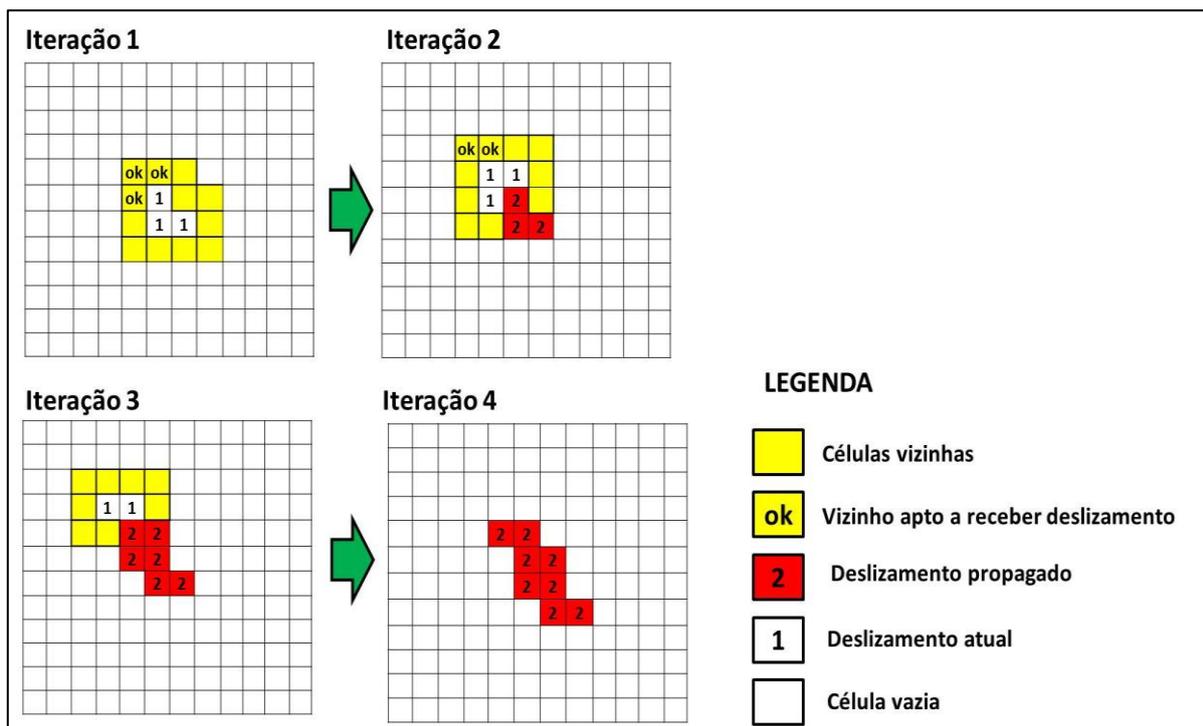


Figura 3.27. Exemplo para processo de iterações realizadas pelo autômato celular na definição da trajetória dos deslizamentos.

Torna-se necessário que a matriz inicial represente as fontes dos escorregamentos como ocorrências atuais, ou seja, nas posições onde se crer que se iniciam os eventos, estes

pontos devem ser representados por valores unitários. Na primeira iteração executada, as células vizinhas às fontes são identificadas e definidas como candidatas a pertencerem à trajetória do deslizamento. O procedimento de identificação da vizinhança é realizado completamente para todos os pontos de escorregamentos ativos, com o auxílio da função que calcula a convolução bidimensional existente no Matlab.

Após isso, o algoritmo percorre toda a matriz à procura dos pixels classificados como vizinhos. Ao encontrar uma destas células, a rotina averigua as regras de transição de deslizamento para o estado do elemento em questão, com relação à sua própria vizinhança. Para um dado passo da evolução do autômato celular, todos os elementos definidos como vizinhança de células de escorregamento são sujeitos aos critérios mencionados anteriormente.

A situação para uma célula analisada, com destaque para a disposição de seus elementos adjacentes está retratada pela Fig. 3.28.

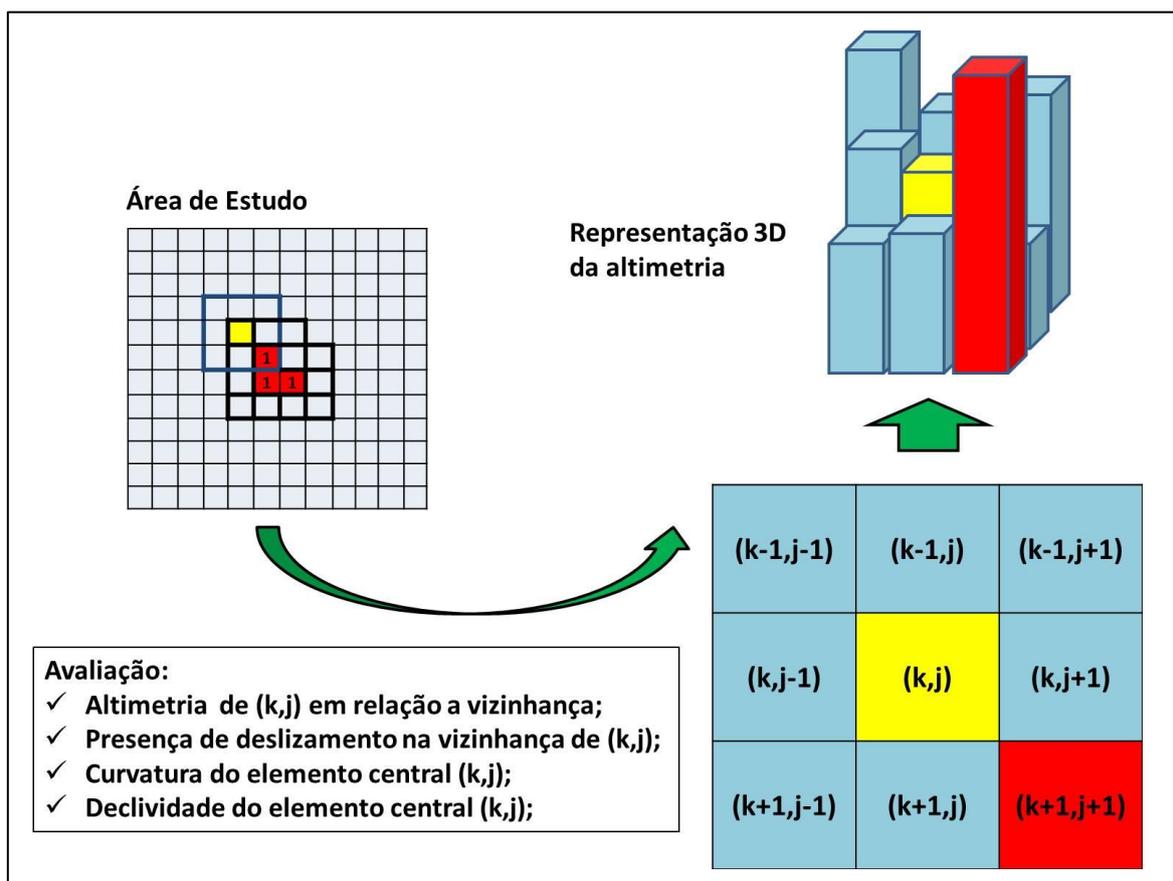


Figura 3.28. Disposição das células contíguas ao elemento avaliado onde são aplicados os critérios pra mudança de estado do autômato celular.

Se uma célula avaliada possui todos os requisitos para que o deslizamento seja transferido para este ponto ou para ser apreciada como uma área de deposição, então este pixel é marcado como tal.

Ao fim de cada iteração, depois que toda a vizinhança é avaliada e aqueles elementos aptos a serem considerados deslizamentos são marcados, as células que figuravam como escorregamentos atuais passam para a condição de escorregamentos propagados. As novas células vizinhas que terão suas características analisadas no passo posterior serão aquelas definidas a partir dos pixels marcados como escorregamentos atuais neste passo.

O procedimento se repete até ser atingido o número de repetições previsto pelo usuário ou até que seja observado que, a partir de um dado momento, não há mais geração de vizinhos de escorregamentos que poderiam se assinalados como potenciais eventos e que deveriam ser averiguados quanto às regras de propagação. Isto se concretiza quando, na vizinhança de deslizamentos ativos, nenhum dos pixels atende aos critérios estabelecidos, o que faz com que no instante seguinte não haja nenhuma célula classificada como escorregamento atual (representada pelo número 1) e, portanto, a matriz responsável por armazenar os elementos da vizinhança é nula. Esta situação está retratada pelas duas últimas iterações presentes na Fig. 3.27. O controle deste critério de parada é estabelecido por uma variável booleana, que monitora quanto à existência, ainda, de vizinhos em um passo iterativo. Desde que a variável não acuse mais a presença de dados desta natureza, permanecendo nula, o processo de atualização do autômato celular encerra-se.

A estimativa da espessura da camada de detritos acumulada consiste em um cálculo paralelo executado pelo programa, de forma coordenada à definição das células de escorregamento, sendo que o artifício utilizado baseia-se no método de direções múltiplas de fluxo.

Segundo esse método, exposto por Gruber & Peckham (2009), a massa presente em uma célula será propagada mediante a sua distribuição entre os seus oitos vizinhos. A fração da massa (d), que é transferida a uma determinada célula vizinha N_{Bi} , é função da declividade da própria vizinhança do escorregamento e pode ser descrita conforme a seguinte equação:

$$d_{NBi} = \frac{\tan(\beta_{NBi})^v}{\sum_{j=1}^8 [\tan(\beta_{NBj})^v]} \quad (3.5)$$

onde:

β consiste no valor da declividade da célula, sendo considerada positiva para elementos mais baixos que aquele em que se encontra a massa a ser propagada, e nula para células com altimetria maior;

v refere-se ao expoente que controla a dispersão do fluxo. Conforme Gruber & Peckham (2009), valores elevados deste parâmetro concentram o fluxo para a direção da descida mais íngreme, enquanto valores baixos resultam em uma dispersão mais forte, devendo v ser maior ou igual à zero.

Gruber & Peckham (2009) ressaltam que, na abordagem denominada TOPMODEL, proposta por Quinn et al. (1991), não considera-se nenhum expoente no controle da dispersão ($v = 1$), enfoque que foi adotado neste trabalho;

A atuação do algoritmo procede no sentido de quantificar a fração de massa, em termos de altura de detritos, que será transferida quando se constata a presença de uma célula vizinha de um deslizamento capaz de comportar a propagação desta ocorrência. Nestas condições, assume-se que o pixel central é aquele que contém o escorregamento, de maneira que o denominador da Eq. 3.5 é calculado considerando-se as oito células em torno deste elemento, tendo-se o cuidado, ainda, de aplicar a restrição que considera, no somatório das declividades, apenas os dados das células com altimetria menor que aquela que contém o escorregamento. O numerador da equação será definido tomando como referência a declividade da célula vizinha, avaliada na respectiva iteração. Isso garante a preservação da massa que será distribuída, já que a soma das parcelas iguala-se à unidade. O processo se desenvolve conforme indicado na Fig. 3.29.

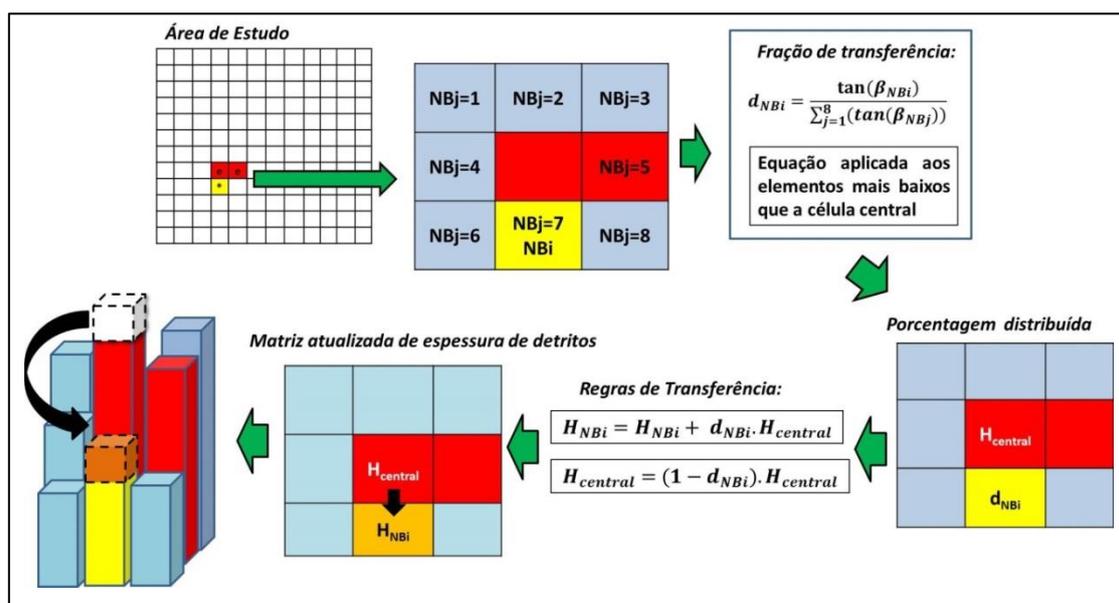


Figura 3.29. Esquema utilizado para transferência de material entre duas células adjacentes.

Uma vez estabelecida a fração que será repassada para a célula vizinha, o programa encarrega-se de atualizar as medidas da espessura de detritos, tanto no pixel que contém o escorregamento, quanto naquele que irá receber esta porcentagem mediante as seguintes regras:

$$H_{NBi} = H_{NBi} + d_{NBi} \cdot H_{central} \quad (3.6)$$

$$H_{central} = (1 - d_{NBi}) \cdot H_{central} \quad (3.7)$$

A célula avaliada tem a sua altura de material H_{NBi} atualizada, incorporando a medida já existente àquela advinda do deslizamento, representada pelo produto entre a proporção estipulada d_{NBi} e a espessura presente no pixel com deslizamento $H_{central}$. Por sua vez, a altura do elemento que contém o deslizamento é reduzida no mesmo montante da quantidade transferida, conforme a Eq. 3.7.

Ao se estudar os oito vizinhos de um membro dos candidatos a se tornarem trajetória da ocorrência, enquanto houver um elemento adjacente marcado como escorregamento, nesta redondeza, que atenda aos fundamentos de classificação empregados no autômato celular, o processo explanado é repetido, permitindo que o fluxo de material de dois ou mais componentes convirja para a célula correspondente.

No início do processo, as informações da matriz de espessura do solo são duplicadas para uma variável similar, na qual são procedidas as manipulações definidas acima, e que resultam na alteração da camada inicial de material inconsolidado nas zonas de ruptura. Findada a estimativa da deposição, a subtração entre essa matriz e a grade de espessura original de solo, nesta ordem, fornece as áreas de ganho (com valores positivo) e perda (com valores negativos) de material.

Como opção complementar oferecida ao usuário, foi implementada a possibilidade de definição do valor mínimo de ganho de detritos a ser considerado como área de acúmulo. Assim, em regiões que apresentam perda de solo superior ao limite numérico estabelecido, a rotina computacional considera como nula a altura de material depositado, o que permite diferenciar, dentro da trajetória delimitada pelo autômato celular, onde o escorregamento tende a fluir daqueles locais atingidos por deposição. Na metodologia para a avaliação da vulnerabilidade proposta por Li et al. (2010), esta discriminação é crucial para sua aplicação.

O terceiro aspecto incluído na estruturação do algoritmo está associado ao arranjo que propõe a modificação da abordagem apresentada por Qiao & Clayton (2013). Para atender às particularidades impostas à avaliação da velocidade de um fluxo de detritos em condições

bidimensionais, foi necessário impor uma adaptação ao procedimento original que, apesar de possibilitar a inserção de variadas inclinações e comprimentos por trecho na representação de um talude, trabalha com uma seção transversal única para cada escorregamento, calculando a velocidade por segmento, de maneira a verificar a continuidade do fenômeno.

A implementação aqui efetivada foi concebida para que em cada célula classificada como pertencente ao evento houvesse, em primeiro lugar, o estabelecimento da estimativa do tempo necessário para o fluxo percorrer a extensão do pixel, mediante as características da célula estudada, bem como levando em consideração as condições presentes nos elementos em torno daquele considerado.

A Fig. 3.30 esquematiza o procedimento para obtenção da velocidade em um ponto circundado por três elementos categorizados como escorregamentos.

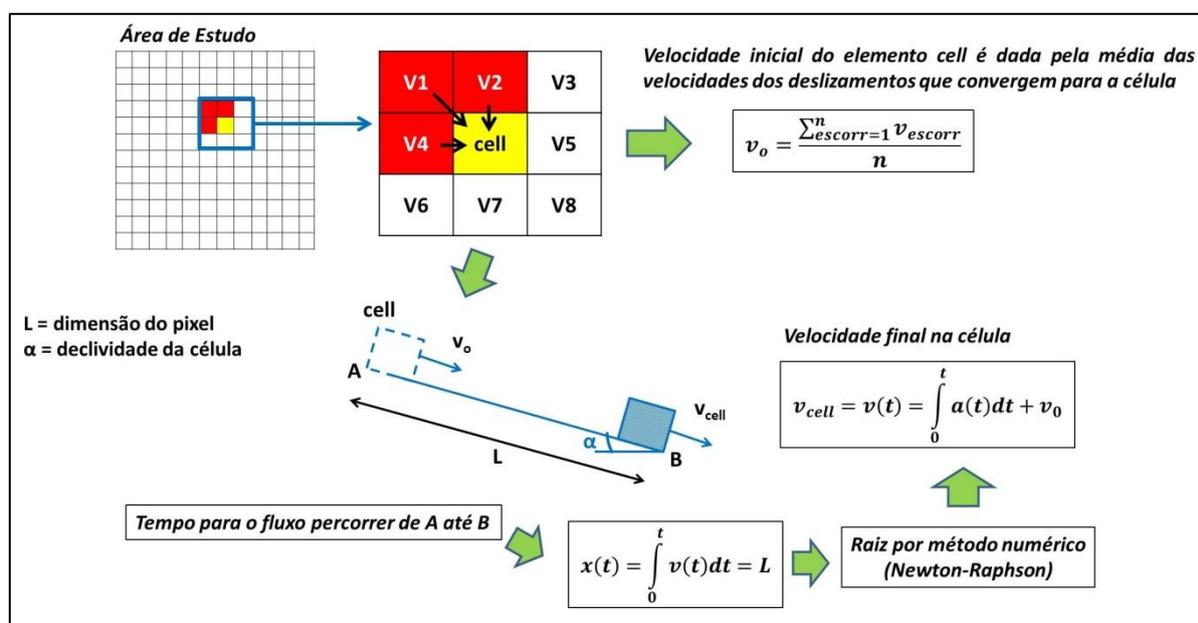


Figura 3.30. Esquema utilizado para o cálculo da velocidade média em cada célula de escorregamento.

No caso exemplificado pela figura acima, o elemento denominado como cell encontra-se sob a influência de três células de escorregamento que a circundam (células V1, V2 e V4), as quais irão transferir a este pixel central uma parcela de material, cada uma com determinada velocidade. Para não impor uma maior ou menor estimativa à velocidade inicial v_0 , realizou-se a programação para o cálculo da média, com base nos registros das células portadoras dos movimentos de massa, que foi estipulada como sendo v_0 para o elemento de fluxo que percorrerá o comprimento de cell.

Tomando as informações da célula estudada, é possível estabelecer as medidas iniciais para o excesso de poropressão no perfil transversal, assim como os termos que irão compor as equações que definem, tanto o espaço percorrido, quanto a velocidade em função do tempo. Na realidade, o que se faz é considerar cada pixel como uma encosta, cujo comprimento e inclinação são, respectivamente, iguais à sua própria dimensão e declividade, na qual o fluxo de detritos irá se desenvolver.

Chama-se a atenção que alguns aspectos específicos, como o número de termos presentes no somatório que compõe a equações de velocidade e de espaço percorrido, assim como o parâmetro usado para reduzir a poropressão inicial, são indicados pelo usuário como parte do input.

Uma vez montada a função que define o espaço percorrido pelo fluxo, é chegado o momento de igualá-la à dimensão do pixel da grade que representa a área de estudo para, então, obter-se a raiz desta equação. Esta fornecerá o tempo gasto no movimento do fluxo, no sentido de transcorrer a distância correspondente à encosta.

Em um momento inicial, procurou-se utilizar as ferramentas para o cálculo de raízes de equações já devidamente incorporadas ao software Matlab, sob um enfoque de variáveis simbólicas. Contudo, os resultados obtidos nesta abordagem não foram aceitáveis, tanto do ponto de vista da quantificação numérica das raízes, quanto de desempenho do código. Valores negativos de velocidade e um longo tempo gasto no processamento fizeram com que os esforços computacionais fossem alterados para uma alternativa usando cálculo numérico na estimativa das raízes.

Dentro do cálculo numérico para aproximação de zeros reais de uma dada equação, distinguem-se duas etapas bem delineadas: uma primeira fase, onde se processa a localização das raízes, conseguindo-se definir um intervalo que contém a raiz e, apoiando-se nesta estimativa inicial, parte-se para a etapa posterior de refinamento sucessivo da raiz, pelo emprego de um método iterativo até se chegar a uma aproximação contida nas margens de uma precisão prefixada.

A primeira parte do processo é pautada no teorema de Bolzano ou do anulamento. Segundo este teorema, dada uma função contínua $f(x)$ no intervalo $[a,b]$, se $f(a)$ e $f(b)$ possuírem sinais contrários, então existe pelo menos um ponto neste intervalo $[a,b]$ que é zero de $f(x)$. De forma habitual, o teorema anterior é conjugado com uma análise gráfica da função para que a restrição imposta ao intervalo inicial possa ser efetivada visualmente. Como se

necessita que o processo de delimitação do intervalo que contém a raiz seja automático e contínuo, buscou-se um artifício apropriado para realizar o isolamento da raiz.

Partindo de limites preestabelecidos para o intervalo inicial $[a,b]$, verifica-se o sinal gerado pelo produto das respectivas imagens $x(a)$ e $x(b)$, definidas pelos limites. Caso o resultado seja negativo, não há o que fazer e pode-se prosseguir para a fase iterativa de refinamento da raiz, pois, conforme o teorema do anulamento, neste primeiro intervalo existe um zero real.

Caso o produto seja positivo, devem-se averiguar alguns fatores e características do comportamento da função em questão, efetuando-se modificações na amplitude do intervalo considerado para garantir a presença de uma raiz no seu interior. Nestas condições, se a derivada da função for negativa no intervalo considerado e $x(a)$ for positivo, isso representa que o zero da função corresponde a uma abscissa superior ao extremo b do intervalo inicial. O programa, desta forma, se encarrega de ampliar o intervalo, deslocando o extremo superior para um valor maior que o atual. Esta metodologia é repetida até que a condição $x(a).x(b) < 0$ esteja satisfeita.

Condição similar tem-se quando $x(a)$ e $x'(t)$ são ambos positivos, com $x(a).x(b) > 0$. O que se deve fazer, portanto, é reduzir o limite inferior do intervalo sucessivamente, para se alcançar $x(a) < 0$ e, conseqüentemente, $x(a).x(b) < 0$. A Fig. 3.31 visa explicar o procedimento.

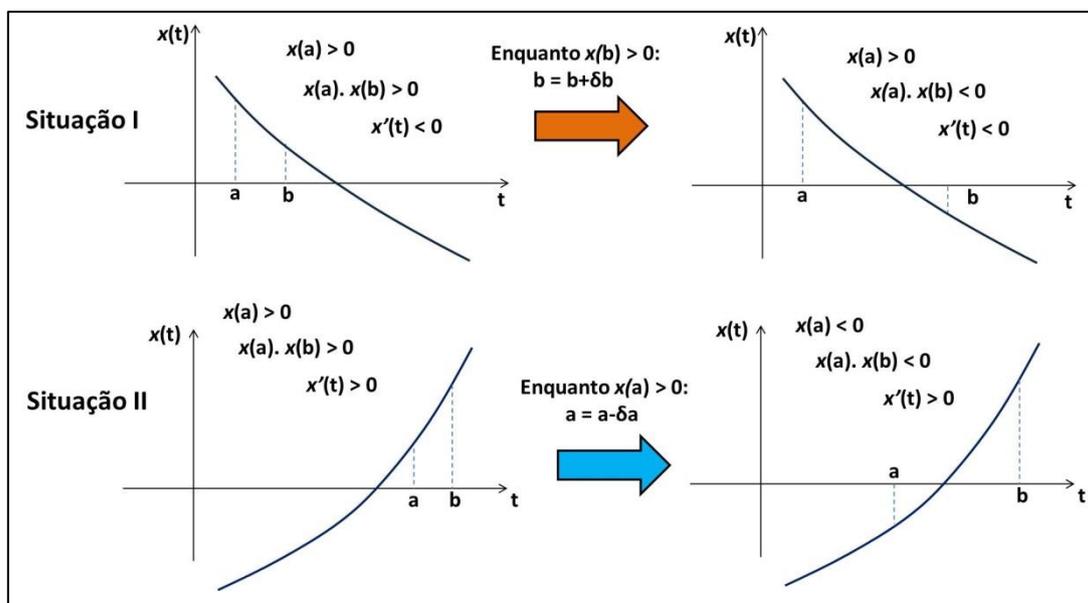


Figura 3.31. Delimitação do intervalo com a raiz da função espaço percorrido $x(t)$, proposta pelo algoritmo para as situações em que $x(a) > 0$ e $x'(t) < 0$ (I) e $x(a) > 0$ e $x'(t) > 0$ (II).

Dois outros casos também podem ocorrer quando o produto $x(a).x(b) > 0$, das imagens dos pontos extremos do intervalo, é positivo:

- $x(a) < 0$ e $x'(t) < 0$. Aplicam-se incrementos negativos ao extremo inferior e testa-se a condição prevista no teorema de Bolzano. Assim que esta for verificada, o novo intervalo estará estabelecido. O limite superior não é alterado;
- $x(a) < 0$ e $x'(t) > 0$. Para esta conjuntura, mantém-se constante o elemento a , de maneira que incrementos positivos são impostos ao limite superior, com o teste de que o produto entre $x(a)$ e o novo $x(b)$ seja negativo.

Com o intervalo $[a,b]$ estipulado a priori, define-se o limite inferior igual a três segundos e o maior valor b como dez segundos. Para todas as células que têm sua velocidade calculada pelo programa, a demarcação do intervalo que servirá de entrada para o refinamento do tempo percorrido pelo fluxo partirá desta primeira atribuição. Os incrementos de modificação das extremidades, tanto positivos quanto negativos, estão fixados em 5% do valor atual. Na Fig. 3.32, estão representadas as duas últimas situações mencionadas de variação do intervalo que contém o zero de $x(t)$.

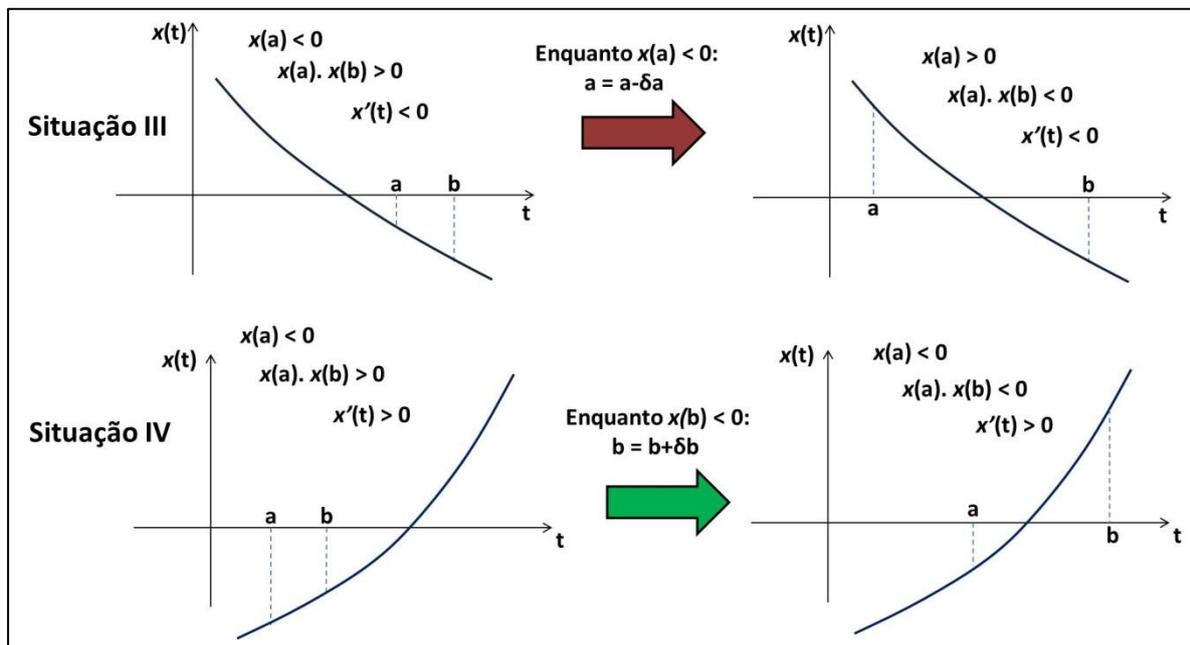


Figura 3.32. Forma para a definição do intervalo com a raiz da função $x(t)$ proposta pelo algoritmo para as situações em que $x(a) < 0$ e $x'(t) < 0$ (III) e $x(a) < 0$ e $x'(t) > 0$ (IV).

De posse do intervalo que se tem a certeza da presença da raiz da função $x(t)$, parte-se para o melhoramento da aproximação utilizando um dos métodos numéricos iterativos de

refinamento do valor de t . Dentre aqueles disponíveis, optou-se pelo método de Newton-Raphson, o qual proporciona uma convergência de resultados relativamente rápida.

O método se desenvolve sob a seguinte consideração: seja $f(x)$ a função da qual se deseja obter o zero real, considerando x_0 como uma boa aproximação da verdadeira raiz r , de maneira que $r = x_0 + h$. Assim, a medida $h = r - x_0$ representa o quão distante se encontra a estimativa x_0 da realidade. Desde que h seja considerado pequeno, pode-se usar uma aproximação linear, considerando uma reta tangente, para concluir que:

$$0 = f(r) = f(x_0 + h) \approx f(x_0) + h \cdot f'(x_0) \quad (3.8)$$

A menos que $f'(x_0)$ esteja próxima de zero, o que não proporcionaria convergência, tem-se:

$$h \approx -\frac{f(x_0)}{f'(x_0)} \quad (3.9)$$

Pode-se escrever, então:

$$r = x_0 + h \approx x_0 - \frac{f(x_0)}{f'(x_0)} \quad (3.10)$$

A nova aproximação para a raiz, agora refinada, seria, conforme a equação abaixo:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (3.11)$$

Generalizando, se x_n refere-se à estimativa atual, a fórmula de recorrência para a próxima aproximação x_{n+1} pode ser escrito como a seguinte expressão.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (3.12)$$

Aplicando-se o exposto ao problema estudado, a rotina considera como estimativa inicial t_0 da raiz de $x(t)$ o limite inferior do intervalo inicialmente proposto, ou seja, três segundos. Caso haja mudança em um dos extremos, provocada pela adequação proposta anteriormente à condição necessária de isolamento do zero da função, este novo limite alterado do intervalo passa a ser considerado o valor de partida para o refinamento da variável tempo.

O critério de parada aqui adotado para o método se traduz na verificação, em cada ciclo, se o módulo da diferença entre a aproximação atual e a anterior é menor que a medida estipulada para este erro. O erro calculado deverá ser inferior a 0,01 para ser aceito.

Também foi adotada uma restrição adicional baseada no número máximo de iterações concentradas no cálculo da raiz em uma dada célula, prevenindo que o programa entre em um “loop” infinito, derivado da não convergência do método. O número inicial fica previsto em 100 ciclos, mas o usuário terá oportunidade de inserir este critério relacionado à quantidade máxima de iterações.

O tempo definido por meio do cálculo numérico é substituído na equação da velocidade, composta pelo somatório correspondente ao mesmo número de termos empregados na formação da função do espaço percorrido. Esse valor encontrado, originalmente na unidade de m/s, é transformado para mm/s (atendendo às considerações da vulnerabilidade) e armazenado em uma matriz específica para este fim.

Enquanto houver a classificação de células como eventos de escorregamentos em um dado ciclo, tanto a espessura de detritos, quanto a velocidade serão determinadas. O procedimento se encerra quando o esquema de verificação para traçar o curso dos deslizamentos, por intermédio dos autômatos celulares, não aponta mais evolução prevista do movimento.

Parte-se, então, para a exposição dos recursos criados no intuito de tornar a experiência do usuário, quanto ao manuseio da ferramenta, na mais produtiva possível, com a consequente explicação de detalhes incorporados para promover isso.

A interface gráfica desenvolvida para utilização da rotina computacional contempla uma série de opções relacionadas às condições iniciais requeridas pelas análises. A Fig. 3.33 traz a GUI responsável pelas interações associadas à simulação das características de trajetória, velocidade e altura de deposição de detritos para os escorregamentos.

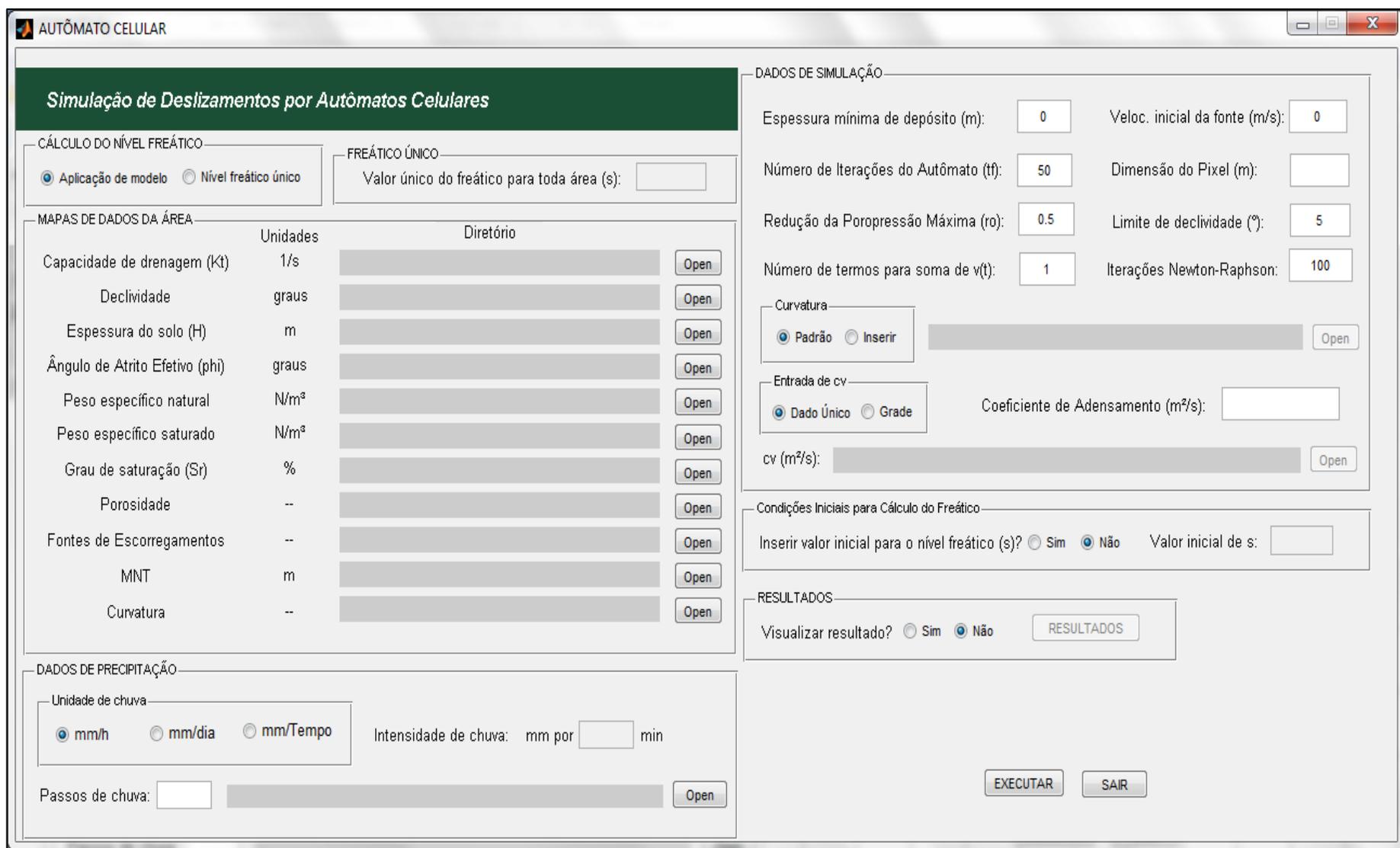


Figura 3.33. Interface gráfica para simulação de deslizamentos segundo autômatos celulares.

O artifício de cálculo para a estimativa da velocidade requer que o plano de informação representativo do nível freático, ao início do processo de ocorrência dos deslizamentos, seja incorporado e isto pode ser procedido de duas formas. É possível optar pela execução do mesmo modelo descrito por Liao et al. (2010), que considera o balanço hídrico no talude, simultaneamente, pelo ganho da infiltração decorrente das chuvas e pela perda ocasionada da percolação, para obtenção da flutuação da camada saturada.

Nesse caso, o usuário terá que entrar com informações de índices físicos e de permeabilidade do solo, além dos dados de chuva, que serão impostos ao terreno e responsáveis pela variação do nível freático. A entrada de dados para a precipitação segue os moldes já apresentados em alguns modelos de ameaça, definindo-se a unidade de intensidade, o número de passos que serão impostos à região e o diretório que contém os arquivos.

Também haverá a oportunidade de fornecer um valor inicial para o nível freático, a partir do qual os passos de chuva irão impor sua influência ou simplesmente considerar que o solo não esteja saturado, ou seja, a espessura da camada saturada em relação à espessura total do solo é inicialmente nula.

A outra opção de definição da espessura saturada disponível como entrada de dados, diz respeito a uma forma mais simplificada, quando não há disponibilidade dos insumos necessários à aplicação do modelo ou quando se pode estimar um valor médio deste parâmetro que represente razoavelmente a condição inicial desta área, sem prejuízos maiores ao resultado da simulação, correspondente à imposição de um único valor global a toda área de estudo. Ao se efetuar a escolha quanto à grade do nível freático, a interface se encarrega de habilitar as componentes que deverão ser preenchidas para a execução do programa, norteando a ação do usuário no sentido de facilitar a interatividade.

Os mapas da área de estudo sob o formato raster, requisitados quando se opta por não utilizar o modelo de Liao et al. (2010) no estabelecimento do nível freático, estão ligados às cartas essenciais ao funcionamento do autômato celular. São elas: as matrizes com o modelo numérico de terreno, a declividade da região, a espessura do solo, ângulo de atrito do solo, curvatura do terreno e mapa das fontes de escorregamento. Ao se utilizar o modelo, esta lista é acrescida pelos planos de informação de capacidade de drenagem, peso específico natural e saturado do solo, grau de saturação e porosidade. O detalhe para a entrada de dados dos mapas está presente na Fig. 3.34.

MAPAS DE DADOS DA ÁREA		Unidades	Diretório
Capacidade de drenagem (Kt)	1/s		Open
Declividade	graus		Open
Espessura do solo (H)	m		Open
Ângulo de Atrito Efetivo (ϕ)	graus		Open
Peso específico natural	N/m ³		Open
Peso específico saturado	N/m ³		Open
Grau de saturação (Sr)	%		Open
Porosidade	--		Open
Fontes de Escorregamentos	--		Open
MNT	m		Open
Curvatura	--		Open

Figura 3.34. Informações requeridas sob a forma de grades da área de estudo.

Além disso, as informações complementares que dizem respeito a iterações empregadas no autômato, regras que definem a propagação do evento e outros dados específicos da malha avaliada vêm com valores nomeados de forma padrão que podem ser modificados a critério daquele que utiliza o programa computacional. Dentre tais, pode-se indicar:

- Velocidade, em m/s, imposta às áreas consideradas como fontes dos escorregamentos, cujo valor padrão é zero;
- Número de passos que serão impostos ao autômato celular na alteração do seu estado (tf). Padronizado em 50 iterações;
- Fator de redução imposto às estimativas de excesso de poropressões máximas (r_0) na base e no topo da camada saturada. Valor padrão de 0,5;
- Dimensão das células, em metros, que formam as matrizes com dados da área de estudo;
- Espessura mínima (em m) de ganho de solo a ser considerada como ponto de acúmulo de detritos. Pontos que alcançaram deposição inferior ao limite estabelecido não são considerados locais onde ocorreu algum acúmulo de material. Valor padrão desta variável corresponde a 0 m;

- Quantidade de termos empregados no somatório que compõem a equação definidora da posição do elemento principal do deslizamento em função do tempo ($x(t)$) e cuja raiz serve para quantificar a velocidade da ocorrência. É proposto um termo;
- Definição do coeficiente de adensamento do solo, em m^2/s . Pode-se contar com um dado único para toda região ou inserir uma grade bidimensional que contemple melhor a variação espacial intrínseca à natureza deste parâmetro, como no caso, por exemplo, de diferentes tipos de solo. Neste caso, deve-se indicar o respectivo arquivo;
- Número de repetições máximas (proposto em 100) usado no método de Newton-Raphson, na tentativa de garantir a convergência do processo iterativo de estimativa da raiz da equação de espaço percorrido (tempo).

O critério de declividade, que discrimina áreas atingidas por escorregamentos de regiões com acúmulo de material proveniente da corrida de detritos, foi fixada em 5° , em decorrência daquilo estabelecido no trabalho de Lai & Dragičević (2011), podendo ser alterado mediante evidências que comprovem uma estimativa mais efetiva.

De maneira análoga, podem-se incorporar as curvaturas mais propensas a abrigarem ocorrências de deslizamentos, por meio de um vetor contendo a sua numeração ou mantendo a curvatura côncava como sendo decisiva no processo. Os dados das células contidos na grade de curvatura devem seguir uma numeração pré-estabelecida que contemple as características de plano e de perfil de curvatura, conforme exposto na Tab. 3.6.

Tabela 3.6. Numeração e respectivas características de curvatura que devem estar presentes na grade correspondente.

Numeração	Características	
	Plano de curvatura	Perfil de curvatura
1	Convergente	Côncava
2	Convergente	Retilínea
3	Convergente	Convexa
4	Planar	Côncava
5	Planar	Retilínea
6	Planar	Convexa
7	Divergente	Côncava
8	Divergente	Retilínea
9	Divergente	Convexa

Na Fig. 3.35 destacam-se os valores padrões propostos para as variáveis ligadas a aspectos específicos da simulação.

DADOS DE SIMULAÇÃO

Espessura mínima de depósito (m):	<input type="text" value="0"/>	Veloc. inicial da fonte (m/s):	<input type="text" value="0"/>
Número de iterações do Autômato (tf):	<input type="text" value="50"/>	Dimensão do Pixel (m):	<input type="text"/>
Redução da Poropressão Máxima (ro):	<input type="text" value="0.5"/>	Limite de declividade (°):	<input type="text" value="5"/>
Número de termos para soma de v(t):	<input type="text" value="1"/>	Iterações Newton-Raphson:	<input type="text" value="100"/>

Curvatura

Padrão Inserir

Entrada de cv

Dado Único Grade

Coefficiente de Adensamento (m²/s):

cv (m²/s):

Figura 3.35. Definição padrão das variáveis do modelo relacionadas à malha, restrições e condições iniciais do autômato celular.

Assim, a opção de curvatura definida como padrão considera aquelas correspondentes aos números 1, 4 e 7, conforme descrito na Tab. 3.6, relativo ao perfil côncavo. Para outras situações, devem-se inserir as curvaturas mais propensas à ocorrência dos deslizamentos por meio da inclusão do vetor, com a representação prevista acima.

Cabe ressaltar que não foi incluído, aqui, nenhum mecanismo prévio de definição ou análise de quais seriam as características geomorfológicas mais ou menos propensas a desencadear o processo de propagação dos deslizamentos, como a regressão logística, por exemplo, utilizada no trabalho de Lai & Dragičević (2011). Desta forma, o usuário deverá executar uma avaliação desta natureza ou buscar subsídios junto a fontes técnicas que permitam esboçar quais aspectos, dentro da configuração que envolve a curvatura e a declividade, possibilitem melhor explicar o fenômeno em questão.

Uma parcela das simulações provenientes das análises pode ser visualizada, propiciando uma noção do panorama criado pelo modelo. Para isso, deve-se selecionar, como positiva, a opção de visualização dos resultados antes de proceder à execução do programa. Ao fim do processo de aplicação do modelo, um botão será habilitado, o qual conduzirá para

uma nova interface onde estarão disponíveis algumas configurações da malha em diferentes instantes.

A interface traz, dentro das alternativas disponíveis, a possibilidade de retratar uma única iteração ou fornecer a propagação dos eventos sob a forma de uma animação. Para auxiliar o usuário na escolha dos dados, no caso da visualização de somente um momento, o número de termos passíveis de serem escolhidos vem expresso no campo “Intervalo disponível”.

Ao inserir um valor válido no campo de seleção do passo e pressionando-se o botão correspondente, a matriz que representa aquele momento dentro da evolução do autômato celular é mostrada na parte esquerda da interface. Também, o número da iteração referente ao valor do dado escolhido, dentre o total proposto para a aplicação no programa, é automaticamente inserido na tela em um campo específico.

Na Fig. 3.36, os componentes desta interface são apresentados. Nesta mesma tela, encontra-se expressa a disposição dos escorregamentos para o décimo sétimo elemento que foi salvo no procedimento, o qual representa o estado 48 da evolução do autômato celular.

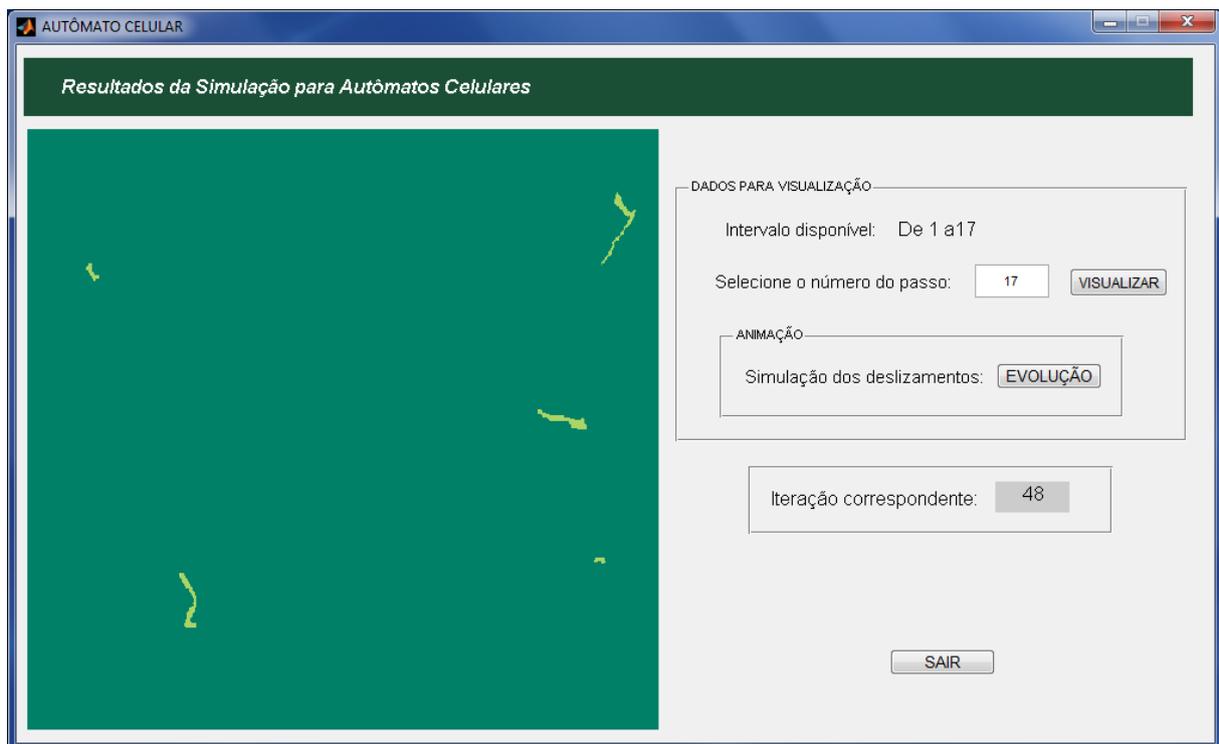


Figura 3.36. Interface para visualização dos resultados para as simulações dos autômatos celulares.

No caso de recorrer à apresentação da animação, à medida que as matrizes são expostas de maneira dinâmica, os respectivos valores das iterações também se alteram, acompanhando a mudança visual.

A implementação do algoritmo para exibição dos resultados foi concebida no sentido de salvar apenas uma parte de todos os estados pelos quais o autômato celular estará sujeito. A quantidade salva para exposição refere-se a um terço do valor de entrada para o número de iterações impostas ao autômato celular mais um. Para isso, procede-se a pré-alocação de uma matriz tridimensional, cujas duas primeiras dimensões são idênticas àquelas dos mapas que representam a região em estudo, enquanto a terceira retrata exatamente o número de passos a serem registrados. Ao se pré-alocar a estrutura de dados, garante-se um melhor desempenho do algoritmo com ganho de velocidade no processamento.

A primeira configuração salva será a composta pelas fontes de escorregamentos e, a partir disto, armazena-se uma matriz a cada três rodadas completadas pela rotina. Desta forma, são salvas as grades para as repetições, além das fontes, de números 3, 6, 9, 12 e assim por diante, até completar o valor calculado da terça parte das repetições.

3.1.4 CUSTOS DAS PROPRIEDADES E DEFINIÇÃO DO RISCO

As duas operações faltantes para a concretização do processo e que se referem à inserção do layer contendo o custo das propriedades e à execução do cômputo da matriz de risco para a área completa, foram aglutinadas em uma única interface gráfica, separadas em painéis distintos. A Fig. 3.37 traz a visão geral da GUI em questão.

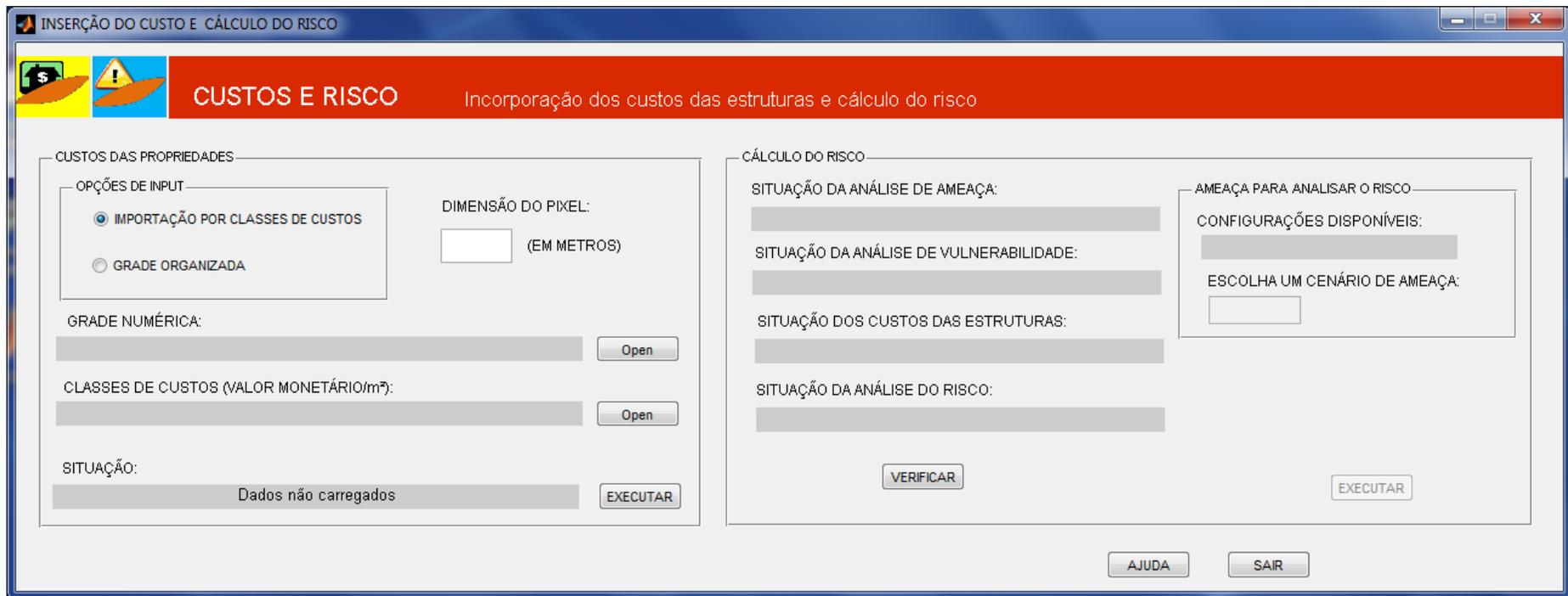


Figura 3.37. Interface gráfica responsável pela inclusão dos custos das estruturas e pela determinação da grade de risco.

Trata-se, agora, de expor as funcionalidades associadas aos componentes de cada uma das opções que se encontram disponíveis ao uso.

3.1.4.1 INCORPORAÇÃO DOS VALORES ECONÔMICOS PARA ESTRUTURAS

O módulo de inclusão das consequências ocasionadas pelas rupturas dos elementos em risco, as quais se manifestam sob a forma de custos monetários, não se fundamentou em um modelo de previsão para estimar suas importâncias, destinando-se, primordialmente, à incorporação, no sistema, dos valores que representam as implicações causadas pelos episódios de movimentos de massa, devidamente organizados.

É necessária uma averiguação sistemática na literatura técnica, bem como em fontes idôneas ligadas a órgãos competentes, responsáveis por avaliar imóveis e estruturas, mensurando, assim, sua cotação, na tentativa de obter as informações pertinentes acerca dos elementos estáticos que se encontram em uma situação de risco.

Para esta etapa, pode-se adentrar com os inputs concernentes aos custos de duas formas principais. A primeira se refere à representação das construções por meio de diferentes classes que se relacionam a uma determinada característica destas estruturas, através da qual se possa estabelecer um valor para cada categoria presente, como, por exemplo, altura da edificação, área construída, entre outras. Neste caso, dois arquivos devem ser carregados no sistema:

- Grade numérica representando a área de estudo, com valores não nulos nas posições válidas das estruturas correspondentes às classes de custos. A presença de um valor nulo encontrado em um dado pixel representa a ausência de estruturas naquele local. A numeração das classes, utilizada para diferenciar os níveis de valores monetários que podem ser utilizados nas análises, será sequencial, a partir de 1 até o número máximo de categorias;
- Vetor coluna trazendo, em cada linha, o registro do valor monetário do metro quadrado de área construída para cada categoria proposta. Assim, na primeira posição do vetor estará o preço relativo à classe de número 1; na segunda posição, o custo ligado à classe 2, e assim por diante, conforme organização de informações já explicada e empregada em situações anteriores. A moeda em que os dados deverão ser apresentados irá depender da escolha e conveniência do usuário, de forma que melhor se adapte a sua realidade e aos propósitos da análise. Ressalta-se que a configuração final do risco será definida em termos da unidade monetária proposta nesta etapa.

De forma complementar, deve-se ingressar, também, com a resolução espacial do pixel que compõe a matriz da área analisada. O algoritmo age de maneira similar ao que foi previsto em momentos anteriores, como no caso da vulnerabilidade: a matriz da região em estudo é varrida integralmente, posição após posição, à procura de uma célula válida que armazena dados não nulos. Ao encontrar pixels nestas condições, o código verifica a numeração ali presente e faz a substituição pelo valor monetário que se encontra guardado na correspondente linha do vetor de classes.

Ao final do processo, esta grade trará, em cada local onde há uma construção, o custo ocasionado por um evento de ameaça por metro quadrado, em conformidade com as categorias previamente estabelecidas. Ao se multiplicar esta matriz pela área do pixel utilizado na análise, obtém-se, por fim, a representação dos gastos totais que cada estrutura estará sujeita, caso seja atingida por um evento de deslizamento.

A razão para que o input fosse assim realizado, fica por conta do fato que a estruturação básica das informações em formato raster nas plataformas de informação geográfica, quando se trata de modelos de dados temáticos, ocorra pelo estabelecimento de inúmeras classes representadas por uma numeração sequenciada. Desta forma, o sistema foi concebido na tentativa de se adequar a este arcabouço, facilitando a manipulação e organização dos dados de entrada.

O segundo modo de incorporação dos custos se dá pela importação direta da matriz que representa os valores econômicos, por metro quadrado, das obras localizadas na região avaliada. Neste caso, também há a necessidade de indicar o tamanho da célula, para que sua área multiplique todos os elementos da grade de custo. A seleção desta opção na GUI faz com que o botão responsável pela escolha do arquivo com o vetor portando as classes de custo torne-se automaticamente desabilitado, indicando que este não é um input requerido para esta alternativa.

O painel ainda indica a situação em que se encontra o carregamento das informações, fornecendo o progresso que este processo alcançou em determinado instante. Com a execução dos módulos até aqui indicados, ficam, então, completamente determinadas as parcelas necessárias para o cálculo do risco.

3.1.4.2 PROCEDIMENTO PARA CÁLCULO DO RISCO

Pressupondo-se que as representações matriciais para a ameaça, a vulnerabilidade e as consequências da ruptura estejam devidamente carregadas e determinadas pelo emprego do

software, é possível efetuar a operação que fornece o resultado para a quantificação do risco do deslizamento.

A área com os componentes gráficos responsáveis pela concretização da obtenção do risco, que se encontra ao lado do painel dos custos, foi criada no sentido de indicar se todas as partes necessárias para a análise de risco estão efetivamente concluídas e aptas a figurarem no procedimento de integração proposto entre as grades de informações. Para tanto, os itens gráficos possuem a função de fornecer o status relativo à concretização das avaliações de ameaça, vulnerabilidade e custos por meio de mensagens em campos específicos. A Fig. 3.38 representa o campo para o cômputo do risco em detalhes.

CÁLCULO DO RISCO

SITUAÇÃO DA ANÁLISE DE AMEAÇA:

SITUAÇÃO DA ANÁLISE DE VULNERABILIDADE:

SITUAÇÃO DOS CUSTOS DAS ESTRUTURAS:

SITUAÇÃO DA ANÁLISE DO RISCO:

AMEAÇA PARA ANALISAR O RISCO

CONFIGURAÇÕES DISPONÍVEIS:

ESCOLHA UM CENÁRIO DE AMEAÇA:

VERIFICAR

EXECUTAR

Figura 3.38. Elementos gráficos voltados para a análise de risco.

Ao pressionar o botão de verificação, mensagens a respeito da execução, ou não, das análises constituintes, serão exibidas. Se todas estiverem concretizadas, serão habilitados tanto o campo de escolha de qual configuração de ameaça será imposta a análise dentre aquelas disponíveis (assinaladas no campo correspondente), quanto o botão voltado para a execução da ação do risco. O controle sobre a execução ou não de uma dada etapa é realizado por meio de variáveis booleanas específicas, que indicam se as fases anteriores já foram empreendidas com sucesso.

A realização do processo de cálculo baseia-se no produto das três parcelas mencionadas acima, para cada uma das posições pertencentes à matriz da extensão avaliada, originando a grade numérica atinente ao risco.

3.1.5 VISUALIZAÇÃO E EXPORTAÇÃO DE RESULTADOS

Ao ser concretizada alguma das análises propostas pelos modelos disponíveis, tem-se a oportunidade de exibir os resultados alcançados em uma área específica da interface gráfica responsável pelas ações de visualização e exportação de dados. A interface gráfica está presente na Fig. 3.39.

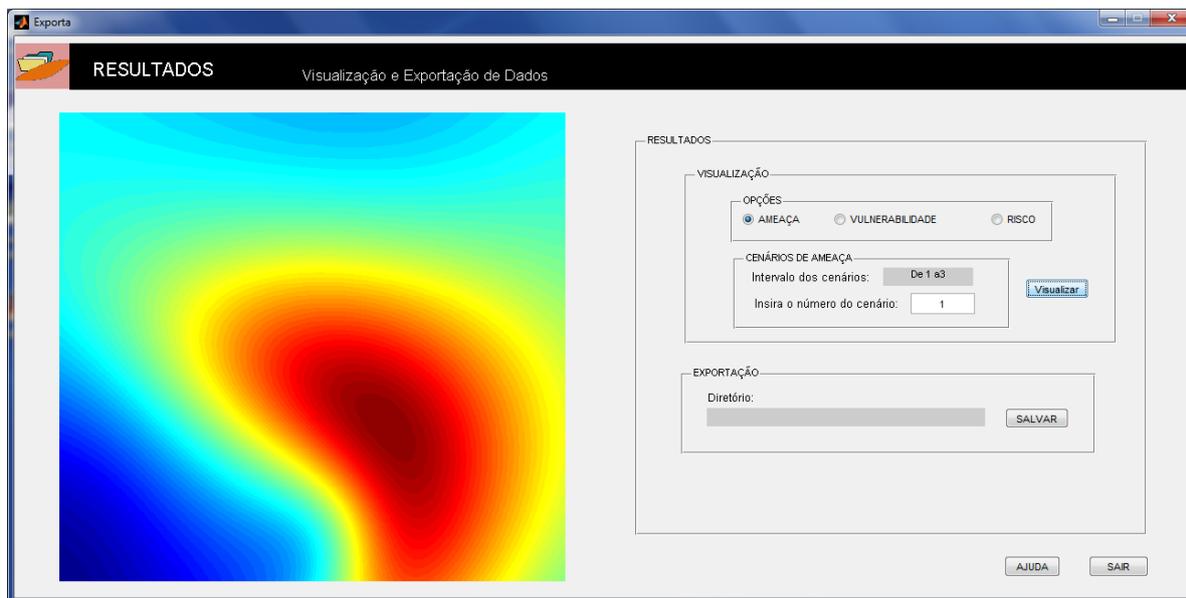


Figura 3.39. Interface gráfica correspondente à exportação e visualização de resultados obtidos no sistema.

O usuário poderá escolher entre os três principais tópicos para esta operação: ameaça, vulnerabilidade ou risco, bastando observar as referidas grades e acionar o botão de visualização. No caso da ameaça, assim como ocorreu na definição do risco, deve-se selecionar uma configuração válida dentre aquelas pertencentes ao intervalo disponível.

Caso alguma das opções não tenha, ainda, sido efetivada, a visualização e exportação, logicamente, não se encontram disponíveis e, conseqüentemente, não serão habilitadas, sendo somente liberadas assim que dados válidos estejam presentes. Na fase de exportação, ao selecionar o botão responsável por salvar uma dada informação, uma tela voltada para a escolha do diretório será aberta, possibilitando que se introduza o nome do arquivo escolhido, permitindo ao usuário gerenciar o armazenamento das suas próprias informações.

3.1.5.1 UTILIZAÇÃO DOS DADOS EXPORTADOS EM AMBIENTE SIG

Os dados exportados representam tão somente a matriz resultante da aplicação de um determinado modelo presente no sistema citado, que se encontra salva em arquivos

específicos. Caso deseje-se incorporá-los a um sistema de informação geográfica, deve-se incluir o cabeçalho proveniente do SIG, como no caso do software Spring (Câmara et al., 1996), com as informações básicas a respeito do projeto em questão (relacionadas à projeção cartográfica, datum, tamanho de células, entre outros), com o intuito de importá-lo para esta plataforma.

Estando devidamente inserida no ambiente SIG, sob a configuração de um plano de informação, a malha com os dados numéricos poderá ser, então, categorizada em variadas classes por meio da ferramenta de fatiamento e classificação, o que proporcionará a criação de uma carta de zoneamento de risco decorrente dos movimentos de massa. Este zoneamento poderá ser efetivado, tomando-se como parâmetro, para a divisão das classes, a divisão hierárquica dos níveis de risco que são propostos para a localidade em estudo.

Faz-se aqui uma consideração em relação à definição do tamanho adequado das células de trabalho, vinculadas ao SIG, a serem utilizadas nas operações de análise de risco devido a deslizamentos. Cada pixel que será analisado refere-se ao domínio correspondente ao território com base no qual se define, calibra e valida um modelo. Conforme Álvarez et al. (2015), a definição do tamanho apropriado da célula depende da escala utilizada, sendo que a sua área mínima, para propósito de operações no SIG, em uma dada escala, relaciona-se com a resolução espacial da cartografia e dos mapas. Calvello et al. (2013) afirmam que a área mínima do pixel está ligada com a resolução da informação da zoneamento que se deseja realizar, explicitando as dimensões adequadas conforme a Tab. 3.7.

Tabela 3.7. Dimensões das células de cálculo (Calvello et al., 2013)

Escala de referência	Dimensões do pixel básico	
	Comprimento (m)	Área (m ²)
1:X	X/1000	X ² /10 ⁶
1:2000	2	4
1:5000	5	25

Por fim, estabelece-se um resumo, sob o formato tabular, no sentido de destacar as particularidades e principais características de cada modelo aqui exposto. Propõe-se que esta tabela possa facilitar a distinção entre os métodos disponíveis, auxiliando na identificação das situações que possam ter sua aplicação imposta de forma mais satisfatória. Tal informação encontra-se disponível na Tab. 3.8.

Tabela 3.8. Síntese das principais características dos modelos utilizados na pesquisa.

Modelo	Finalidade/Abordagem	Escala recomendada de uso	Informação básica requerida	Fundamento Teórico	Característica de cálculo
Curvas de PR	Ameaça /Baseada em Inventários Históricos	Média 1:50.000 a 1:25.000	-Dados de chuva; -Dados e distribuição espacial das curvas de PR;	Baseia-se em Azevedo (2011)	Obtém as PR a partir da aplicação das medidas de chuva em cada curva/função correspondente;
SHIA Landslide	Ameaça/Probabilística	Grande e Média 1:10.000 a 1:50.000	-Dados de chuva; -Declividade; -Distribuição espacial de dados geotécnicos e hidrológicos; -Fatores de correção e condições iniciais hidrológicos; -Dados de coeficientes de variação;	Trabalho de Aristizábal (2014)	Cálculo da PR pela acoplagem de modelo hidrológico baseado em tanques e análise geotécnica de talude infinito, modificada por métodos probabilísticos;

Modelo	Finalidade/Abordagem	Escala recomendada de uso	Informação básica requerida	Fundamento Teórico	Característica de cálculo
Inicialização de Deslizamentos/Fluxo de Detritos	Ameaça/Determinística e Probabilística	Média 1:50.000 a 1:25.000	-Dados de chuva; -Variáveis topográficas; - Distribuição espacial de distritos homogêneos; -Variáveis geotécnicas de cada distrito (estáticas); -Intervalo de duração, taxa de infiltração da chuva e precipitação antecedente (variáveis dinâmicas); - Dados de coeficientes de variação;	Baseia-se em Papa et al. (2011)	Análise da estabilidade pelo modelo de taludes infinitos em conjunto com a avaliação do aumento da poropressão decorrente da chuva pela Equação de Richards. A PR pode ser calculada pela forma originalmente proposta por Papa et al. (2011) ou pelo método probabilístico;

Modelo	Finalidade/Abordagem	Escala recomendada	Informação básica requerida	Fundamento Teórico	Característica de cálculo
SLIDE	Ameaça/Probabilística	Grande e Média 1:5.000 a 1:50.000	-Dados de chuva; -Declividade; - Distribuição espacial de dados geotécnicos; - Coeficientes de variação das variáveis randômicas; -Condição inicial do nível freático;	Trabalho de Liao et al. (2010)	Uso da análise probabilística em conjunto com modelo que considera a infiltração das chuvas e a saturação parcial do solo na resistência ao cisalhamento;
Vulnerabilidade baseada em cenários de ameaça	Vulnerabilidade/Quantitativa	Acompanha aquela adotada no mapeamento de ameaça	-Características da ameaça; -Características das estruturas;	Referente à Li et al. (2010)	A quantificação da vulnerabilidade é dada em função da intensidade da ameaça e da capacidade de resistência das estruturas envolvidas;

Modelo	Finalidade/Abordagem	Escala recomendada	Informação básica requerida	Fundamento Teórico	Característica de cálculo
Geração de trajetórias, velocidade e áreas de acúmulo para deslizamentos	Vulnerabilidade (Ameaça)/Determinística via autômatos celulares	Grande e Média 1:5.000 a 1:50.000	<ul style="list-style-type: none"> - Ameaça (fonte dos deslizamentos); - Dados geotécnicos; -MNT, declividade e curvatura; -Dados de calibração; 	Trabalhos de Lai & Dragičević (2011), Qiao & Clayton (2013) e Gruber & Peckham (2009)	Aplicação dos autômatos celulares para definição da trajetória dos deslizamentos a partir de células consideradas fontes. Para cada célula da trajetória, estimativa da velocidade e espessura de detritos com base em características geotécnicas e geomorfológicas do terreno

Modelo	Finalidade/Abordagem	Escala recomendada	Informação básica requerida	Fundamento Teórico	Característica de cálculo
Cálculo do risco dos deslizamentos	Risco/Determinístico	Segue a escala estabelecida nas demais etapas	- Ameaça; - Vulnerabilidade; - Custo de ruptura;	Baseado em Fell et al. (2005)	Integração dos insumos básicos pelo produto das três grades numéricas

4 RESULTADOS E ANÁLISES

O presente capítulo tem por finalidade, apresentar uma série de aplicações práticas empreendidas com o auxílio do sistema criado, como forma de mostrar a capacidade de cálculo dos módulos disponíveis no programa, do ponto de vista da sua funcionalidade, diante de distintos tipos de configurações ambientais. Um segundo objetivo corresponde à realização das avaliações e geração de conclusões sobre os casos estudados, com base na interpretação dos dados obtidos, permitindo uma melhor compreensão dos resultados alcançados.

Estes exemplos tiveram seu processo de elaboração paralelo à implementação do programa computacional, sendo concretizados à medida que as rotinas responsáveis pelos modelos eram finalizadas ou mesmo nas primeiras versões do programa, onde as funcionalidades de input não se mostravam tão eficientes. Também são citados os estudos iniciais relacionados às análises preliminares de susceptibilidade, que serviram de base para a construção de todo o arcabouço voltado para a avaliação do risco.

Pelo fato da criação das interfaces gráficas, segundo a aparência mostrada no capítulo anterior, terem se configurado como a última etapa a ser estabelecida no escopo do desenvolvimento do trabalho, em alguns dos casos relatados como exemplos, a entrada de dados se efetivou de maneira mais trabalhosa e restrita, pelo uso direto de linhas de código para o carregamento das informações e a organização dos arquivos com nomes específicos e situados em um diretório comum. Conforme visto anteriormente, estas dificuldades foram transpostas na atual versão do sistema, sendo que todos os exemplos realizados nas primeiras versões podem ser concluídos com o auxílio da recente configuração do programa, de forma muito mais intuitiva e facilitada.

Nas análises seguintes, optou-se, como plataforma baseada em sistemas de informação geográfica a ser utilizada, pelo software Spring 5.2, no sentido de estruturar as informações de cunho espacial, assim como no pós-processamento para a visualização das grades geradas. Os exemplos são explicados nas seções a seguir.

4.1 ANÁLISE DA SUSCEPTIBILIDADE A ESCORREGAMENTOS RASOS PELO USO DE MÉTODOS PROBABILÍSTICOS

O exemplo inicial a ser exposto refere-se à determinação da susceptibilidade à ocorrência de escorregamentos sob a representação da probabilidade de ruptura das encostas. Este trabalho foi considerado como o precursor do produto computacional proposto nesta tese, já que foi neste exemplo que se estabeleceram os principais embasamentos ligados à

organização e manipulação de dados ambientais, em conjunto com o tratamento da variabilidade atrelada aos parâmetros do solo.

O foco da abordagem desenvolvida pautou-se em um processo de integração entre as informações com um caráter espacial, oriundas da plataforma baseada em sistemas de informação geográfica e uma ferramenta computacional, na qual estivessem implementados os métodos probabilísticos responsáveis pelo desdobramento das análises. Foram aplicados o método FOSM e o Método das Estimativas Pontuais em uma determinada região do Distrito Federal (DF), tomando-se, para isto, uma função de desempenho que representa um modelo de estabilidade de encostas, no intuito de se obter a probabilidade de ruptura vinculada aos taludes em estudo.

A função de desempenho considerada no processo probabilístico, a qual relaciona a variável dependente, representada pelo fator de segurança, com as variáveis independentes, utiliza, como base, o modelo de talude infinito para solos homogêneos em termos de uma análise de tensões efetivas (Biondi et al., 2000). A equação do fator de segurança é dada pela seguinte fórmula:

$$FS = \frac{c'}{\gamma.H.\sin \beta.\cos \beta} + \frac{\tan \phi'}{\tan \beta} \quad (4.1)$$

onde:

c' representa a coesão efetiva;

ϕ' relaciona-se ao valor do ângulo de atrito efetivo do solo;

γ refere-se ao peso específico do solo;

H é a profundidade da camada de solo na superfície de ruptura potencial;

β representa a inclinação do talude.

Utilizou-se, como base em termos de informações para elaboração desta pesquisa, o trabalho desenvolvido por Silva et al. (2007), o qual consistiu na aplicação do modelo de previsão de escorregamentos rasos SHALSTAB, no intuito de fornecer um estudo sobre a susceptibilidade destes fenômenos na área conhecida como Vila Varjão, a qual configura-se, conforme Silva et al. (2007), como um dos mais antigos assentamentos presentes no DF. O trabalho citado teve um caráter essencialmente determinístico e forneceu, como resultado, um mapa de susceptibilidade a escorregamentos rasos para a região alvo do estudo, medida em termos de fatores de segurança.

Para o presente trabalho, foram criados, inicialmente, um banco de dados e um projeto no software Spring 5.2, voltados para a estruturação das informações espaciais essenciais para

a elaboração das análises. Destacam-se as seguintes informações referentes às principais características deste projeto:

- Projeção e datum utilizados: UTM/ Astro-Chuá, zona 23;
- Retângulo envolvente estabelecido em coordenadas planas:
X1 = 188.000 m; Y1 = 8.259.000 m;
X2 = 196.000 m; Y2 = 8.264.000 m;
- Escalas dos planos de informação: 1/40.000;

Destaca-se, ainda, que os planos de informação empregados neste trabalho são compostos por células com resolução espacial de 20 m. Assim sendo, a região em análise seria representada por planos de informações em formato matricial, compostos por 246 linhas e 394 colunas. A área de estudo definida pelas coordenadas do retângulo envolvente do projeto refere-se à mesma utilizada no artigo Silva et al. (2007).

A etapa seguinte consistiu na definição das informações necessárias para o desenvolvimento do modelo previsto. Procedeu-se a importação das curvas de nível da região avaliada para a plataforma SIG, procedentes da base cartográfica do Sistema Cartográfico do Distrito Federal (SICAD), disponíveis no endereço eletrônico da Secretaria de Estado de Desenvolvimento Urbano e Habitação do DF. Estas isolinhas estão contidas nas ortocartas presentes nas folhas 86, 87, 103 e 104 da articulação SICAD, sendo que as entidades citadas foram importadas sob a forma de amostras. Estas curvas de nível serviram de input para a geração, a partir dos recursos presentes no programa SIG, dos planos de informação referentes ao MNT e à declividade da área. Os valores das declividades foram categorizadas seguindo as mesmas classes temáticas adotadas por Silva et al. (2007).

Os valores médios adotados para os demais parâmetros da função que define o fator de segurança em relação a cada célula de trabalho são os mesmos empregados por Silva et al. (2007) em seu trabalho, os quais foram obtidos a partir de ensaios geotécnicos realizados em amostras indeformadas, procedentes de solos da Vila Varjão e que se encontram listados a seguir:

- Coesão efetiva: 10 kPa;
- Ângulo de atrito efetivo do solo: 20°;
- Peso específico do solo: 16,5 kN/m³;
- Espessura do solo: 0,5 m;

Desta maneira, para cada pixel concernente a uma dada porção representativa do terreno analisado, todas as variáveis necessárias para a obtenção do fator de segurança

estariam definidas. A Fig. 4.1 traz a representação da carta de declividade para a área de estudo.

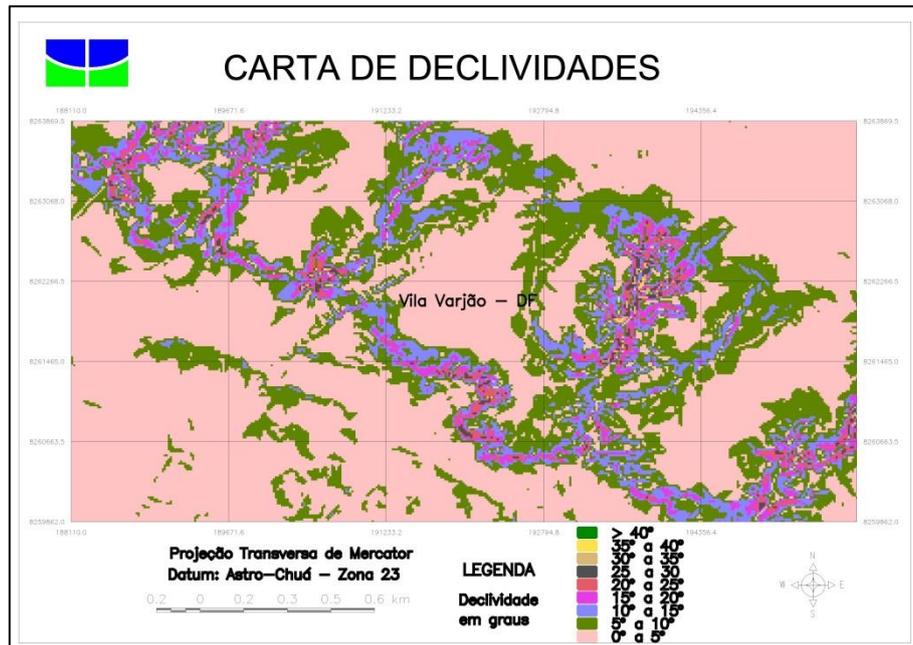


Figura 4.1. Mapa de declividades em graus para a Vila Varjão.

Os instrumentos responsáveis pelo processamento dos cálculos atrelados aos métodos probabilísticos corresponderam a rotinas computacionais desenvolvidas no Matlab. Os dados de entrada foram exatamente os valores dos parâmetros citados anteriormente. Como hipótese, assumiu-se que, tanto as variáveis independentes quanto o fator de segurança, apresentariam distribuições estatísticas e probabilísticas normais, respectivamente.

A coesão efetiva e a tangente do ângulo de atrito efetivo do solo foram as variáveis independentes consideradas com natureza aleatória, em ambos os métodos aplicados, para fins das análises probabilísticas. Os desvios padrões para estas variáveis foram computados por intermédio dos coeficientes de variação destes parâmetros atribuídos a Assis et al. (2012) anteriormente e cujos valores são mostrados na Tab. 4.1.

Tabela 4.1. Parâmetros da análise probabilística.

Parâmetro	Valor médio	Coefficiente de variação	Desvio Padrão
c' (kPa)	10	40%	4
$\tan \phi'$	$\tan 20^\circ$	10%	0,036

A execução do método FOSM correspondeu ao cálculo, para cada célula de trabalho, do fator de segurança médio e da sua variância, conforme exposto na Eq. 2.20 e na Eq. 2.19, respectivamente.

Os resultados para os fatores de segurança médios (μ_{FS}) e das suas respectivas variâncias foram armazenados em matrizes distintas com a mesma resolução espacial já citada, onde cada elemento da matriz representava uma determinada célula da área de estudo. A extração da raiz quadrada da variância forneceu o desvio padrão do fator de segurança (σ_{FS}) em relação a cada pixel. Pode-se afirmar, ainda, que cada um destes pares correspondentes de parâmetros define uma distribuição normal de probabilidade para o fator de segurança, associada à determinada célula da área de estudo. As derivadas da função, que define o fator de segurança, em relação às variáveis probabilísticas, foram obtidas analiticamente e, então, implementadas.

Como se adotou, por hipótese, que o fator de segurança segue uma distribuição de probabilidade normal, fez-se necessário averiguar se este modelo de distribuição representa bem os dados em questão, já que a variável aleatória associada ao fator de segurança admite apenas valores não negativos. Assim sendo, efetuou-se a verificação da seguinte condição para o conjunto de informações obtido anteriormente:

$$\mu_{FS} - 3\sigma_{FS} > 0 \quad (4.2)$$

Para todos os dados das matrizes, observou-se o processamento da regra estabelecida acima, o que permite afirmar que 99,7% dos valores da variável aleatória vinculada à curva da distribuição normal do fator de segurança mostram-se com valores positivos, o que pode ser considerado aceitável.

Na etapa voltada para o cálculo da probabilidade de ruptura, verificou-se, pela primeira vez, a necessidade da transformação, para a sua obtenção, do valor crítico do fator de segurança para a variável aleatória padronizada, a qual foi considerada como limite superior no cálculo da integral para a função de distribuição normal padronizada. A definição da variável Z_{crit} foi realizada usando os valores contidos nas matrizes dos fatores de segurança médios e dos desvios padrões estabelecidos anteriormente, através da Eq. 3.3, anteriormente citada. No presente exemplo, o valor unitário para FS foi considerado como crítico.

Os pares de valores correspondentes a μ_{FS} e σ_{FS} se alteraram, à medida que as matrizes referentes a estes parâmetros foram percorridas. Para cada valor Z_{crit} , a integral da curva normal padronizada para valores menores ou iguais a este ponto foi calculada, representando, assim, a probabilidade de ruptura para uma determinada posição da matriz e, por

consequência, para uma determinada célula da região em análise. O processo resultou em uma matriz contendo os registros das probabilidades de ruptura para toda região estudada. Esta abordagem, estabelecida neste primeiro caso de estudo, mostrou-se de suma importância do ponto de vista de implementação computacional, sendo também estendida para as subsequentes análises de ameaça que haveriam de ser programadas.

Esta grade numérica foi, então, transferida para o software Spring 5.2, onde o mapa de probabilidades de ruptura foi obtido pela geração de uma representação temática com o uso do procedimento de fatiamento em classes. De forma similar àquela empregada na computação da probabilidade de ruptura, mas pela aplicação, agora, da Eq. 2.24, tornou-se possível determinar a matriz de índices de confiabilidade para a área analisada.

Em se tratando do método de Rosenblueth (1975), a função de desempenho foi avaliada para os valores das estimativas pontuais das variáveis com caráter aleatório, calculados na média mais um desvio padrão e média menos um desvio padrão, de forma que os valores assumidos por estes parâmetros encontram-se mostrados na Tab. 4.2.

Tabela 4.2. Valores empregados na execução do método das estimativas pontuais.

Parâmetro	Valor mínimo	Valor máximo
c' (kPa)	6	14
$\tan \phi'$	0,328	0,399

Para cada célula do terreno, os valores para o fator de segurança foram determinados através da combinação das variáveis citadas anteriormente, resultando em quatro casos de análise. Estes casos forneceram elementos que serviram para compor uma amostra da variável dependente. Usando como base esta amostra com quatro elementos, calculou-se sua média e seu desvio padrão, os quais se configuraram como os parâmetros da distribuição estatística do fator de segurança. Assim, ao operar este procedimento para toda a área estudada, novamente gerou-se as matrizes dos fatores de segurança médios e dos desvios padrões com as mesmas dimensões daquelas estabelecidas no método FOSM.

Aqui também foi realizada a verificação da boa adequação da hipótese de adoção de uma distribuição de probabilidade normal para a representação dos dados obtidos, conforme estabelecido anteriormente pela Eq. 4.2, a qual apresentou um resultado satisfatório. O cálculo da probabilidade de ruptura, bem como a elaboração do seu respectivo mapa, obedeceu aos mesmos passos assinalados previamente.

Tratando-se dos resultados alcançados, entre todas as células da área de trabalho, as porcentagens mínimas e máximas de probabilidade de ruptura estabelecidas por ambos os métodos probabilísticos foram 0,01% e 5,07%, respectivamente. O mapa de probabilidade de ruptura foi elaborado conforme a seguinte divisão de classes:

- Células com probabilidade de ruptura inferior a 1%;
- Células com probabilidade de ruptura variando de 1% a 2%;
- Células com probabilidade de ruptura entre 2% e 3%;
- Células com probabilidade de ruptura variando de 3% a 5,07%;

Ao efetuar a tabulação cruzada entre os planos de informações relativos à probabilidade de ruptura e a declividade, observou-se que 99% da área analisada apresentou probabilidade de ruptura inferior a 1%, sendo que destes, 98% possuíam declividades que variam de 0° a 20°. Apenas 1% de toda região situou-se na faixa de 1% a 5,07% de probabilidade de ruptura, distribuindo-se por declividades que variaram de 20° a 40°.

Os mapas relativos à probabilidade de ruptura, conseguidos a partir do método FOSM e do método das estimativas pontuais, encontram-se representados na Fig. 4.2 e na Fig. 4.3, respectivamente.

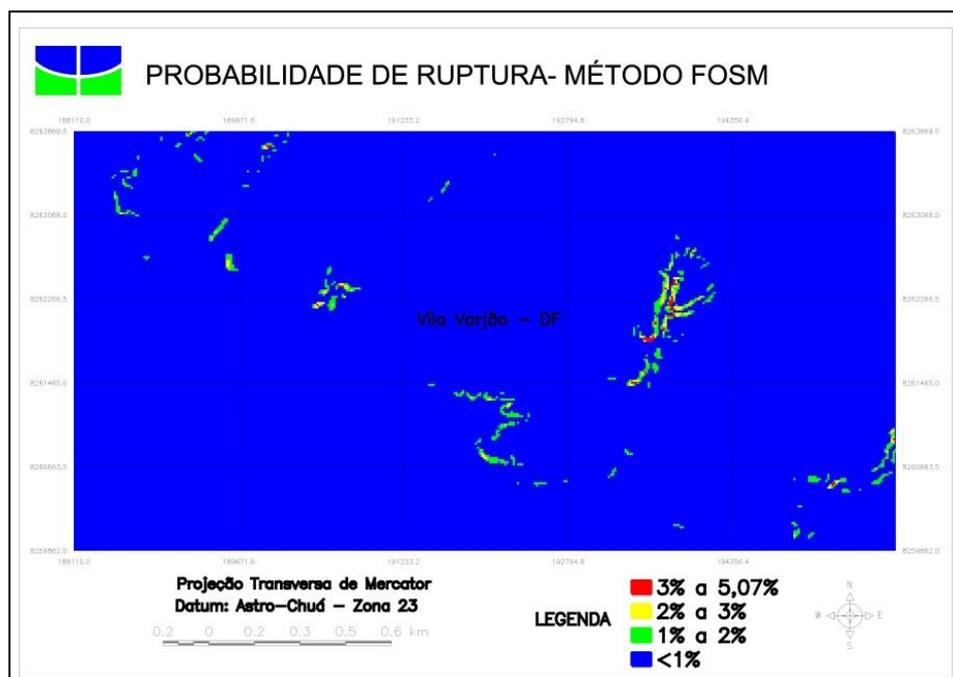


Figura 4.2. Carta de probabilidade de ruptura determinada pelo método FOSM.

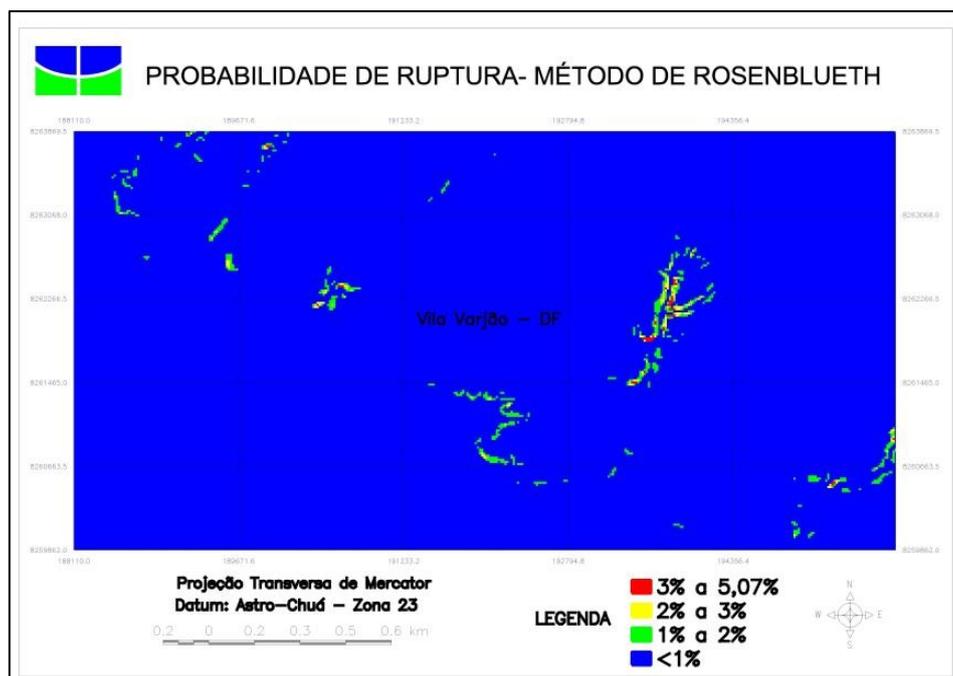


Figura 4.3. Carta de probabilidade de ruptura determinada pelo método das estimativas pontuais.

Observa-se que as análises fornecidas pelos métodos probabilísticos indicaram uma probabilidade de ruptura mais proeminente para inclinações variando de 25° a 40°. Algumas poucas células tiveram divergência de classificação em relação aos dois métodos probabilísticos empregados, quanto às classes de declividade e probabilidade de ruptura, o que praticamente não representou mudança significativa nos percentuais demonstrados acima.

Mediante o exposto, apesar de esse exemplo carregar certa simplicidade quanto ao modelo adotado e às limitações em relação à representatividade das características do meio físico (uso de apenas um tipo de solo), além de não incorporar a ação de nenhum agente deflagrante na determinação dos eventos de escorregamentos, o grande mérito que envolve este estudo reside na definição de diretrizes básicas de tratamento e integração do fluxo de informações entre as plataformas utilizadas. Soma-se a isto, o ganho em termos de técnicas e desempenho de programação, quanto a técnicas mais adequadas, que abriram frente para a concretização dos pilares técnicos nos quais os novos enfoques estariam assentados.

4.2 PROCEDIMENTO PARA AVALIAÇÃO DO RISCO DECORRENTE DE DESLIZAMENTOS APLICADO À REGIÃO DE MEDELLÍN, COLÔMBIA

Um segundo trabalho em destaque, agora no intuito de investigar a potencialidade do sistema computacional desenvolvido em relação a todas as fases previstas, buscou aplicar o aparato a um exemplo completo, voltado para a previsão dos riscos provocados por deslizamentos. A região escolhida para a simulação correspondeu à localidade de La Ilusión, situada no setor rural denominado San Cristóbal, o qual pertence ao município de Medellín, Colômbia. Elegeu-se esta área piloto, na medida em que houve uma facilidade quanto à disponibilidade e captação dos insumos e informações básicas, necessários para a concretização do caso de estudo.

Seguindo as diretrizes estabelecidas pela metodologia, nomearam-se, primeiramente, as alternativas do sistema mais adequadas ao contexto do exemplo, mediante a observação das informações que foram conseguidas e o arranjo por elas apresentadas. Selecionou-se a opção alicerçada no trabalho de Papa et al. (2011), como modelo atuante na realização do estudo voltado para a ameaça de deslizamentos. A operação para atribuição da vulnerabilidade relacionada às estruturas correspondeu à utilização da alternativa na qual estão inseridos os valores dos parâmetros ligados às construções, oriundos da pesquisa de Li et al. (2010), uma vez que não se conseguiu um grau de detalhamento satisfatório, quanto à caracterização das edificações fixadas na região avaliada. Estas escolhas direcionaram os procedimentos que deveriam ser efetivados e os arquivos a serem criados para a aplicação do sistema de risco.

Inicialmente, criou-se um banco de dados e um projeto no software Spring 5.2, específicos para este exemplo, exatamente no intuito de, além de definir a área física de trabalho, preparar e estruturar as informações espaciais pertinentes no caso do exemplo. Destacam-se algumas características do projeto que foram estabelecidas no ato da sua definição:

- Projeção e datum utilizados: UTM/ Bogotá, zona 18;
- Retângulo envolvente que define a área piloto de estudo, determinado em coordenadas planas:

$X1 = 828.983 \text{ m}; Y1 = 1.187.988 \text{ m};$

$X2 = 830.504 \text{ m}; Y2 = 1.189.017 \text{ m};$

A partir disto, importaram-se informações geográficas desta região, contidas em arquivos sob o formato shape, as quais são originadas do projeto SIGMA (Sistema de Información Geográfica para Medellín y sus Alrededores). O conjunto de dados relevantes

para a concretização do trabalho engloba curvas de nível da área analisada, hidrologia caracterizada pelas linhas de drenagem, malha viária e estruturas presentes, as quais foram dispostas ordenadamente em categorias e planos de informações definidos, todos sob o formato inicial de entidades vetoriais.

Dentre as operações procedidas, gerou-se o modelo numérico de terreno com base nas amostras vetoriais determinadas pelas curvas de nível e considerando-se a drenagem como representantes das linhas de quebra de relevo. A partir do modelo numérico de terreno, derivaram-se os planos de informações atrelados à declividade e à grade de fluxo acumulado. A resolução espacial dos planos de informação que comportam os dados matriciais foi estabelecida pelo uso de pixels com dimensões de 5 m, sendo que a matriz gerada para a região inteira é composta por 206 linhas e 304 colunas.

Em relação aos parâmetros de caracterização do solo, foram utilizados os resultados presentes na pesquisa de Lôbo (2012), conseguidos a partir de ensaios próprios de laboratório e de campo realizados por Lôbo (2012), concernentes ao setor de Pajarito, localizado a sudeste da área piloto. Os coeficientes de variação para as variáveis γ_s , ϕ e c foram fornecidos por González (1997) e Quevedo (2002) citados por Rodríguez (2010) e, para a permeabilidade, por Duncan (2000). Estimou-se a espessura média da camada de solo igual em 8 m, com desvio padrão de 4 m. Por hipótese, considerou-se o comportamento isotrópico do solo, de forma a apresentar os mesmos valores de permeabilidades nas direções K_z e K_x .

Na Tab. 4.3 encontram-se os valores médios das variáveis γ_s , ϕ , c e K_z , em conjunto com os coeficientes de variação empregados e os seus respectivos desvios padrões.

Tabela 4.3. Definição dos valores médios, coeficientes de variação e desvios padrões para as características do solo.

Variáveis	Média (μ)	Coefficiente de variação (em %)	Desvio padrão (σ)
γ_s (kN/m ³)	19	5	0,95
ϕ (°)	43	8	3
c (kPa)	24,6	34	8,4
K_z (mm/s)	0,0127	90	0,0114

Ressalta-se que as variáveis citadas acima foram adotadas, obedecendo a uma distribuição normal. Pela ausência de informações provenientes de ensaios para a execução dos ajustes específicos sobre a difusividade hidráulica máxima característica, lançou-se mão

do uso de trabalhos que exemplificassem a quantificação do fator mencionado. Papa et al. (2011) utilizaram o parâmetro de calibração D_0/K_z igual a 0,1 para o seu cálculo, enquanto Iverson (2000) citou um valor de difusividade hidráulica correspondente a $10^{-3} \text{ m}^2/\text{s}$, para um solo com propriedades aproximadas daquele que constitui a presente região de estudo. Definiram-se as difusividades empregadas no exemplo, respaldando-se no exposto, situando-as em ordens de grandezas que variaram de 10^{-4} a 10^{-3} , numericamente similares a K_z .

Determinados os valores médios das variáveis estáticas e suas variabilidades, foi possível criar diferentes combinações para estes fatores que obedecessem à distribuição estatística predita, a fim de representar os distritos homogêneos do ponto de vista das características geotécnicas, mas permitindo, também, congrega a variação inerente ao meio físico.

Para as variáveis Z_T , γ_s , ϕ , c e K_z , geraram-se aleatoriamente, com o auxílio do programa Matlab, elementos que satisfizessem as distribuições normais definidas a partir das médias e desvios padrões citados anteriormente. Uma vez definida a permeabilidade K_z , foi possível obter as variáveis D_0 e K_x por meio das considerações já citadas há pouco.

Com base no procedimento assumido, vinte e cinco combinações distintas das variáveis estáticas foram estabelecidas, representando, desta forma, os distritos definidores da área analisada. A Tab. 4.4 traz os valores destas diferentes combinações que constituíram os dados de entrada para as variáveis estáticas.

Tabela 4.4. Valores dos dados de entrada para as variáveis do solo.

Distritos	Z_T (m)	γ_s (kN/m ³)	ϕ (°)	c (kPa)	K_x (mm/s)	K_z (mm/s)	D_0 (m ² /s)
1	4,55	19,17	45	27,9	0,0166	0,0166	0,0017
2	8,31	18,02	43	30,5	0,0095	0,0095	0,0009
3	3,14	19,90	44	23,7	0,0147	0,0147	0,0015
4	3,55	19,29	41	24,3	0,0190	0,0190	0,0019
5	7,97	19,13	46	25,4	0,0164	0,0164	0,0016
6	14,13	19,49	38	18	0,0053	0,0053	0,0005
7	4,92	19,25	45	36,6	0,0068	0,0068	0,0007
8	9,49	18,11	40	24,7	0,0013	0,0013	0,0001
9	7,10	18,85	42	30,4	0,0237	0,0237	0,0024
10	12,47	18,86	45	17,5	0,0124	0,0124	0,0012
11	3,64	18,49	47	15,6	0,0055	0,0055	0,0005

Distritos	Z_T (m)	γ_s (kN/m ³)	ϕ (°)	c (kPa)	K_x (mm/s)	K_z (mm/s)	D_0 (m ² /s)
12	8,13	20,60	39	23,9	0,0202	0,0202	0,0020
13	10,21	18,17	47	22,5	0,0071	0,0071	0,0007
14	12,40	18,54	45	34,7	0,0176	0,0176	0,0018
15	14,18	18,32	43	29,8	0,0294	0,0294	0,0029
16	8,34	17,88	42	29,2	0,0230	0,0230	0,0023
17	2,03	18,82	42	12,5	0,0109	0,0109	0,0011
18	5,03	18,74	42	16,5	0,0114	0,0114	0,0011
19	3,75	20,45	43	26,4	0,0340	0,0340	0,0034
20	17,40	18,76	43	21,7	0,0108	0,0108	0,0011
21	5,54	17,99	46	35,5	0,0160	0,0160	0,0016
22	10,99	20,52	48	35,9	0,0163	0,0163	0,0016
23	7,23	20,17	45	19,8	0,0229	0,0229	0,0023
24	11,55	18,78	42	32	0,0151	0,0151	0,0015
25	4,94	17,57	45	36,4	0,0167	0,0167	0,0017

A configuração para a distribuição espacial das características pela área de estudo se deu pela geração de uma matriz randômica, cuja numeração, pertinente aos distritos homogêneos, estaria disposta aleatoriamente nos seus elementos.

Assim, cada elemento da matriz teria um número que varia de 1 a 25, ligando-se, por conseguinte, a uma das representações de variáveis do solo propostas para a pesquisa.

O input para as variáveis dinâmicas, associadas ao evento deflagrador, foi determinado por meio da análise da série histórica de precipitação proveniente da rede pluviométrica do projeto SIATA (Sistema de Alerta Temprana de Medellín).

A estação pluviométrica pertencente ao SIATA, a qual se encontra mais próxima da região avaliada, denomina-se estação “Fundación Hogares Claret”. Utilizaram-se informações disponíveis de chuva para o ano de 2011, referentes aos meses de agosto a novembro, as quais se encontram expressas em acumulados que foram registrados a cada cinco minutos.

Compilaram-se esses registros, procedendo-se o acumulado horário e mensal da precipitação para o período mencionado, sendo que isto permitiu a observação da evolução dos eventos chuvosos e orientou a adoção dos dados de entrada. Os acumulados de chuva mensais medidos encontram-se na Tab. 4.5.

Tabela 4.5. Acumulados de chuva mensais para o período considerado.

Mês	Acumulado mensal (mm)
Agosto	164,05
Setembro	208,49
Outubro	349,41
Novembro	302,45

Tomou-se como base para a definição da variável $(I_z)_{\text{estacionário}}$, o maior valor do acumulado pluviométrico mensal estabelecido, que, no caso, refere-se ao mês de outubro. De maneira análoga, investigou-se, dentro dos registros dos acumulados horários, qual a duração dos eventos de precipitação mais intensos. Verificou-se que o evento mais intenso apresentou um pico pluviométrico de 38,8 mm/h, com duração de 20 horas entre o primeiro registro não nulo e o valor máximo mencionado. Pela ausência de um montante maior de informações disponíveis correspondentes às chuvas características, optou-se por majorar o valor da precipitação horária, simulando um evento chuvoso extremo que proporcionasse uma condição adversa à estabilidade. A Tab. 4.6 reproduz o input para as variáveis dinâmicas.

Tabela 4.6. Valores de input para as variáveis dinâmicas.

Variáveis	Valores		
	Mínimo	Máximo	Nº valores
T	0 h	20 h	5
I_z	0 mm/h	60 mm/h	10
$(I_z)_{\text{estacionário}}$	0 mm/mês	350 mm/mês	10

A título de simulação para a chuva atuante como deflagradora, projetou-se a intervenção de um evento chuvoso de três horas, formulado em grades numéricas criadas artificialmente no próprio software Spring 5.2 e que representam a precipitação horária em milímetros. O valor máximo conferido ao episódio pluviométrico foi de 50 mm/h, enquanto que seu valor mínimo foi de 2 mm/h. A diferenciação entre os planos onde se encontram as medidas pluviométricas forneceu exatamente o progresso do evento de chuva submetido à área piloto, conforme explicitado pela Fig. 4.4.

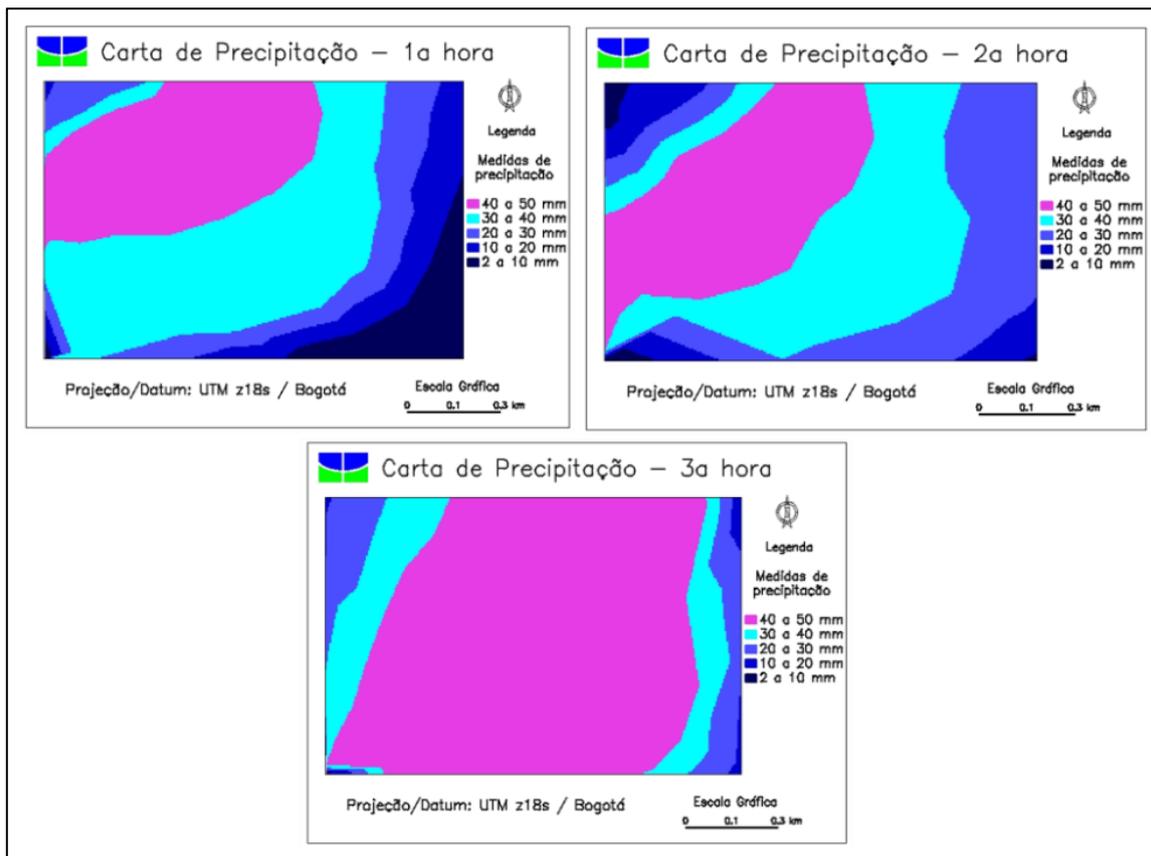


Figura 4.4. Configuração para o evento de precipitação aplicado à análise de ameaça.

Diante disto, executou-se a análise precedente que forneceu o panorama de ameaça, refletindo a conjuntura imposta pela simulação. Destaca-se que o tempo de processamento necessário para a obtenção dos limiares em função da precipitação antecedente foi de 72 min para a opção de toda a extensão da área piloto. Pelo fato de assumir a conjuntura mais desfavorável à estabilidade das encostas, elegeu-se a grade numérica, que traz os resultados da probabilidade de ruptura para o acumulado pluviométrico antecedente de 350 mm/mês, combinada com a configuração do evento de chuva ligado à terceira hora, como a representante da mensuração da ameaça imposta à área piloto.

A respeito das edificações, as informações que se encontram disponibilizadas referem-se somente à projeção das construções, sem fazer qualquer menção sobre suas características. Assim sendo, houve a necessidade de definir um conjunto de classes que pudessem representar as propriedades das estruturas presentes na área piloto, apoiando-se, para tanto, nos fatores pré-determinados presentes no sistema computacional. Determinaram-se quatro categorias de características para as edificações pertencentes à área piloto, cujos critérios de atribuição dependem da área de projeção das estruturas. As classes propostas com as suas respectivas características encontram-se citadas na Tab. 4.7.

Tabela 4.7. Critérios de classificação e características atribuídas às construções situadas na região analisada.

Classes	Critérios	Características das estruturas				
		ξ_{sty}	ξ_{smn}	ξ_{sht}	Valor limite de deformação	Profundidade das fundações (m)
1	Área $\leq 50 \text{ m}^2$	0,1	0,1	0,1	0,002 (Inclinação parcial da fundação)	0,6
2	$50 \text{ m}^2 < \text{Área} \leq 150 \text{ m}^2$	0,2	0,4	0,1	120 mm (Recalque)	1,5
3	$150 \text{ m}^2 < \text{Área} \leq 250 \text{ m}^2$	0,4	0,8	0,1	120 mm (Recalque)	2
4	Área $> 250 \text{ m}^2$	0,8	1,2	0,4	200 mm (Recalque médio da fundação)	3

A atribuição de cada característica às construções foi realizada com o auxílio do software Spring 5.2, em quatro planos de informações distintos, resultando, ao serem exportados, em parte dos arquivos de entrada necessários para a determinação da vulnerabilidade pelo sistema.

O cenário fornecido pela quantificação da ameaça orientou a delimitação dos deslizamentos que seriam simulados, estando estes condicionados às regiões com as maiores porcentagens de probabilidade de ruptura. Como o módulo das análises baseadas nos autômatos celulares, responsável pela geração das trajetórias dos deslizamentos e suas características de forma automática, ainda não tinha sido desenvolvido neste ponto da pesquisa, fez-se necessário o delineamento manual destes eventos, bem como uma estimativa dos aspectos de velocidade e de áreas de acúmulo de detritos a eles associados.

Nas áreas comportando níveis de probabilidade de ruptura acima de 12%, demarcaram-se 25 feições de escorregamentos inventariadas, tomando-se como referência o princípio de maiores declividades nas proximidades de edificações. A Fig. 4.5 representa a distribuição das probabilidades de ruptura pela área piloto sob a forma da carta temática, derivada da matriz que se apresentou como fruto do módulo de análise de ameaça, em conjunto com as delimitações concernentes aos deslizamentos considerados.

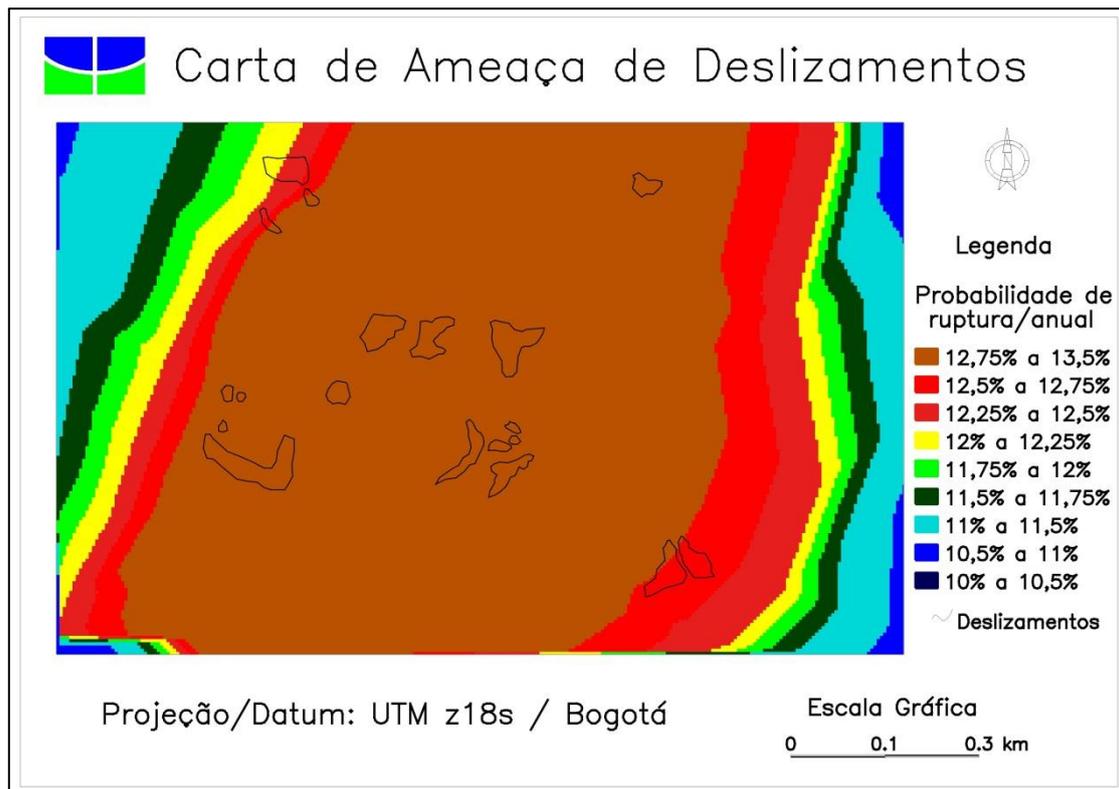


Figura 4.5. Carta de ameaça com a identificação dos deslizamentos utilizados nas análises.

Conferiram-se quatro possíveis classes de velocidades médias de deslizamentos, voltadas para a associação dos polígonos dos acidentes. A atribuição dos valores às feições foi realizada considerando a tendência direta entre a magnitude do acidente e a velocidade adquirida pela massa deslizante.

Dividiram-se as regiões possivelmente impactadas por estes escorregamentos em zonas distintas, cada uma representando uma profundidade média de detritos. Neste exemplo, três zonas foram utilizadas, representando, por sua vez, espessuras médias de 8 m, 2,5 m e 1 m, sendo que a distribuição da quantidade de zonas atingidas, por deslizamento, variou entre duas e três, apoiando-se na dimensão dos episódios previstos. Na Fig. 4.6 está demonstrada a distribuição das velocidades dos escorregamentos, assim como as faixas de acumulo de detritos e o posicionamento das construções em relação aos acidentes. Para as edificações localizadas no interior de uma determinada massa deslizante, a deformação por elas apresentadas, em decorrência desta condição, foi estipulada em função da velocidade média conferida ao escorregamento, no qual a construção encontra-se situada, e do tipo da estrutura, seguindo os valores inclusos na Tab. 4.8.

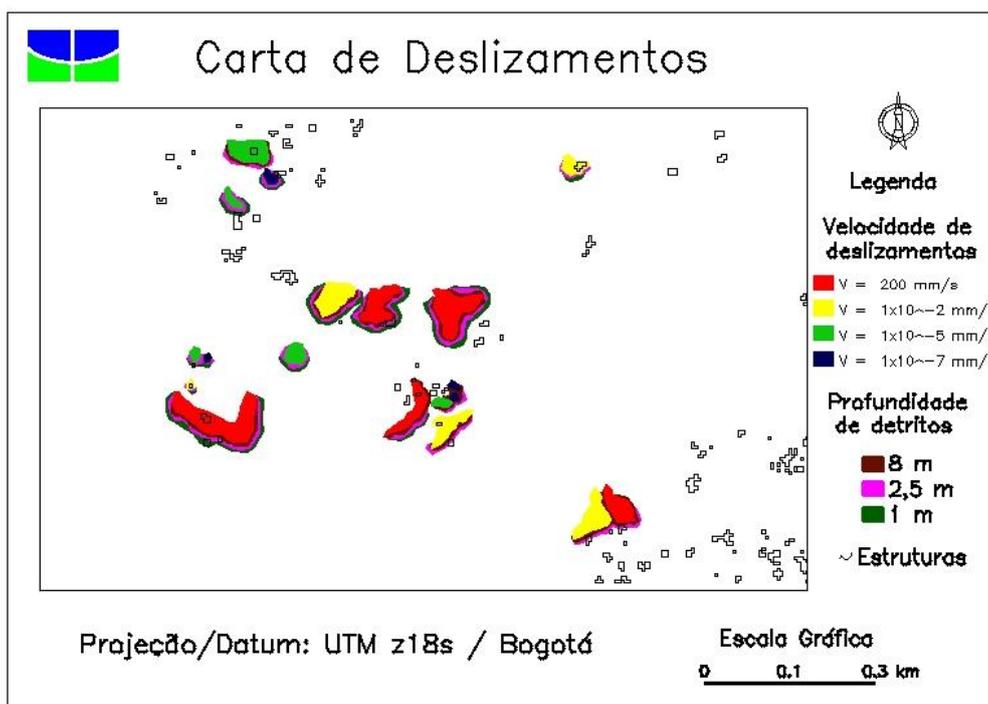


Figura 4.6. Representação das estruturas, das velocidades e áreas de acumulo de detritos para os deslizamentos pertencentes à área piloto.

Tabela 4.8. Deformações impostas às estruturas presentes no interior de deslizamentos.

Critérios	Medida de deformação	Velocidade de deslizamento (mm/s)			
		1×10^{-7}	1×10^{-5}	1×10^{-2}	200
Área $\leq 50 \text{ m}^2$	Inclinação parcial da fundação	5×10^{-4}	1×10^{-3}	2×10^{-3}	1×10^{-2}
$50 \text{ m}^2 < \text{Área} \leq 150 \text{ m}^2$	Recalque (mm)	50	100	150	300
$150 \text{ m}^2 < \text{Área} \leq 250 \text{ m}^2$	Recalque (mm)	20	50	90	200
Área $> 250 \text{ m}^2$	Recalque médio da fundação (mm)	20	50	80	200

As consequências das rupturas, em termos de custos monetários, foram retiradas do trabalho de Restrepo (2011), onde se encontram compilados os valores econômicos das estruturas para Medellín, de acordo com os estratos socioeconômicos que compõem a população da cidade. Dentre os seis estratos citados pela pesquisa, consideraram-se as quatro primeiras configurações concernentes às edificações residenciais, de forma que, neste caso também, os intervalos relativos às áreas construídas serviram para a distinção entre os valores concedidos às diferentes estruturas existentes na região, conforme expresso na Tab. 4.9.

Tabela 4.9. Valores de custo conferidos às edificações (modificado de Restrepo, 2011).

Classes	Cr�terios	Custo monet�rio das estruturas (US\$/m ²)
1	�rea ≤ 50 m ²	223
2	50 m ² < �rea ≤ 150 m ²	277
3	150 m ² < �rea ≤ 250 m ²	415
4	�rea > 250 m ²	672

A matriz de custos foi montada de modo que, a cada elemento constru do, este ter  sua  rea de proje o multiplicada pelo respectivo valor monet rio a que lhe   imposto, sendo salvo na sua pr pria localiza o dentro da  rea piloto.

Com isso, todos os elementos necess rios para a consolida o da carta de risco ficaram definidos. Como a etapa decorrente da an lise da amea a j  se encontra demonstrada, a averigua o da vulnerabilidade e a importa o da grade de consequ ncias econ micas tornaram-se os passos faltantes   efetiva o do produto final buscado. Conforme instituido pela metodologia, apenas as obras sujeitas   interfer ncia de eventos de escorregamentos, ou por pertencerem    rea do acidente, ou por estarem na faixa impactada, possuem valor n o nulo de vulnerabilidade. As medidas de vulnerabilidade s o apresentadas pela Fig. 4.7.

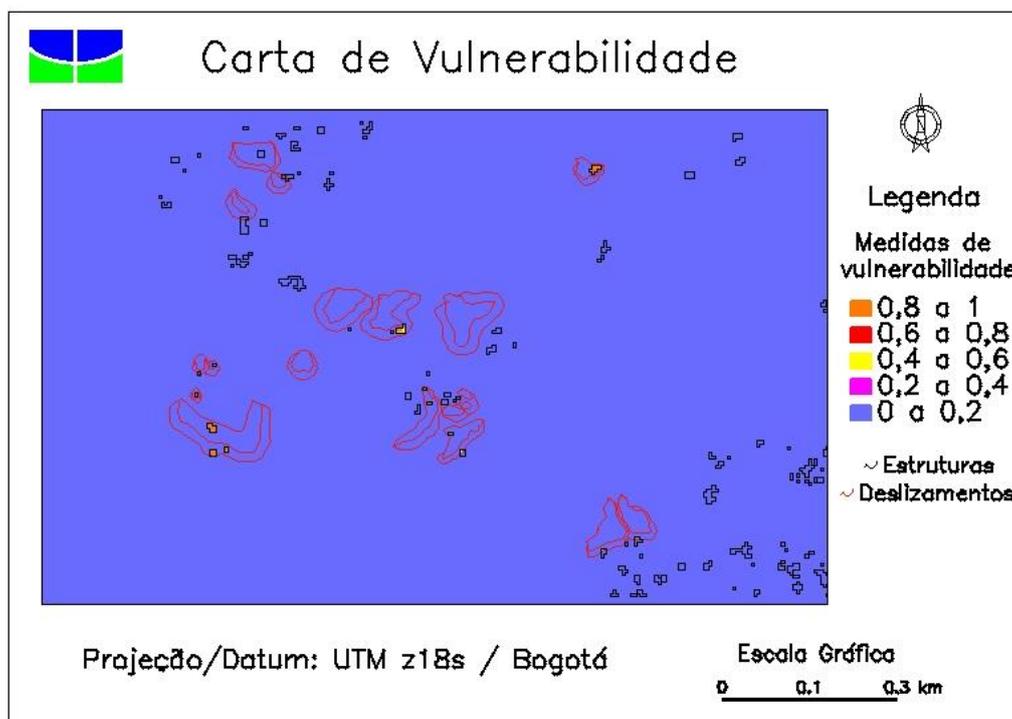


Figura 4.7. Carta de vulnerabilidade para as estruturas na  rea piloto.

É observado que, em alguns casos onde existem edificações situadas em posições de influência dos acidentes assinalados, a vulnerabilidade fornecida pelo programa se posicionou no menor intervalo proposto. Isto decorre em virtude da relação entre características das estruturas e dos escorregamentos, combinando obras de maior resistência com velocidades menores para os eventos, o que favorece uma integridade maior dos elementos em risco.

A carta de risco para o cenário de ameaça adotado no exemplo é confeccionada a partir da classificação da grade numérica ditada pelo sistema. Basicamente, as regiões com maior grau de risco seguem a tendência comandada pela vulnerabilidade, sendo que o valor máximo encontrado atingiu US\$ 8.113, como é mostrado na Fig. 4.8.

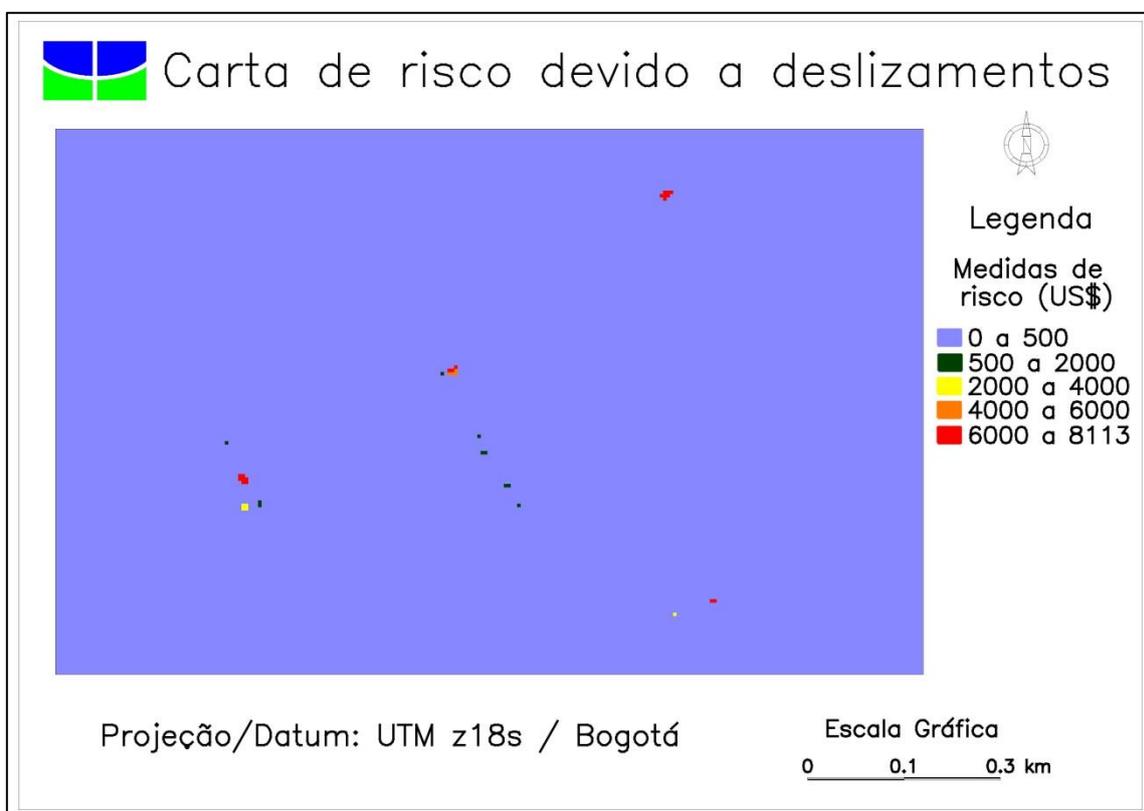


Figura 4.8. Carta final de risco para a área piloto.

Por se tratar de uma localidade essencialmente rural e, portanto, ser pouco povoada, o panorama ligado à distribuição dos escorregamentos não interferiu em um grande número de edificações, levando à formação de uma carta de risco, em termos de perdas, definida por patamares baixos, no geral.

4.3 CENÁRIOS DE AMEAÇA DE DESLIZAMENTOS PELO USO DE CURVAS CONTÍNUAS DE PROBABILIDADE DE RUPTURA EM FUNÇÃO DA CHUVA

A utilização de modelos que empregam funções para descrever a probabilidade de ruptura a partir da chuva, representou o foco direcionado a este caso analisado, especificamente a opção concernente às curvas contínuas. O objetivo principal se concentrou em avaliar configurações reais de precipitações ocorridas na cidade do Rio de Janeiro, as quais foram captadas pelo Sistema Alerta-Rio, no sentido de simular e quantificar a sua ação em relação à estabilidade das encostas.

A metodologia aplicada para a determinação dos cenários de ameaça associados a escorregamentos tomou como base, em termos de informações iniciais, o trabalho desenvolvido por Azevedo (2011), já mencionado anteriormente. No presente trabalho, foram empregadas as curvas obtidas para as precipitações acumuladas de 24 h para a definição das probabilidades de ruptura.

Como passo inicial, delimitou-se uma sub-região contida na área de estudo original proposta por Azevedo (2011), considerando, como critério para esta seleção, a existência de delimitações entre as áreas de influência de diferentes estações. Tal medida permitiu verificar o comportamento diferenciado de cada estação frente à ação das chuvas, uma vez que cada estação é representada por uma dada curva. A área selecionada englobou cinco estações distintas, as quais estão mostradas na Fig. 4.9.

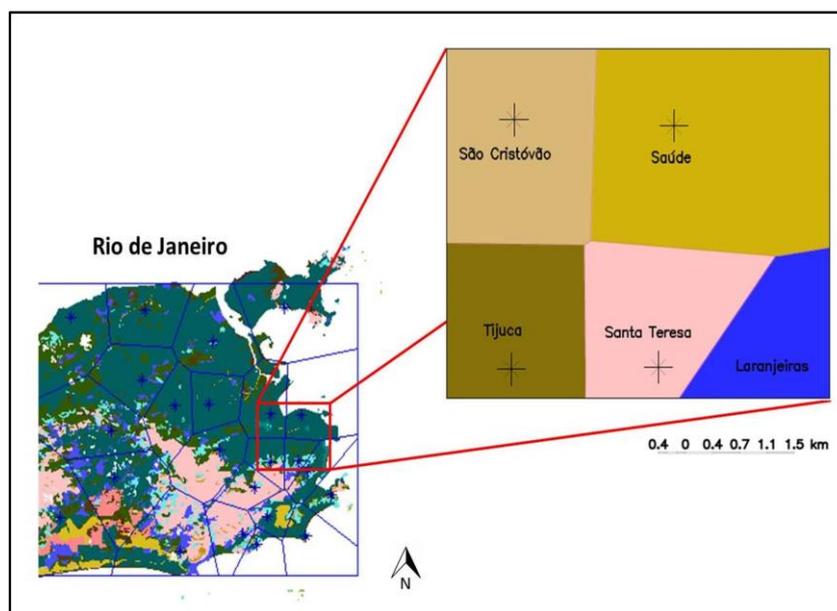


Figura 4.9. Destaque da área de estudo utilizada na pesquisa.

Dentre os postos pluviométricos existentes na área de estudo, observa-se que três deles possuem as mesmas curvas que relacionam probabilidade de ruptura dos taludes e precipitação acumulada. A Tab. 4.10 retrata os coeficientes que definem as diversas funções gaussianas empregadas no trabalho, as quais estão mostradas na Fig. 4.10. Estes coeficientes procedem do trabalho original de Azevedo (2011).

Tabela 4.10. Coeficientes que definem as curvas de probabilidade de ruptura em função da chuva acumulada.

Estações	Coeficientes das curvas (24 horas)		
	a	b	c
São Cristóvão	0,1686	138,3	77,14
Tijuca	0,1686	138,3	77,14
Saúde	0,1686	138,3	77,14
Santa Teresa	0,4321	144,7	38,87
Laranjeiras	0,2865	155,1	65,76

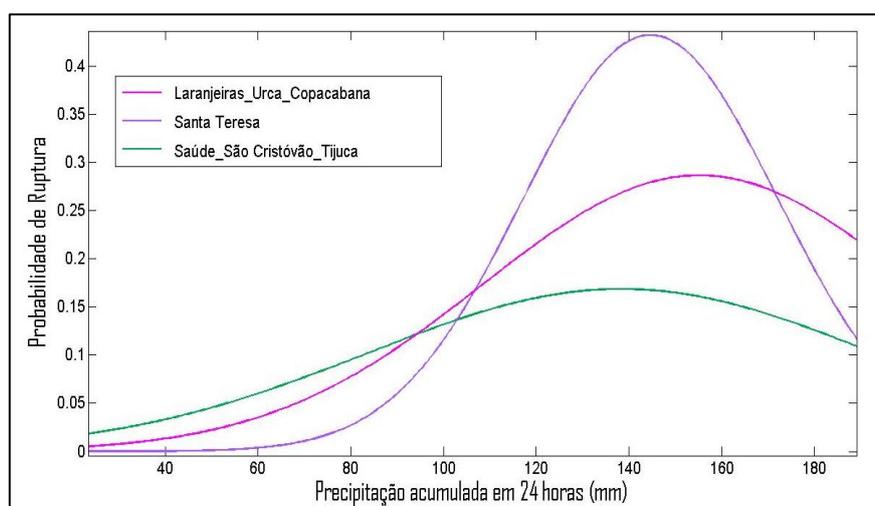


Figura 4.10. Curvas de probabilidade de ruptura em função da precipitação acumulada em 24 horas empregadas na pesquisa (Azevedo, 2011).

Para estes postos pluviométricos, efetuou-se o estudo das suas séries de chuva correspondentes ao período compreendido de 2010 a 2012, definindo-se as medidas pluviométricas mensais para cada uma das estações envolvidas dentro deste tempo. Com isto, foi possível verificar, historicamente, que abril se apresentou, em média, como o mês mais chuvoso. As médias pluviométricas mensais alcançadas para o período considerado encontram-se explicitadas na Tab. 4.11

Tabela 4.11. Médias pluviométricas mensais das estações pluviométricas envolvidas para os anos de 2010 a 2012.

Mês	Precipitação média (mm)
Janeiro	166
Fevereiro	46
Março	166
Abril	231
Maio	104
Junho	62
Julho	54
Agosto	23
Setembro	68
Outubro	104
Novembro	82
Dezembro	142

Uma vez definindo abril como o mês mais chuvoso em média, passou-se a estudar as precipitações diárias para este mês, em relação aos três anos englobados na pesquisa. Para os registros de certo ano, buscou-se, então, definir o pico de chuva alcançado em 24 h, neste mês, em relação a todas as estações envolvidas. Os eventos pluviométricos simulados em cada ano foram compostos pelo maior valor pluviométrico diário encontrado em uma determinada estação, juntamente com os respectivos valores registrados em 24 h nos demais postos pluviométricos. Na Tab. 4.12, encontram-se os valores de precipitação que foram empregados nas análises.

Tabela 4.12. Valores de precipitação acumulada em 24 h para o mês de abril de cada ano utilizados na simulação.

Ano	Postos pluviométricos				
	Saúde	Santa	Laranjeiras	São Cristóvão	Tijuca
2010	124	190,8	180	117,6	216,8
2011	47	83	57,2	16,4	171,6
2012	4,8	7,8	11	3	14,4

Pelo uso do software Spring 5.2, foi possível encontrar uma representação da distribuição da chuva por toda a área de estudo para cada ano, a partir da interpolação espacial das amostras pontuais de precipitação mostradas na Tab. 4.12, atribuídas às suas devidas localizações na região analisada, usando um interpolador do tipo Spline Mitasova.

Operação semelhante foi realizada perante os três coeficientes que definem cada uma das curvas de probabilidade de ruptura para as estações envolvidas, de forma que, para cada coeficiente, gerou-se um plano de informação resultante da interpolação dos seus valores individuais dispostos nas respectivas posições das estações. Isto garantiu o processamento das análises, de forma contínua, por toda a extensão. Ressalta-se que os planos de informação empregados apresentaram pixels com resolução espacial de 10 m.

Estas informações espaciais foram exportadas pelo software Spring 5.2, sob a forma de grades regulares, servindo de insumo para a rotina computacional correspondente à opção de análise definida como “Curvas de Probabilidade de Ruptura Contínuas” no menu para os modelos de curvas de PR, presentes na Fig. 3.9. As cartas temáticas conseguidas por meio das interpolações espaciais e que compreendem os panoramas de precipitação usados nas análises, encontram-se mostrados na Fig. 4.11.

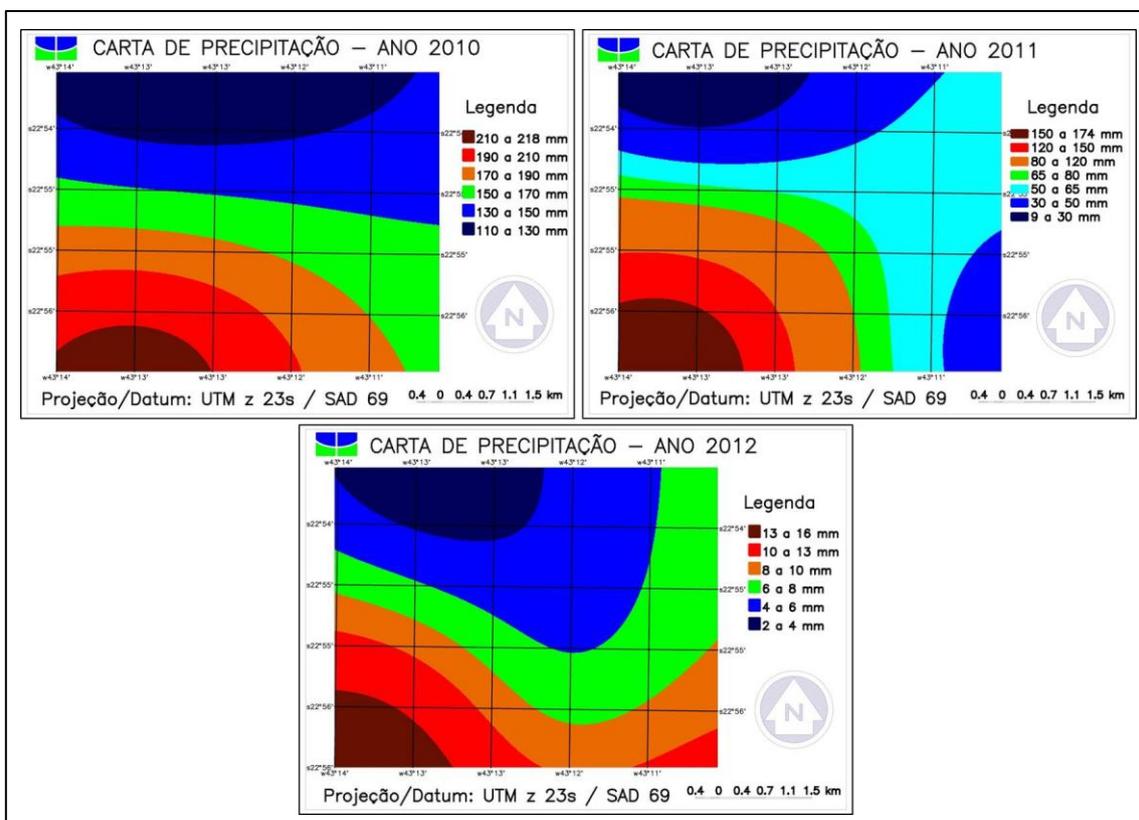


Figura 4.11. Cenários de precipitação acumulada em 24 horas, que foram simulados na pesquisa para os meses de abril.

Como resultados, destacam-se as configurações de probabilidade de ruptura alcançadas nas análises, apresentadas nas cartas confeccionadas, referindo-se à distribuição espacial da ameaça pela área de estudo, as quais derivam dos delineamentos da pluviometria aplicada à pesquisa. As Fig. 4.12, Fig. 4.13 e Fig. 4.14 referem-se a estas representações.

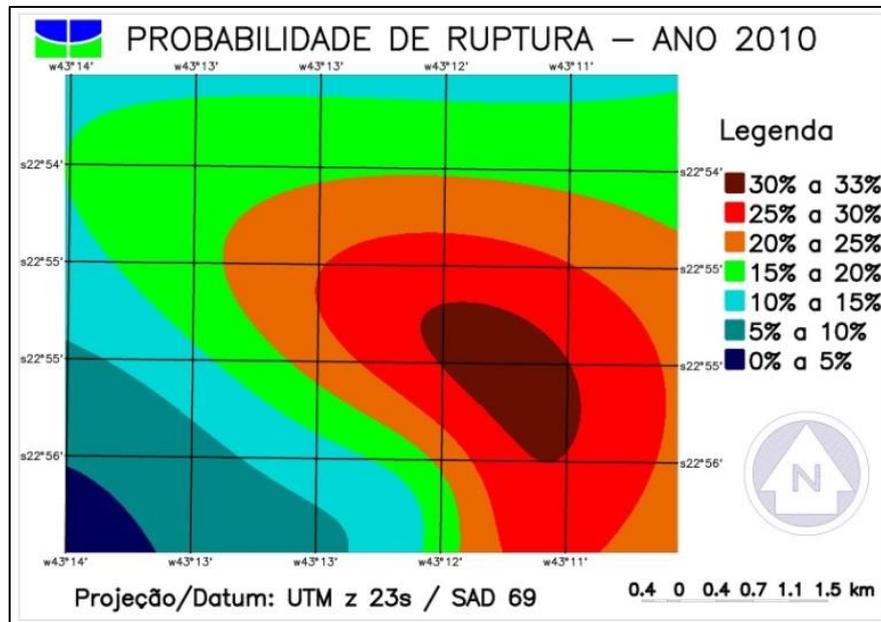


Figura 4.12. Carta de ameaça para a configuração de chuva referente ao mês de abril, para o ano de 2010.

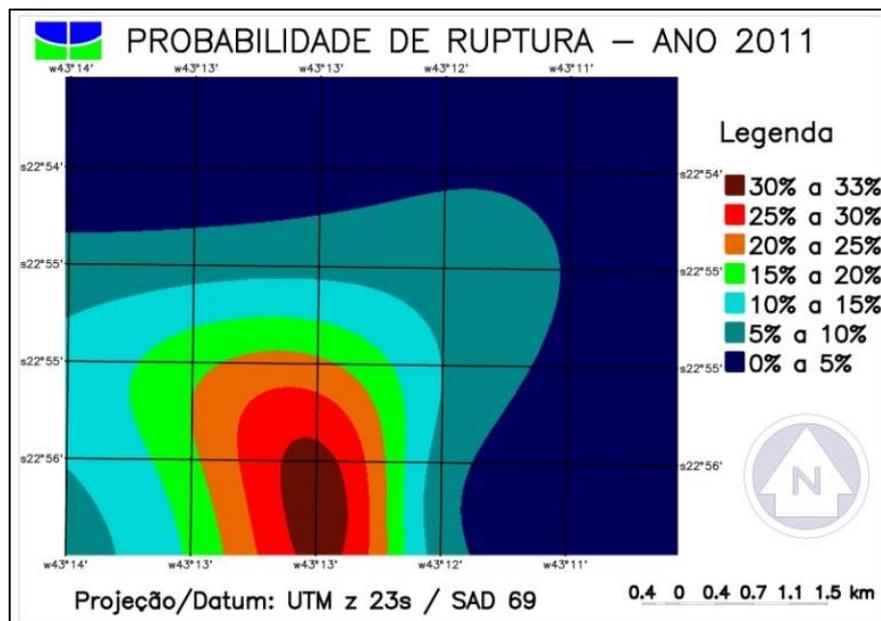


Figura 4.13. Carta de ameaça para a precipitação relativa ao mês de abril, para o ano de 2011.

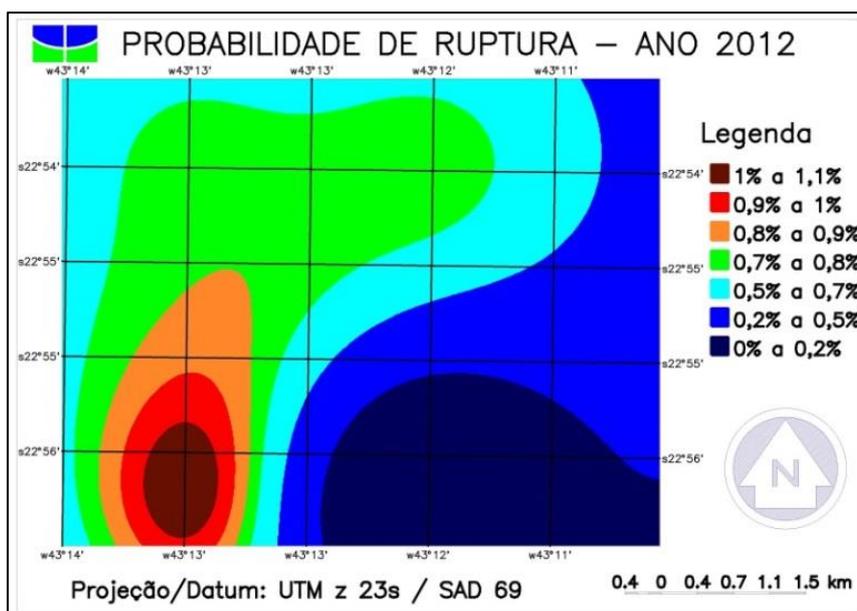


Figura 4.14. Carta de ameaça para a configuração de chuva referente ao mês de abril, para o ano de 2012.

A partir desses mapas, fica clara a atuação da medida de chuvas no estabelecimento do panorama de ameaça associado a escorregamentos. Nitidamente, as configurações mais desfavoráveis à estabilidade do terreno são resultado da aplicação das grades de precipitações correspondentes aos meses de abril do ano de 2010 e 2011, com as maiores probabilidades de ruptura, classificadas no intervalo que varia de 30% a 33%. Em termos absolutos, a maior medida de PR, determinada pelo método aplicado, refere-se ao valor de 32,3% para o ano de 2011. A Tab. 4.13 indica os demais limites de probabilidades de ruptura, gerados com a metodologia aplicada.

Tabela 4.13. Valores de PR máximas e mínimas obtidos nas análises.

Ano	Probabilidade de Ruptura	
	Mínima	Máxima
2010	3,8%	31,6%
2011	0,8%	32,3%
2012	0%	1,1%

Observa-se, ainda, que, de forma visual, nem sempre os maiores montantes de chuva irão definir as parcelas mais acentuadas de probabilidade de ruptura, como para o ano de 2010. No sentido de verificar tal situação, realizou-se o procedimento de tabulação cruzada entre os planos de informação relativos às probabilidades de ruptura e às precipitações.

Para o ano de 2010, as PR mais acentuadas, correspondentes à classe de 30% a 33%, estão associadas às categorias de pluviometria, que variam de 150 mm a 170 mm, com a classificação de 3,8% da área total, e de 170 mm a 190 mm, englobando apenas 0,4% da região avaliada. Classes com chuvas presentes entre 110 mm e 130 mm, relacionam-se à PR entre 10% e 25%.

Em 2011, o delineamento encontrado estabelece que chuvas entre 9 mm e 80 mm determinam níveis baixos de PR (22,27 km²), os quais variam entre 0% e 5%. Valores de probabilidade de ruptura situados entre 15% e 30% são definidos por chuvas maiores que 80 mm. A categoria de pico para PR responde ao intervalo de precipitação variando de 80 mm a 150 mm, abrangendo 1,75% de toda a região.

Comparativamente, o ano de 2012 é aquele cujo cenário pluviométrico apresentou as medidas mais amenas, em comparação aos anteriores. A categoria que varia de 2 mm a 4 mm de chuvas proporcionou as probabilidades mais baixas para este mês de abril, ligando-se a valores que variaram entre 0,5% e 0,8%. Analisando-se os extremos, foi possível verificar que os valores que definem as PR, apresentadas no intervalo de 0% a 0,2%, não correspondem aos menores valores de pluviometria, e sim a classes intermediárias, cujos limites referem-se às medidas de 6 mm a 13 mm. As maiores PR (entre 1% e 1,1%) associam-se, exclusivamente, a precipitações acima de 13 mm.

Esses resultados refletem o comportamento histórico do terreno frente ao desenvolvimento de eventos de deslizamentos, o qual se encontra embutido em cada curva empregada para a realização da análise. Em alguns casos, como em 2010, valores altos de precipitação não determinam as maiores PR, já que, conforme verificado e discutido por Azevedo (2011), nestas situações, os escorregamentos, em sua maioria, já teriam sido deflagrados por precipitações com menor intensidade. Desta forma, o incremento de chuva não aumentaria, de forma significativa, a ameaça definida por este fator.

Para o ano de 2012, como as chuvas aplicadas possuem pequena intensidade, estas correspondem ao tramo inicial e crescente das funções que relacionam precipitações e PR, provocando, com isso, valores baixos de ameaça, mas que seguem uma tendência de aumento.

Esse exemplo retrata bem o potencial existente na ferramenta implementada, quanto ao seu caráter de permitir o monitoramento de eventos associados a movimentos de massa, já que, uma vez definidas as curvas que representam a variação da PR com a quantidade de precipitação acumulada, em termos dos coeficientes destas funções, podem-se impor,

indefinidamente, planos de informações contendo dados de precipitação no sentido de gerar os cenários de ameaça correspondentes.

Ao se utilizar uma rede pluviográfica, onde os registros são efetuados, em geral, de maneira contínua dentro de um determinado intervalo de tempo, como no caso do sistema Alerta Rio, ou imagens oriundas de radares meteorológicos, capazes de estimar a intensidade da chuva, o uso do recurso apresentado permitiria uma geração sucessiva, e em tempo real, dos estados de estabilidade de encostas em uma dada área, viabilizando a implantação de um sistema de alerta contra deslizamentos.

4.4 DEFINIÇÃO DE AMEAÇA DE DESLIZAMENTOS PELA ASSOCIAÇÃO ENTRE O MODELO SLIDE PROBABILÍSTICO E AUTÔMATOS CELULARES

Um quarto exemplo empregou, inicialmente, a estrutura do modelo SLIDE, para mensurar o quadro de ameaça de deslizamentos em uma área de estudo pertencente ao Distrito Federal, mediante a imposição das configurações de chuvas características e, posteriormente, a geração das possíveis trajetórias para os prováveis eventos de deslizamentos que seriam desencadeados, tomando-se como base o cenário estabelecido pela análise de ameaça. Não foram estabelecidas, entretanto, estimativas de velocidade e espessura de detritos nesta abordagem. A área escolhida para aplicação do método exposto engloba, principalmente, a cidade satélite de Ceilândia e parte de Taguatinga, situada no DF, Brasil.

O primeiro passo consistiu na busca e aquisição dos dados pluviométricos concernentes à área de estudo. Para a definição destas informações, foram utilizadas as séries históricas das duas estações pluviométricas pertencentes à rede operada pela Agência Nacional de Águas (ANA), que se encontram no interior da região avaliada. Os postos denominam-se Taguatinga ETA RD e Taguatinga, sendo que a primeira estação possui registros de dados pluviométricos diários referentes ao período de 1971 a 2007 e o segundo posto, a partir de junho de 2009 até julho de 2011. Efetuou-se o cálculo das médias históricas das precipitações máximas diárias, para cada mês, dentro do intervalo de tempo mencionado, obtendo-se os resultados apresentados na Tab. 4.14.

Tabela 4.14. Média das precipitações máximas diárias por meses.

Mês	Média da precipitação máxima diária (mm)
Janeiro	44,5
Fevereiro	47,5

Mês	Média da precipitação máxima diária (mm)
Março	50,6
Abril	35,5
Mai	16,8
Junho	5,0
Julho	4,7
Agosto	10,3
Setembro	18,4
Outubro	39,8
Novembro	47,1
Dezembro	55,2

Optou-se por simular a ação da precipitação diária atuante de novembro a janeiro, já que este período corresponde à estação chuvosa no Distrito Federal, com valores registrados expressivos. Desta forma, gerou-se um evento baseado em três passos (três dias) sequenciais, com cada um correspondendo a um plano de informação de intensidade de chuva nos valores de 47,1, 55,2 e 44,5 mm/dia, na tentativa de simular uma situação extrema imposta ao terreno.

Ressalta-se que, em cada intervalo considerado, aplicou-se a mesma medida de chuva a toda área avaliada por meio de planos de informações homogêneos. Destaca-se, ainda, que se adotou o valor inicial do parâmetro m igual a zero para o modelo SLIDE.

Tratando-se das características físicas da área de estudo, observa-se a presença de quatro tipos pedológicos distintos, conforme a classificação fornecida por Reatto et al. (2004). As propriedades destes solos foram, então, determinadas a partir de trabalhos pertinentes na literatura técnica. Definiram-se os valores de c' e ϕ' por meio da pesquisa de Roque & Souza (2008), enquanto que se obtiveram as características S_r , n , G_s e K_t por intermédio de Castro (2011). Já os parâmetros A , λ e α , foram retirados do trabalho de Montrasio & Valentino (2008).

Com auxílio do software Spring 5.2, derivou-se, ainda, a grade numérica de declividades para a área de estudo, a qual representaria as inclinações das encostas, tomando-se como base o modelo numérico de terreno obtido junto à Companhia Imobiliária de Brasília (Terracap) e os planos de informações correspondentes às curvaturas planas e de perfil. Foram utilizados pixels com resolução espacial de 15 m.

A carta de ameaça correspondente à precipitação do mês de janeiro serviu de referência como input para as fontes dos deslizamentos a serem empregados na rotina dos

autômatos celulares. A simulação empreendida contou com testes para verificação do número de ciclos necessários para estabilizar o processo de definição das trajetórias, o qual foi definido em 100 passos. A Tab. 4.15 destaca os tipos de solos presentes na área de estudo, bem como a suas características.

Tabela 4.15. Tipos de solos presentes na área de estudo e suas características incorporadas nas análises.

Solo	Parâmetros do solo									
	c'	ϕ	H	K_t	G_s	n	S_r	A	λ	α
	kPa	°	m	1/s	-	%	%	-	-	-
Latossolo vermelho	10	20	2	10^{-6}	2,8	0,6	0,4	100	0,4	3,4
Latossolo vermelho amarelo	10	20	2	10^{-7}	2,8	0,5	0,6	100	0,4	3,4
Gleissolo háplico	20	10	3	10^{-9}	2,7	0,4	0,95	80	0,4	3,4
Cambissolo háplico	5	25	0,7	10^{-8}	2,7	0,3	0,6	100	0,4	3,4

Para a entrada desses dados no sistema, foram estabelecidas, com o auxílio da plataforma SIG, as grades numéricas correspondentes aos parâmetros do solo, devidamente distribuídos espacialmente, conforme o mapa pedológico da área avaliada mostrado na Fig. 4.15.

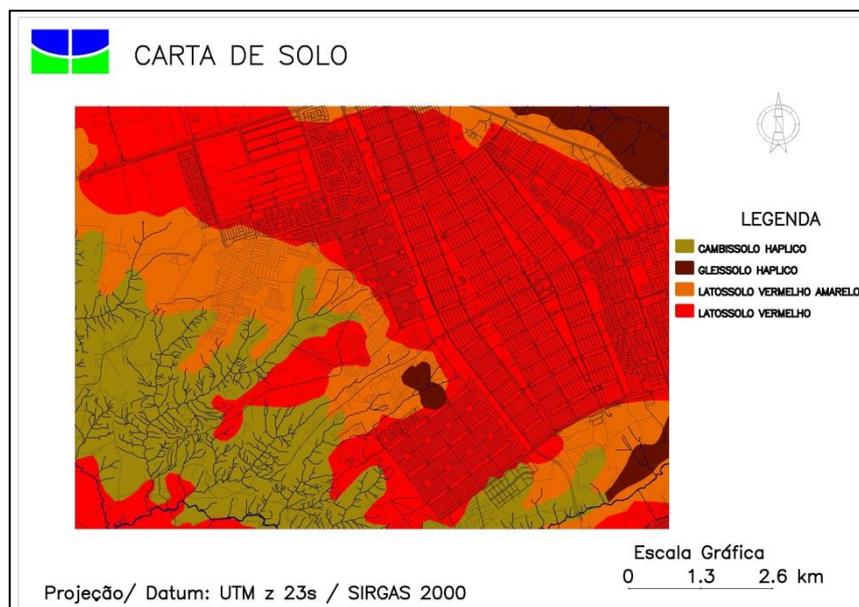


Figura 4.15. Carta de solos para a área analisada conforme Reatto et al. (2004).

Em relação às análises probabilísticas empreendidas, optou-se pela utilização do método FOSM, com coeficientes de variação únicos e considerados padrão (a coesão e o ângulo de atrito são considerados como variáveis aleatórias com coeficientes de variação iguais, respectivamente, a 40% e 10%). Para a saída de dados, escolheu-se a opção de todos os passos referente à obtenção dos cenários de PR gerados para os três meses considerados.

Destacam-se, como um primeiro produto, as cartas de probabilidades de ruptura alcançadas pelo uso das chuvas associadas aos meses de novembro a janeiro. Nestes resultados, torna-se evidente a ação da precipitação na estabilidade das encostas, por meio do acréscimo observado na quantificação da probabilidade de ruptura ao longo do período estabelecido, assim como as áreas que são mais afetadas pela atuação das chuvas. É possível observar a evolução temporal do aspecto relacionado à ameaça para a área analisada, sob a forma da PR, à medida que se impõe as configurações de chuvas diárias provenientes de cada mês, processando-se as análises relativas ao modelo empregado.

É constatado que os valores mais significativos de PR situam-se nas regiões com maiores declividades, principalmente nas vertentes ao longo das drenagens. É visível, também, o incremento da ameaça em algumas áreas, à medida que decorre a progressão temporal das análises. As configurações de PR alcançadas estão presentes em Fig. 4.16, Fig. 4.17 e Fig. 4.18.

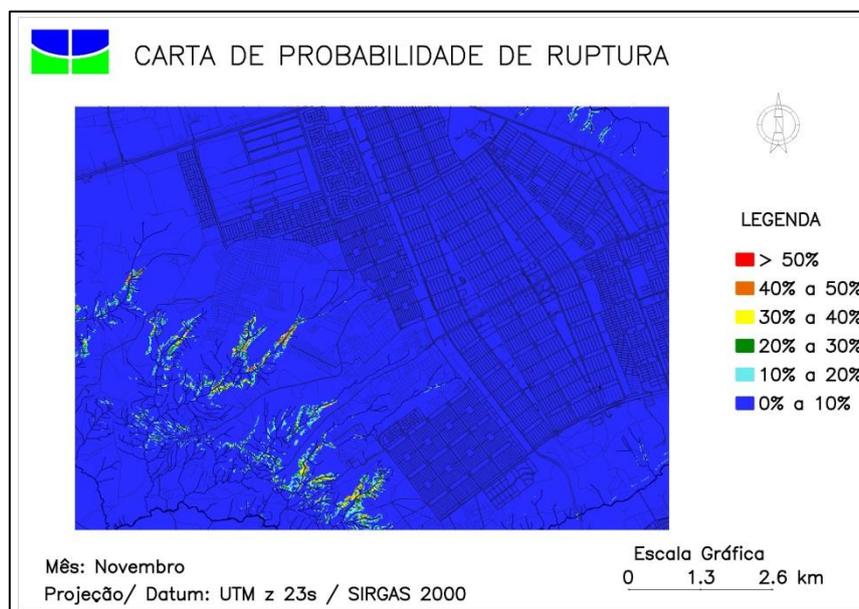


Figura 4.16. Carta de probabilidade de ruptura para a chuva diária do mês de novembro.

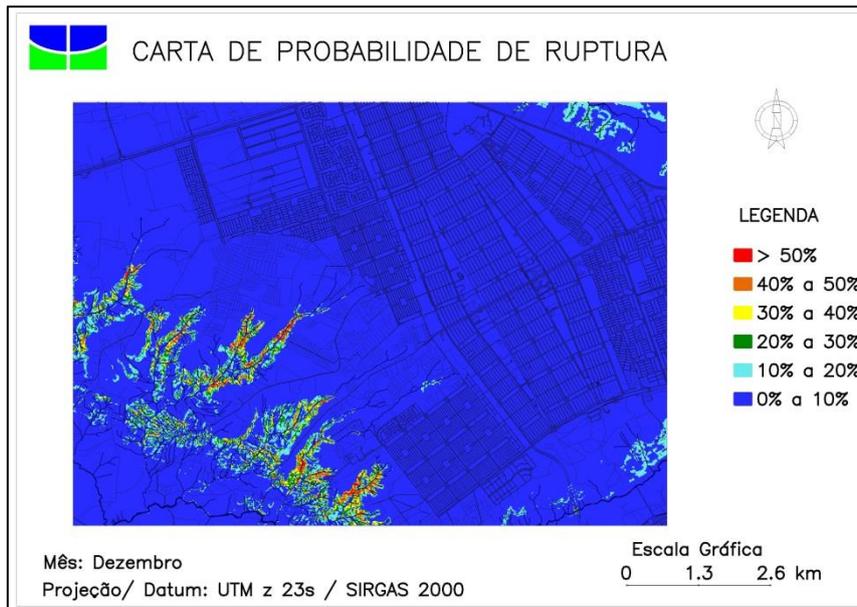


Figura 4.17. Carta de probabilidade de ruptura para a chuva diária do mês de dezembro.

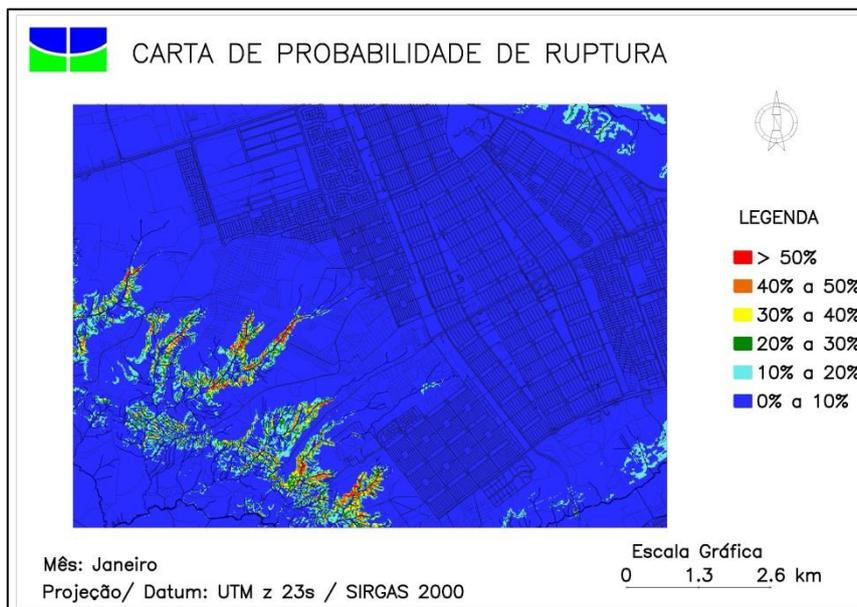


Figura 4.18. Carta de probabilidade de ruptura para a chuva diária do mês de janeiro.

Inicialmente, para a precipitação do mês de novembro, apenas as regiões com declividade mais acentuada mostram-se com probabilidades de ruptura superiores a 40%. Para a chuva do mês de dezembro, há um crescimento das medidas de PR concentrado, além da região sudoeste, nas porções nordeste e sudeste da área de estudo, com tendência de estabilização no mês de janeiro. Ao se realizar a medida das classes de PR para cada um dos planos alcançados, pôde-se constatar o crescimento das categorias que representam os maiores graus de ameaça, conforme exposto na Tab. 4.16.

Tabela 4.16. Disposição da porcentagem da área de estudo pelas classes de probabilidade de ruptura em relação às chuvas simuladas.

Classes de PR	Meses ligados à chuva simulada		
	Nov.	Dez.	Jan.
0% a 10%	98,23%	93,93%	93,89%
10% a 20%	1,08%	3,54%	3,56%
20% a 30%	0,42%	1,35%	1,36%
30% a 40%	0,19%	0,68%	0,69%
40% a 50%	0,08%	0,37%	0,37%
> 50%	0,01%	0,13%	0,13%

Com relação aos autômatos celulares, utilizaram-se as áreas com PR iguais ou superiores a 50% (correspondendo a 0,15 km²), presentes na carta de chuva ligada ao mês de janeiro, para servirem de ponto inicial dos escorregamentos, já que esta configuração foi a última da série simulada e, portanto, a mais crítica.

No momento em que se elaborou o presente caso de estudo, não se tinham disponíveis, ainda, os algoritmos responsáveis por fornecer as estimativas de velocidade e da espessura de detritos acumulada decorrente da massa deslizante, apenas o modelo qualitativo baseado no trabalho de Lai & Dragičević (2011). Os critérios de curvatura e declividade adotados para a parada do processo referiram-se a feições côncavas e inferiores a 5°, respectivamente, coincidindo com as condições padrões fixadas no sistema.

Para os resultados alcançados, observa-se que o delineamento geral das trajetórias simuladas já se estabelece na iteração 25, ampliando em menor proporção, a partir disso, sua área, principalmente ao longo dos cursos d'água presentes na região em estudo. Uma ideia da evolução da trajetória dos deslizamentos, ao longo do tempo, pode ser estabelecida por meio da mensuração da categoria das trajetórias para os distintos instantes aqui apresentados, resultando na classificação de uma área de 0,43, 0,47, 0,48 e 0,48 km² para 25, 50, 75 e 100 iterações, respectivamente. A título de exposição de resultados, dentre os passos obtidos, destacam-se as iterações de números 25, 50, 75 e 100, além do plano de informação tomado como fonte inicial dos deslizamentos, todos apresentados, sequencialmente, a partir da Fig. 4.19 até a Fig. 4.23.

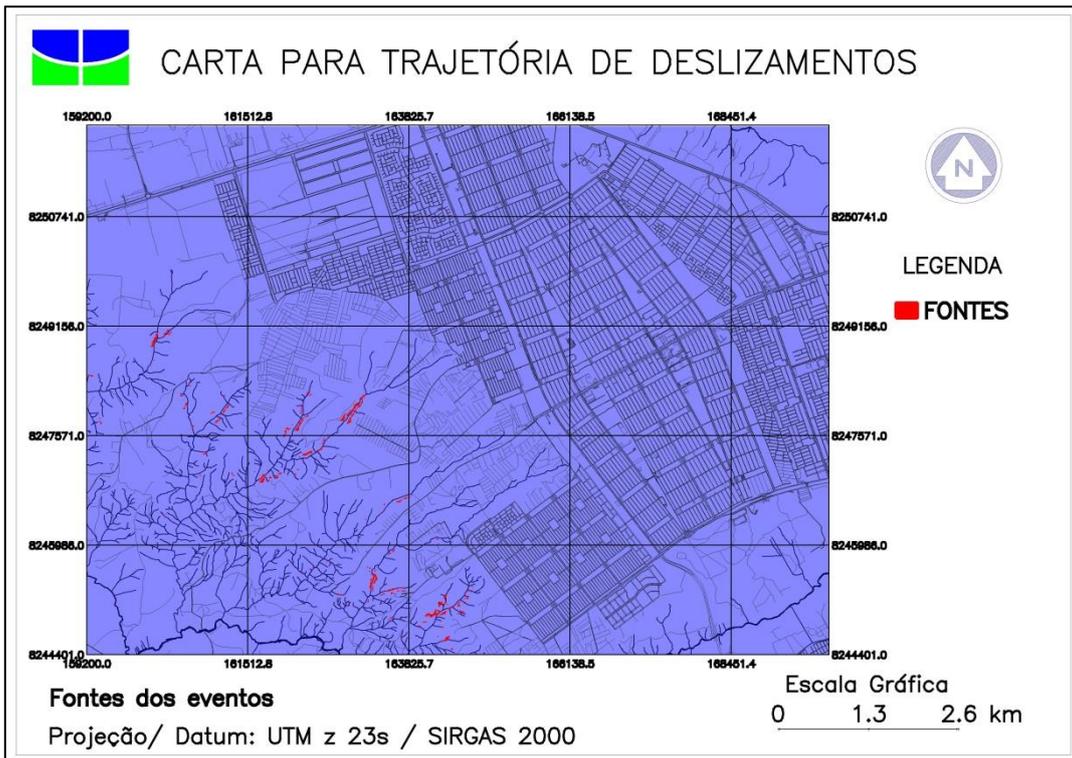


Figura 4.19. Carta com as fontes simuladas dos deslizamentos.

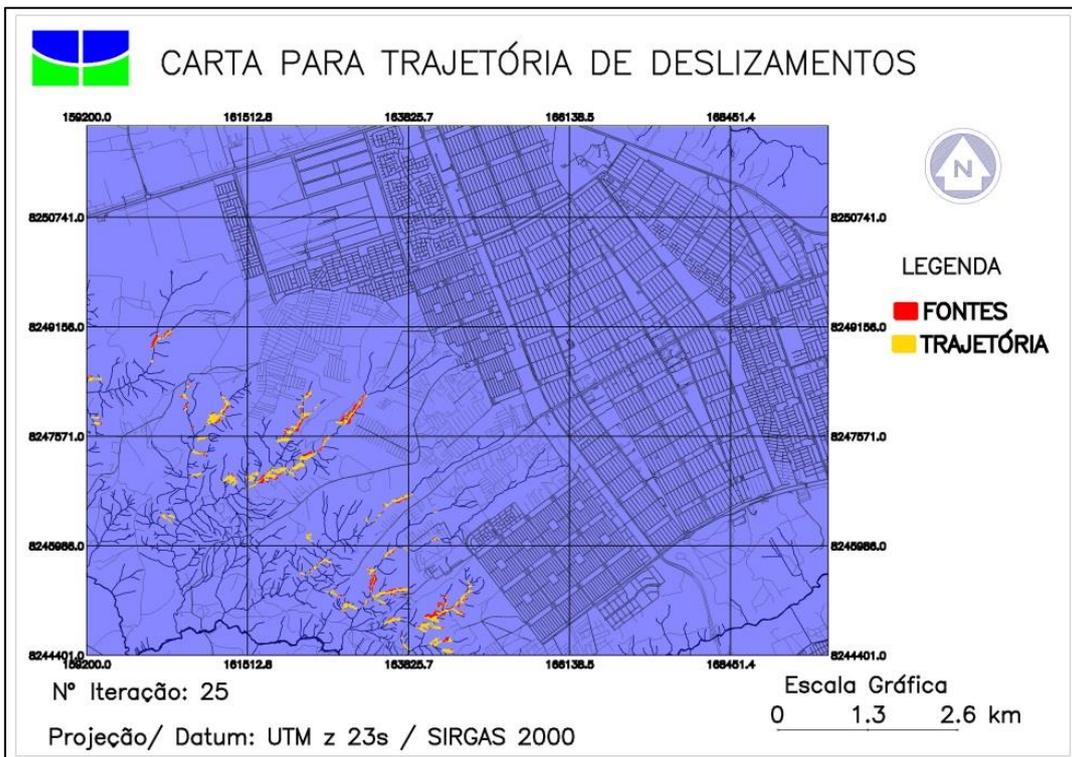


Figura 4.20. Carta com a simulação para a iteração 25.

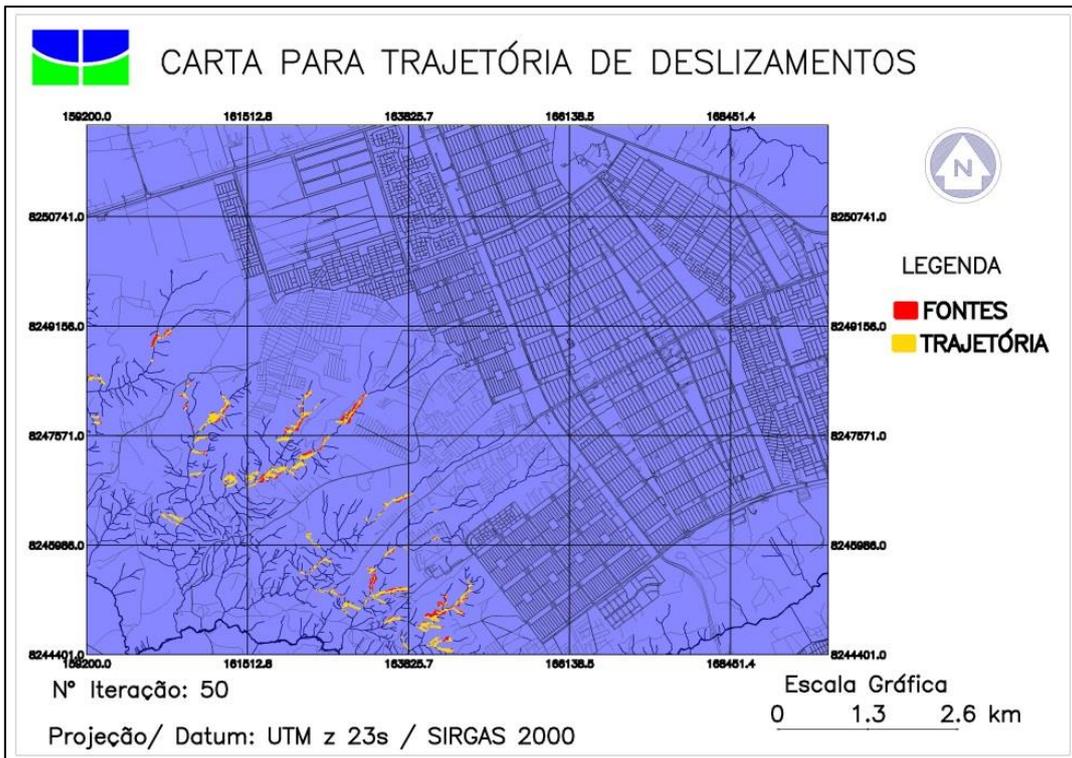


Figura 4.21. Carta com a simulação para a iteração 50.

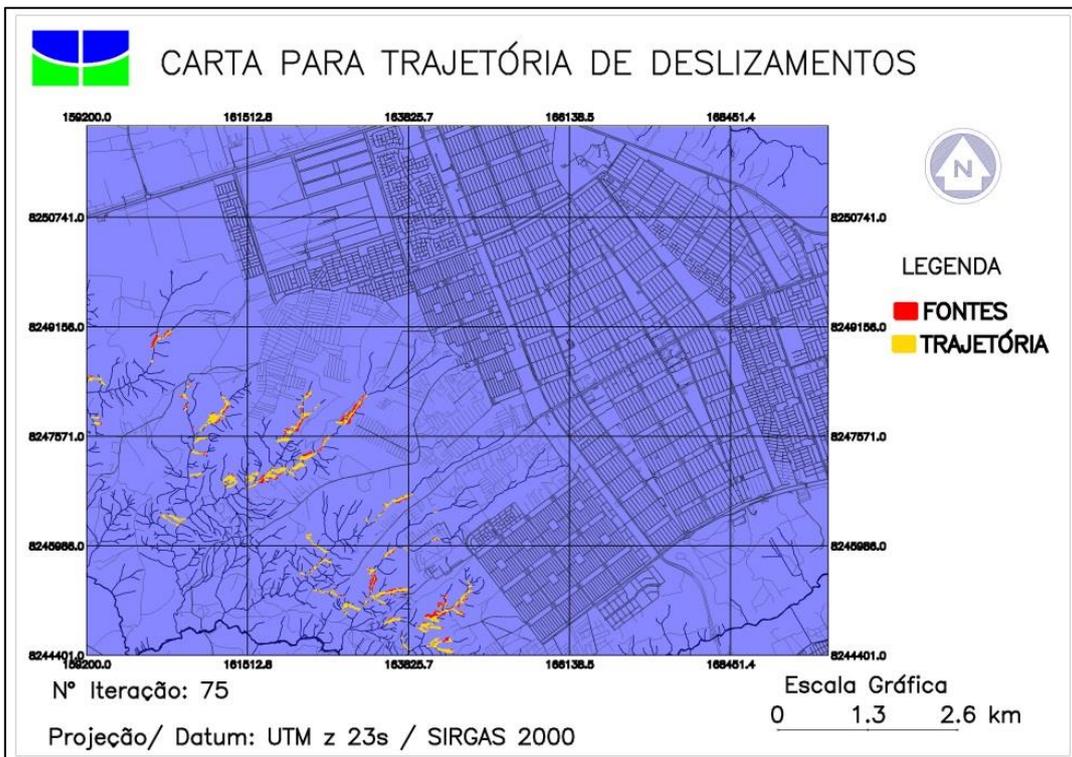


Figura 4.22. Carta com a simulação para a iteração 75.

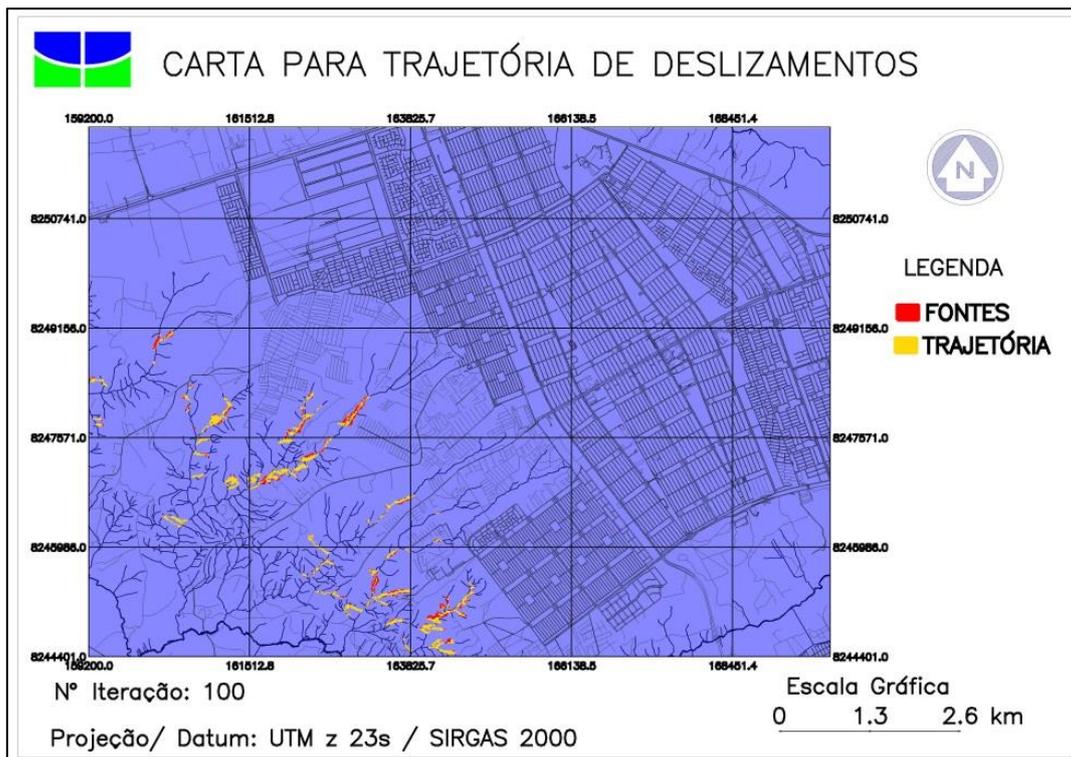


Figura 4.23. Carta com a simulação para a iteração 100.

Verifica-se que, após os 25 primeiros intervalos de tempo, 89,6% da região prevista para os escorregamentos já se encontra definida, sendo que, ao fim dos ciclos previstos, 0,6% da área total estudada, que é de 104,01 km², referem-se a regiões que seriam atingidas por movimentos de massa (considerando-se, também, suas fontes). Percebe-se que o progresso dos eventos se dá essencialmente ao longo da distribuição das drenagens, aproximando-se, em certos pontos, da infraestrutura local. Observa-se, ainda, que o processo se estabiliza a partir da iteração 75. Ao se realizar a tabulação cruzada entre os planos de informação contendo a trajetória prevista para 100 passos e a distribuição de PR obtida pela medida de chuva diária associada ao mês de janeiro, pode-se concluir que os autômatos celulares permitem avaliar a propagação dos deslizamentos, mesmo em locais inicialmente classificados com um baixo nível de ameaça (PR variando de 0% a 10%).

Alguns pontos importantes podem ser destacados, quanto às implicações alcançadas no exemplo mostrado. Verifica-se que o cenário de probabilidade de ruptura é afetado em função da ação imposta pela precipitação, a qual proporciona um incremento das áreas com valores mais altos de PR.

No fim da primeira etapa da análise, referente à chuva diária do mês de novembro, verificou-se que, apesar de se considerar a camada de solo inicialmente não saturada em toda a sua extensão, mas com uma intensidade diária de chuva moderada simulada para este passo,

algumas encostas já apresentaram comportamentos críticos em relação à estabilidade, com valores de PR que ultrapassaram 50%, referindo-se, principalmente, às regiões com maiores declividades localizadas na porção sudoeste da área de estudo.

Nas áreas com maior ameaça de ocorrência de acidentes, segundo o modelo SLIDE, não se observa a presença de infraestruturas ou edificações passíveis de danos. Entretanto, em virtude da progressão das análises com o uso dos autômatos celulares, observa-se a propagação das ocorrências na direção de áreas habitadas, de forma que isto se mostra como um aspecto relevante a ser considerado nos períodos de precipitação intensa.

4.5 PROPOSTA PARA DEFINIÇÃO PRÉVIA DO PANORAMA DE SUSCEPTIBILIDADE PELO MÉTODO SHALSTAB COM ENFOQUES DETERMINÍSTICO OU PROBABILÍSTICO

Como um último caso a ser apresentado, expõe-se uma técnica implementada com o objetivo de fornecer uma visualização prévia da situação de susceptibilidade do terreno, quanto à ocorrência de deslizamentos.

A principal aplicação da metodologia a ser explanada consiste em um estudo inicial capaz de indicar aquelas áreas com uma maior pré-disposição a desenvolverem eventos desastrosos. Isto permite que o foco dos estudos se mantenha nos pontos problemáticos, restringindo a área total a ser avaliada e, conseqüentemente, reduzindo o custo computacional e o tempo gasto no uso dos modelos do sistema.

Com isso, os usuários do programa teriam à disposição mais uma opção principal de análise dentro das já existentes, aumentando a gama de possibilidades e fornecendo uma flexibilidade maior ao sistema, ao disponibilizar uma alternativa mais rápida para um tratamento inicial.

No entanto, apesar das rotinas computacionais já estarem devidamente implementadas, depuradas e testadas, inclusive no uso do exemplo que será apresentado, a inserção desta ferramenta junto ao programa principal não se encontra efetivada. Para tanto, seria necessária a elaboração das correspondentes interfaces gráficas para a interação com o usuário, bem como a sua integração com os demais menus já existentes. Assim sendo, esta tarefa fica sujeita a um posterior desenvolvimento, constituindo-se como uma recomendação para futuros trabalhos, visando aprimorar a presente pesquisa.

O embasamento aqui desenvolvido, pautou-se na utilização de um recurso que permite avaliar a susceptibilidade do terreno ao desencadeamento de escorregamentos rasos,

confrontando, neste contexto, a perspectiva tradicional determinística com uma alternativa probabilística. Utilizou-se, neste sentido, o modelo já citado SHALSTAB, proposto por Montgomery & Dietrich (1994), o qual se baseia na associação de teorias de um modelo de estabilidade de talude infinito em conjunto com um modelo hidrológico de estado uniforme de recarga, descrito por Beven & Kirkby (1979) e O’loughlin (1986). A formulação resultante da integração dos dois métodos mencionados, que representa o referido modelo, corresponde à equação abaixo:

$$\log \frac{q}{T} = \frac{b}{a} \cdot \sin \theta \cdot \left[\frac{\rho_s}{\rho_w} \left(1 - \frac{\tan \theta}{\tan \phi} \right) + \frac{c'}{(\cos \theta)^2 \cdot \tan \phi \cdot \rho_w \cdot g \cdot z} \right] \quad (4.3)$$

onde:

q = quantidade de chuva, com unidade de m/dia;

T = valor da transmissividade do solo, dada em m²/dia;

θ = declividade, em graus (°);

a = área de contribuição, em m²;

b = comprimento de contorno da fronteira inferior de cada elemento, em m;

c' = coesão efetiva do solo, em N/m²;

ρ_s = densidade do solo, em kg/m³;

ρ_w = densidade da água, igual a 1000 kg/m³;

g = aceleração da gravidade, igual a 9,8 m/s²;

z = espessura do solo, em m;

φ = ângulo de atrito efetivo do solo, em graus (°);

Destacam-se duas condições extremas previstas pelo modelo. A primeira refere-se ao caso em que o solo encontra-se seco, com ausência de coluna d’água, na qual se observa que o ângulo de atrito é incapaz de compensar a elevada declividade da encosta, segundo a expressão abaixo:

$$\tan \theta \geq \tan \phi \quad (4.4)$$

Nesse caso, essas regiões são consideradas como incondicionalmente instáveis. De outra maneira, mesmo que a camada de solo esteja completamente saturada, se tanθ for menor ou igual a tan[1-(ρ_w/ρ_s)], o talude será tomado como incondicionalmente estável, já que não existem condições propícias para desencadear escorregamentos rasos.

Conforme mencionado por Reginatto et al. (2012), a instabilidade de uma área é diretamente proporcional à relação q/T, razão entre a quantidade de chuva e a

transmissividade do solo requerida para estabelecer uma situação contrária à estabilidade, a qual pode ser empregada para qualificar a região avaliada em um dos níveis de susceptibilidade citados por Gerente et al. (2014) e mostrados na Tab. 4.17.

Tabela 4.17. Classes de estabilidade segundo modelo SHALSTAB.

Classes de Estabilidade	Qualificação
Instabilidade crônica	Incondicionalmente instável
$\log q/T < -3,1$	Muito instável
$-3,1 < \log q/T < -2,8$	Medianamente instável
$-2,8 < \log q/T < -2,5$	Pouco instável
$-2,5 < \log q/T < -2,2$	Pouco estável
$\log q/T > -2,2$	Medianamente estável
Estável	Incondicionalmente estável

Para a implementação do modelo, valeu-se de uma relação conjunta entre instrumentos computacionais capazes de organizar e processar os dados espaciais requeridos. De forma análoga ao ocorrido anteriormente, a plataforma baseada em sistemas de informação geográfica escolhida ficou a cargo do software Spring 5.2, enquanto a manipulação e cálculos matriciais foram realizados por meio do Matlab.

Uma vez decorridas as análises, a matriz resultante do SHALSTAB foi transferida para o SIG, havendo a possibilidade de fatiamento e associação com as classes temáticas apresentadas na Tab. 4.17, considerando o atributo de qualificação.

Somando-se à abordagem determinística, a vertente probabilística correspondente ao método das estimativas pontuais, foi introduzida como uma modificação do modelo original capaz de inserir parâmetros de variabilidade em algumas variáveis independentes que constituem o modelo SHALSTAB. O método das estimativas pontuais leva em consideração, no emprego da Eq. 4.3 e, para cada pixel, as combinações das estimativas pontuais máximas ($\bar{X}_i + \sigma_i$) e mínimas ($\bar{X}_i - \sigma_i$), em relação a cada uma das i variáveis ponderadas com natureza randômica, tomando, para isto, o seu valor médio \bar{X}_i e seu desvio padrão σ_i . Isto permite compor uma amostra com 2^i elementos de $\log q/T$, da qual se pode obter uma média e um desvio padrão da variável dependente, associados a uma dada posição do terreno, e, portanto, uma distribuição de probabilidade, aqui representada como normal. O processo é realizado de maneira contínua para todas as células que representam a área de estudo, de forma semelhante ao já relatado no uso do método das estimativas pontuais associado a outros modelos, sendo

que, ao final, os dados com as médias e desvios padrões de $\log q/T$ são armazenados em grades distintas.

Foi definida como limite crítico de estabilidade, a medida a partir da qual as áreas são categorizadas como muito instáveis ($\log q/T < -3,1$). Desta maneira, a probabilidade relacionada ao método das estimativas pontuais está ligada à área a qual se encontra sob a curva unitária da função densidade de probabilidade, definida para valores inferiores ao limite de $-3,1$. A quantificação das probabilidades foi alcançada por meio do uso de integração numérica, sendo guardadas em uma grade específica, permitindo sua avaliação e comparação com o resultado determinístico.

As variáveis independentes consideradas estatísticas correspondem às características do solo referentes à coesão efetiva, ao ângulo de atrito efetivo e à densidade do solo, com coeficientes de variação, fixados segundo Assis et al. (2012), em 40%, 10% e 3%, respectivamente.

A área de estudo escolhida situa-se na bacia do Ribeirão Contagem, localizada na porção norte do Distrito Federal e pertencente à região administrativa de Sobradinho. Conforme Braga (2013), este local já foi alvo de estudos anteriores, que comprovaram a perda laminar de solo, em volumes significativos em virtude das altas declividades presentes no local, sendo palco de registros de episódios abrangendo feições erosivas e movimentos de massa. O projeto específico criado no Spring 5.2 tem como principais atributos os seguintes pontos:

- Projeção e datum utilizados: UTM/ SIRGAS 2000, zona 23;
- Retângulo envolvente, definido em coordenadas planas:
X1 = 188.000 m; Y1 = 8.268.000 m;
X2 = 194.000 m; Y2 = 8.270.850 m;

Os planos de informação apresentam resolução espacial de 5 m, valor empegado para representar a medida do comprimento de contorno da fronteira inferior de cada elemento. Os dados referentes ao meio físico, que são exigidos pelo modelo SHALSTAB foram importados para a plataforma Spring 5.2 e organizados em categorias específicas e planos de informação distintos, sob o formato matricial. Dentre eles, figura-se o modelo numérico de terreno, do qual foram derivadas as grades de declividade e de fluxo acumulado.

Destaca-se, também, que o mapa pedológico aqui utilizado advém do processo de atualização da carta de solos digital do Distrito Federal, fornecido pela Empresa Brasileira de Pesquisa Agropecuária - EMBRAPA (Reatto et al., 2004). Nesta configuração, aparecem

presentes os tipos pedológicos explicitados pela Tab. 4.18 em conjunto com suas respectivas características de resistência, em termos de valores médios, aplicadas ao trabalho.

Tabela 4.18. Tipos de solos encontrados na área de estudo, classificados conforme Roque & Souza (2008).

Pedologia	Características		
	Coesão efetiva (N/m ²)	Ângulo de Atrito (°)	Espessura (m)
Latossolo vermelho	10.000	20	3
Latossolo vermelho amarelo	10.000	20	3
Cambissolo háplico	5.000	25	0,7
Espodosolo ferrilúvico	20.000	10	4

A densidade para todas as classes de solo citadas foi considerada igual a um valor médio de 1.500 kg/m³ em épocas chuvosas, seguindo a mesma consideração proposta por Roque & Souza (2008).

Por meio da metodologia aplicada, obteve-se a geração de resultados pertinentes à pesquisa, os quais se traduzem na distribuição espacial correspondente ao indicativo de estabilidade das encostas em termos do método utilizado, bem como no comparativo efetivado entre as abordagens referentes a este artifício. A Fig. 4.24 traz a carta resultante à utilização do modelo SHALSTAB para a área de estudo.

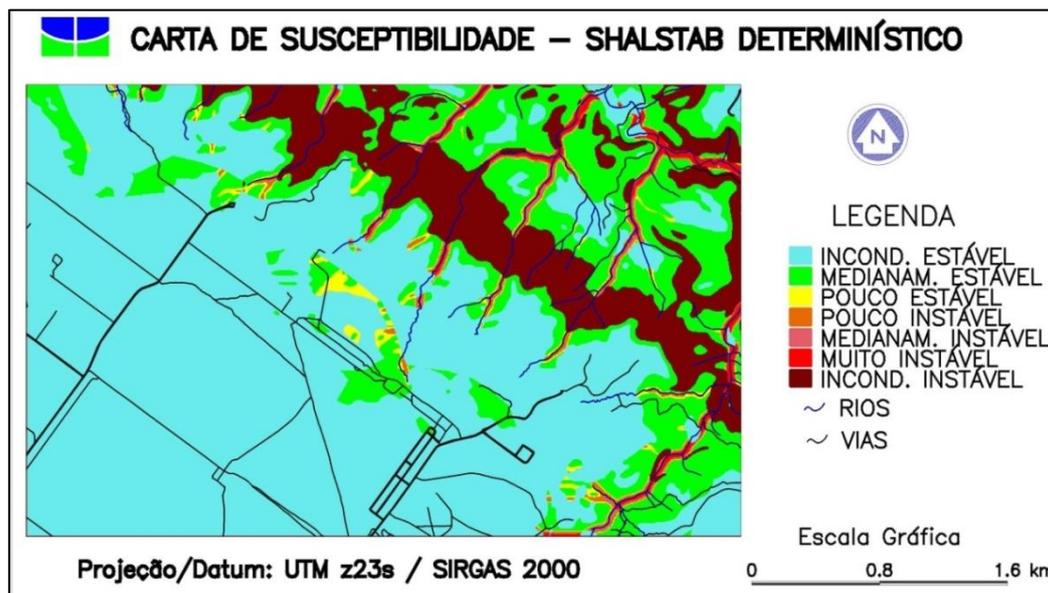


Figura 4.24. Carta de susceptibilidade à ocorrência de deslizamentos pelo uso do modelo SHALSTAB.

Por intermédio da carta apresentada, a predominância das regiões consideradas como críticas, do ponto de vista da estabilidade, categorizadas como incondicionalmente instáveis pelo modelo, estão situadas na zona intermediária entre as quebras de relevo positiva e negativa, a partir de declividades com valores superiores a 10°. Esta classe culmina, em termos de maior número de pixels assim classificados, para o intervalo de 30° a 40°. Neste sentido, é possível verificar que a predição estabelecida pelo SHALSTAB é fortemente influenciada pela distribuição das inclinações das vertentes.

Conforme os dados mostrados na Tab. 4.19, na qual se expõe a distribuição da região analisada em função da qualificação de estabilidade prevista pelo método, um pouco mais que a metade da área avaliada encontra-se estabelecida na situação incondicionalmente estável, exatamente nos pontos onde o terreno possui declividades que não ultrapassam 15°.

No limite inferior, uma percentagem de 16,6% representa os locais onde o valor da tangente da inclinação do talude suplanta a tangente do ângulo de atrito do solo.

Dentre as classes que não correspondem a uma das condições extremas previstas, aquela que mais se destaca está relacionada à característica medianamente estável, congregando um percentual considerável de 26,3%, relativos a 4,5 km² dos 17,1 km² totais analisados. As demais categorias mostram-se de forma mais modesta, no que diz respeito à participação na composição da distribuição espacial, sendo que, juntas, somam 4,7%.

Tabela 4.19. Classificação da área de estudo entre as categorias de estabilidade previstas pelo SHALSTAB.

Classes de qualificação	Porcentagem da área de estudo
Incondicionalmente instável	16,60%
Muito instável	1,27%
Medianamente instável	0,82%
Pouco instável	1,04%
Pouco estável	1,57%
Medianamente estável	26,30%
Incondicionalmente estável	52,40%

Observa-se que as regiões muito instáveis coincidem substancialmente com as áreas próximas de cursos d'água presentes na região, mostrando que as vertentes dos rios, com maiores inclinações, apresentam-se como localidades com alta susceptibilidade a escorregamentos rasos. De forma geral, maiores declividades combinadas a concentrações de

linhas de fluxo proporcionam menor grau de estabilidade, o que é exatamente observado para esta classe.

O que foi mencionado anteriormente é ratificado ao se proceder a uma análise da tabulação cruzada entre os planos de informações que contêm as grades de susceptibilidade e declividade alcançadas. Declividades baixas, variando de 0° a 5°, são mais bem representadas pelo nível de susceptibilidade denominado incondicionalmente estável. Este intervalo de declividade não sustenta regiões definidas com as duas maiores medidas de instabilidade.

Apesar de se constatar a presença de todas as classes de susceptibilidade no intervalo compreendido entre 10° a 30°, a categoria medianamente estável é aquela que predomina. Acima de 30°, encontram-se exclusivamente áreas incondicionalmente instáveis.

Já o procedimento probabilístico, visou comparar este tipo de análise com o resultado atingido de maneira determinística, por meio do panorama gerado pela imposição de um limite para o parâmetro $\log q/T$, considerado como crítico. Na Fig. 4.25, está exposta a carta alcançada na aplicação da proposta probabilística do modelo SHALSTAB.

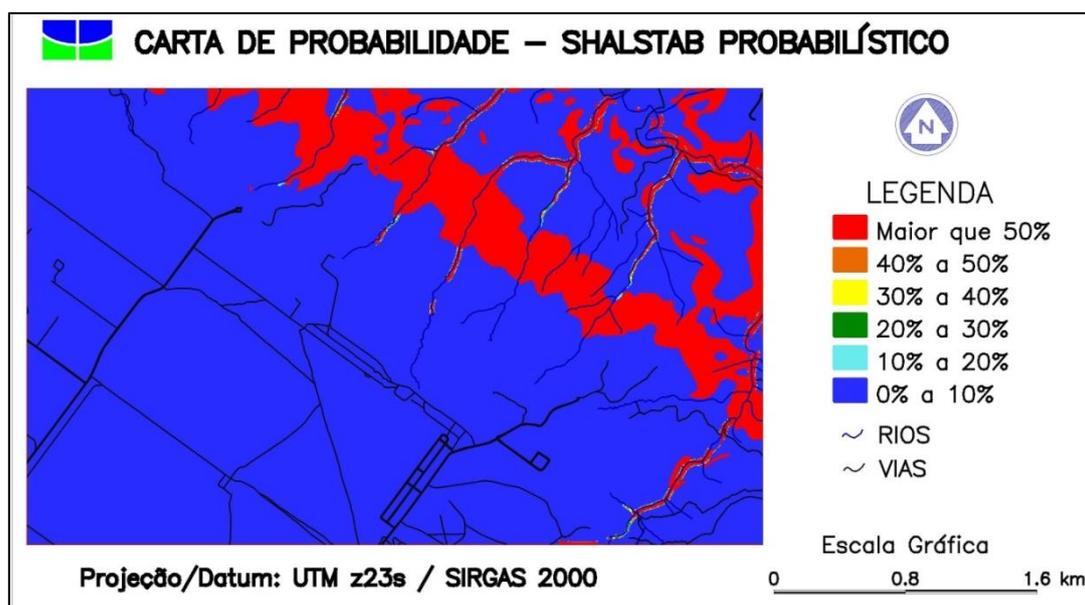


Figura 4.25. Carta de probabilidade de classificação de áreas muito instáveis, obtida pela aplicação do método de Rosenblueth (1975).

As porcentagens da distribuição para a probabilidade de que um dado pixel seja classificado na categoria de muito instável, encontram-se expressas na Tab. 4.20.

Tabela 4.20. Distribuição da região avaliada pelas classes de probabilidade para a condição muito instável.

Probabilidade de $\log q/T < -3,1$	Porcentagem da área de estudo
0% a 10%	81,56%
10% a 20%	0,19%
20% a 30%	0,12%
30% a 40%	0,10%
40% a 50%	0,11%
Maior que 50%	17,92%

A tabulação cruzada, efetuada entre os enfoques probabilístico e determinístico, forneceu subsídios para embasar as comparações entre análises.

A classe 0% a 10% representa mais que 80% do território, uma vez que também engloba, neste intervalo, as áreas consideradas como incondicionalmente estáveis pela versão determinística, cujo valor de $\log q/T$ possui baixíssima possibilidade de ser inferior ao limite crítico definido.

Nenhum pixel definido como muito instável ou incondicionalmente instável faz parte desta categoria de probabilidade. Para estas duas condições de susceptibilidade, as probabilidades que as contemplam referem-se às medidas superiores a 50%. Em contrapartida, verifica-se que elementos com condição de estabilidade igual ou superior à classe pouco instável concentram-se exclusivamente no intervalo de probabilidade considerada de 0% a 10%.

A classe medianamente instável é a única que se encontra distribuída por todos os intervalos propostos de probabilidade para $\log q/T < -3,1$. Por meio desta observação, averigua-se que diferentes elementos, por mais que estejam classificados com um mesmo nível de estabilidade frente à susceptibilidade, podem apresentar uma maior ou menor tendência, em virtude da variabilidade de suas características, de transporem o limite que distingue o patamar de muito instável. Por exemplo, 0,0432 km² da área medianamente instável têm chances que variam de 0% a 10% de transpor o limiar proposto para muito instável, enquanto 0,0087 km² da mesma classe já apresentam probabilidade muito maior, iguais ou superiores a 50%, do índice de susceptibilidade ser inferior a -3,1.

Diante dos resultados expostos, pode-se inferir que esse exemplo fornece uma óptica diferenciada sobre os alcances tradicionais promovidos pelo modelo SHALSTAB, ao incorporar considerações acerca da variabilidade dos parâmetros que o compõe,

possibilitando, assim, uma avaliação mais ampla do quão susceptível uma região se encontra. Em outras palavras, o resultado do método de Rosenblueth (1975) permite indicar qual a probabilidade apresentada por um pixel de ser classificado como um elemento muito instável.

Outro aspecto a ser considerado, quando da efetivação desta opção junto à estrutura principal, refere-se à possibilidade de incorporação da natureza probabilística de outras variáveis independentes do modelo, que possam melhor representar o comportamento do fenômeno em questão.

5 CONCLUSÕES

As análises processadas ao longo da pesquisa permitiram estabelecer algumas conclusões acerca da estrutura desenvolvida, focada na análise do risco dos deslizamentos. Ressalta-se que o presente trabalho teve como objetivo maior, a concepção e a concretização de uma metodologia capaz de inserir variados elementos que corroboram para a avaliação dos efeitos trazidos por um determinado panorama de acidentes. Conclui-se que esta meta foi devidamente alcançada, com a finalização e o bom funcionamento do sistema computacional proposto e aqui apresentado, que, de modo geral, mostrou-se como o mérito principal do trabalho.

Serão apontados, a seguir, alguns tópicos que merecem destaque, tratando-se das conclusões, em função dos principais pontos constituintes da metodologia apresentada.

5.1 MODELOS TEÓRICOS

Tratando-se dos modelos empregados no trabalho, buscou-se utilizar os mais variados elementos disponíveis no meio científico, selecionando-se as abordagens mais representativas, em termos de análises, e que melhor traduzissem os termos que determinam o risco devido a deslizamentos. Procurou-se, também, estabelecer a melhor forma de integrar estas componentes previstas por cada modelo, sob uma robusta definição de risco, na intenção de que o paradigma escolhido refletisse este conceito satisfatoriamente. A maneira encontrada para isto correspondeu à adoção do risco como a medida do produto entre a ameaça ocasionada por um cenário de eventos de escorregamentos de terra e os custos das suas consequências.

Em termos da análise de ameaça, a maior parte dos modelos escolhidos fornecia, originalmente, uma medida determinística para a representação da estabilidade do terreno, caracterizada pelo fator de segurança, exceto o método de curvas de probabilidade de ruptura conjugada à precipitação acumulada. Este último já possui, na sua formulação, um tratamento estatístico responsável por fornecer resultados relacionados à probabilidade de ocorrências de eventos de ameaça. Com relação às demais abordagens selecionadas, figuram-se a conjugação entre modelos hidrológicos, que simulam a ação da precipitação como agente deflagrador das ocorrências, e a componente geotécnica, representada pelo modelo de estabilidade de taludes infinito (análise simples e satisfatoriamente rápida, do ponto de vista computacional).

O emprego dos métodos probabilísticos no estudo, de maneira conjugada aos modelos originais para o cálculo da ameaça, permitiu a inserção de aspectos relacionados às incertezas

associadas às variáveis geotécnicas envolvidas na formulação inicial. Isto permitiu que os resultados das análises de estabilidade de talude passassem a refletir a variabilidade dos parâmetros geotécnicos envolvidos, fornecendo, ao invés de um único fator de segurança, uma distribuição probabilística para esta variável, permitindo avaliar probabilidades de ocorrência de FS menores ou iguais àquele considerado como crítico e, conseqüentemente, probabilidade de ruptura.

A forma de avaliação da vulnerabilidade, embora possua uma natureza empírica e necessite de uma análise crítica e calibração em relação aos valores dos parâmetros propostos para cada caso de estudo, tem um caráter quantitativo, essencial para o objetivo do trabalho. O modelo da vulnerabilidade leva em conta, não apenas a intensidade dos deslizamentos (função da velocidade e da profundidade de detritos de um deslizamento ou da deformação de estruturas decorrente de um evento de escorregamento de terra), mas também, a resistência dos elementos considerados expostos (relacionada à sua habilidade em resistir certa intensidade), na sua definição.

Destaca-se, ainda, que a incorporação e criação de novas técnicas como instrumentos para transpor os obstáculos revelados pelas próprias limitações impostas por outros modelos, os quais só despontaram durante o processo de desenvolvimento do sistema, tornou-se uma etapa bastante significativa dentro do trabalho. A modelagem das trajetórias dos eventos pela aplicação dos autômatos celulares e as estimativas de seus parâmetros, como velocidade média e espessura de detritos nas áreas de deposição destes deslizamentos, expressam bem a preocupação em diminuir as lacunas presentes entre os artifícios usados quando avaliados de forma individual, buscando-se a melhor integração possível entre as componentes que estruturam o conceito de risco.

Com o advento de novas pesquisas nesta área do conhecimento, espera-se que sejam desenvolvidas novas alternativas de análise que melhor representem o fenômeno estudado e que se adaptem ao contexto teórico aqui empregado. Neste sentido, em um momento futuro, pretende-se aperfeiçoar o conteúdo teórico atual, quer seja pela inclusão de um novo modelo, quer seja pela complementação ou a melhor adequação de uma técnica já existente no conjunto de opções, ou mesmo pela sua substituição por uma abordagem mais coerente.

5.2 SISTEMA COMPUTACIONAL

O sistema computacional foi concebido visando fornecer, não apenas eficiência do ponto de vista da execução dos cálculos envolvidos na análise, mas também se preocupando

com a experiência por parte dos usuários, valendo-se de interfaces gráficas que facilitassem a manipulação do programa. Entende-se que o objetivo destas interfaces foi tornar o manuseio do programa mais intuitivo quanto ao uso dos recursos disponíveis, bem como fornecer um aspecto visual mais agradável ao sistema.

Outro ponto de destaque positivo corresponde à gama de opções disponíveis, quanto às técnicas de análise, bem como as diferentes possibilidades de entrada de dados. Em relação a um mesmo modelo, foram disponíveis modos de input distintos, dentro, logicamente, das possibilidades de demanda apresentadas pela técnica, no sentido de permitir uma maior flexibilidade quanto à organização das informações iniciais requeridas, minimizando a existência de problemas desta natureza.

Fez-se uso do potencial existente no software Matlab, relacionado à manipulação de matrizes para a concepção da arquitetura mais eficiente para o sistema, no sentido de extrair e promover o máximo de desempenho do código computacional, o qual pudesse se traduzir por meio de ganho em tempo de processamento.

Diversos testes foram realizados, na tentativa de se verificar qual a melhor forma de implementação de uma dada tarefa, comparando-se os tempos gastos entre as distintas possibilidades. Um exemplo disto revela-se diante da operação relativa à multiplicação de termo a termo entre duas matrizes: verificou-se que a maior eficiência era atingida quando se utilizava o operador “.*”, responsável por estender a operação a todos os elementos da matriz de uma única vez, em contraponto ao emprego do duplo for (que varre elemento por elemento da grade).

Além disso, a variada disponibilidade de funções que já se encontram presentes no software Matlab facilitou a execução de tarefas específicas requeridas pelos métodos. Apesar disso, as dificuldades que surgiram, e que tiveram que ser contornadas, podem ser listadas, como o extenso tempo gasto inicialmente para o processamento da integração da função de probabilidade do fator de segurança para os valores menores que a medida crítica, que fornece o valor da probabilidade de ruptura. Para suplantar este problema, foi necessário comparar funções disponibilizadas pelo Matlab e ponderar os seus respectivos desempenhos frente a esta tarefa. Em alguns testes, observou-se que a entrada de dados, sob o formato de arquivos, fora dos moldes estabelecidos no trabalho ou portando informações inválidas (representadas por valores negativos nas células que não pertencem ao retângulo do projeto, por exemplo), resultou na interrupção do sistema, em decorrência da execução de alguma operação inválida, com a consequente geração de uma mensagem de erro própria do Matlab. Desta forma, um

ponto a ser considerado na etapa de melhoramento do programa corresponde ao desenvolvimento de uma série de medidas voltadas para o tratamento de exceções deste tipo e de outras naturezas, que possam ocorrer com o uso do sistema.

Em avaliações executadas com a opção do método de Monte Carlo, observou-se que, dependendo da resolução da área de estudo utilizada e o número de iterações propostas, a dimensão da matriz tridimensional requerida para o armazenamento dos valores aleatórios das variáveis é extremamente grande, acarretando erros de estouro de memória. Por exemplo, no quarto caso avaliado, referente à região de Ceilândia e Taguatinga, as matrizes utilizadas para representar a área de estudo tinham dimensões estabelecidas em 510 linhas e 907 colunas. Caso se fixasse 500 iterações para a geração dos números aleatórios, a matriz tridimensional de cada variável considerada aleatória deveria apresentar 231.285.000 elementos. Se a resolução melhorasse e a dimensão dos pixels correspondesse, agora, a 5 m, a grade do terreno seria composta por 1.529 linhas e 2.722 colunas, resultando, para as mesmas 500 iterações, em mais de dois milhões de posições necessárias para armazenar os resultados de apenas uma variável probabilística.

Para situações em que a quantidade de células de armazenamentos necessárias não é tão expressiva, como no caso do uso de resoluções menores, o método de Monte Carlo é recomendado. Espera-se testar este procedimento em computadores mais robustos, do ponto de vista da capacidade de processamento, e testar novas alternativas de implementação, mais eficazes, para este método probabilístico.

Apesar da flexibilidade fornecida quanto à criação de algoritmos, quando comparado com outras linguagens de programação, e da variedade existente de funções já implementadas, o Matlab é classificado como uma linguagem de programação de alto nível, apresentando, portanto, estas facilidades ao programador. E sendo uma linguagem de alto nível, seu desempenho é inferior, quando comparado com aquele fornecido por uma linguagem de médio nível, como C ou C++. Esta, então, refere-se a uma questão que deve ser analisada, como a possibilidade de migração do sistema para linguagens de médio nível, sugerida, inclusive, nas recomendações futuras.

5.3 CASOS DE APLICAÇÃO

Os casos de aplicação serviram, além de promover a exemplificação da atuação do sistema, para fornecer uma ideia da evolução e de como foi realizado o procedimento de desenvolvimento do programa, desde a concepção inicial, ligada à implementação simples do

modelo de taludes infinitos, associado aos métodos probabilísticos até a aplicação da ferramenta dos autômatos celulares e o cálculo de todas as etapas para o risco do deslizamentos.

Com o primeiro exemplo, foi possível averiguar que a acoplagem entre as ferramentas que utilizam SIG e a programação em Matlab mostrou-se eficaz e com uma satisfatória implementação no tratamento de problemas relacionados a estabilidades de taludes, sendo que a utilização de métodos probabilísticos possibilitou a inserção da variabilidade de alguns parâmetros constituintes do modelo de estabilidade de encostas, proporcionando o cálculo da probabilidade de ruptura. Este trabalho foi considerado como marco inicial, com os fundamentos para o desenvolvimento dos demais recursos do sistema.

O caso correspondente à região de Pajarito, na Colômbia, envolveu todas as componentes propostas para a definição de risco, contemplando a criação de uma carta, onde esta variável foi expressa de maneira quantitativa, trazendo sua classificação em categorias monetárias, conforme idealizado. Neste exemplo, os recursos de geração automática das características dos deslizamentos ainda não se encontravam desenvolvidos, de forma que a definição da evolução e das características dos possíveis eventos a serem simulados concretizou-se de forma manual.

O potencial, quanto ao uso do software em monitoramentos que envolvam características de sistemas de alerta, é verificado com a concretização do exemplo relacionado à aplicação das curvas de probabilidade de ruptura, em função da precipitação acumulada. Exige-se um esforço inicial, associado a esta opção de análise, quanto à determinação das curvas correspondentes de PR, mas, uma vez estabelecidas estas funções, pode-se impor uma grande série de grades numéricas contendo dados pluviométricos para obter os panoramas de ameaças correspondentes, de maneira relativamente rápida.

Analisou-se, ainda, o caso de integração entre o cálculo da ameaça, por meio do modelo SLIDE probabilístico, e a, até então, recém-implementada ferramenta de avaliação das trajetórias dos deslizamentos com o auxílio dos autômatos celulares. Houve a possibilidade de se verificar como a probabilidade de ruptura é afetada em função da ação imposta pela precipitação, a qual proporciona um incremento das áreas com valores mais expressivos de ameaça, assim como a propagação e alcance dos eventos por meio do modelo qualitativo, que utiliza as características geomorfológicas e topográficas como variáveis.

Por fim, uma nova abordagem, em termos de susceptibilidade, aplicando o modelo SHALSTAB, é proposta para incorporação ao sistema como melhoria e uma alternativa de

análise prévia, para a identificação das regiões mais propensas a desenvolverem eventos que venham a se tornar ameaças.

5.4 RECOMENDAÇÕES PARA PESQUISAS FUTURAS

Alguns temas são destacados, como tarefas a serem concretizadas, no intuito de aperfeiçoar as técnicas aqui desenvolvidas:

- Incorporação de novos modelos ao sistema que venham a ser criados, sob a forma de módulos adicionais, em conformidade ao objeto de estudo apresentado;
- Possibilidade de implementação do sistema em uma nova linguagem de programação, a qual forneça, comprovadamente, melhores resultados, em termos de ganhos de desempenho;
- Realização do tratamento de exceções, garantindo a intervenção diante de condições que alteram o fluxo normal da execução do sistema;
- Aplicação do sistema a novos casos de estudo, em variadas localidades;
- Elaboração de um conjunto de critérios, responsáveis por estabelecer a categorização dos níveis de risco perante um determinado contexto ambiental, social e econômico, inclusive com a definição daqueles valores considerados como aceitáveis, de forma que estes critérios deverão ser aplicados ao resultado quantitativo fornecido pelo sistema, a fim de nortear as alternativas que poderão ser aplicadas para a gestão do risco.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABNT (2009). Gestão de riscos - Princípios e diretrizes. NBR ISO 31000, Associação Brasileira de Normas Técnicas, Rio de Janeiro, RJ, 32 p.
- AKGUN, A., KINCAL, C. & PRADHAN, B. (2012). Application of remote sensing data and GIS for landslide risk assessment as an environmental threat to Izmir city (west Turkey). *Environmental Monitoring and Assessment*, 184(1): 5453-5470.
- ALEOTTI, P. (2004). A warning system for rainfall-induced shallow failures. *Eng. Geol.*, 73(1): 247-265.
- ALIMOHAMMADLOU, Y., NAJAF, A. & YALCIN, A. (2013). Landslide process and impacts: A proposed classification method. *Catena*, 104(1): 219-232.
- ALONSO, E.E., GENS, A. & DELAHAYE, C.H. (2003). Influence of rainfall on the deformation and stability of a slope in overconsolidated clays: a case study. *Hydrogeology Journal*, 11(1): 174-192.
- ÁLVAREZ, G.E.A., PEÑA, C.E.C., BECERRA, A.E.G., BELLO, E.M., CASTIBLANCO, E.A.R., PINEDA, C.E.R. & PEÑA, G.L.R. (2015). Methodology guide for research of hazard, vulnerability and risk for mass movement (in Spanish). Servicio Geológico Colombiano, Bogotá, Colômbia, 182 p.
- ANBALAGAN, R. & SINGH, B. (1996). Landslide hazard and risk assessment mapping of mountainous terrains - a case study from Kumaun Himalaya, India. *Engineering Geology*, 43(1): 237-246.
- ANDERSON, M.G. & LLOYD, D.M. (1991). Using a combined slope hydrology-stability model to develop cut slope design charts. *Proc. Inst. Civ. Engineers*, 91(1): 705-718.
- ANDERSON, S.A. & SITAR, N. (1995). Analysis of Rainfall-induced Debris Flows. *Journal of Geotechnical Engineering*, 121(7): 544-552.
- ANG, A.H.S. & TANG, W. (1975). *Probability Concepts in Engineering Planning and Design: Basic Principles*. John Wiley & Sons, New York, USA, 422 p.
- ANG, A.H.S. & TANG, W. (2007). *Probability Concepts in Engineering: Emphasis on Applications to Civil and Environmental Engineering*. John Wiley & Sons, New York, USA, 420 p.
- ARISTIZÁBAL, E.V., MARTÍNEZ, H. & VÉLEZ, J.I. (2010). A review on the study of mass movement triggered by rains (in Spanish). *Revista de la Academia Colombiana de Ciencias Exactas, Físicas y Naturales*, 34(131): 119-227.
- ARISTIZÁBAL, E.V. (2014). SHIA_LANDSLIDE: Developing a Physically Based Model to Predict Shallow Landslides Triggered by Rainfall in Tropical Environments. Doctor of Engineering Thesis, Universidad Nacional de Colombia, Medellín, Colombia, 223 p.

ARSYAD, A. (2008). The Effect of Limited Site Investigations on the Design and Performance of Pile Foundations. Thesis for the degree of Master of Engineering Science, University of Adelaide, Adelaide, Australia, 177 p.

ASSIS, A.P., ESPÓSITO, T.J., GARDONI, M.G., SILVA, P.D.E.A. & MAIA, J.A. (2012). Métodos estatísticos e probabilísticos em geotecnia. Apostila Publicação G.AP-002/2012, UnB, Brasília, Distrito Federal, 278 p.

AUGUSTO FILHO, O. & VIRGILI, J.C. (1998). Estabilidade de Taludes. Geologia de Engenharia, A.M.S. Oliveira & S.N.A. Brito (eds.), Oficina de textos, São Paulo, SP, pp. 243-269.

AZEVEDO, G.F. (2011). Análise da Relação Chuva-Escorregamentos em Regiões Tropicais Montanhosas Urbanas, Caso de Estudo Rio de Janeiro, Brasil. Dissertação de Mestrado, Universidade de Brasília, Brasília, Distrito Federal, 120 p.

BAECHER, G.B. (1987). Statistical Analysis of Geotechnical Data. Report GL-87-1, USACE Waterways Experiment Station, Vicksburg, USA, 145 p.

BAECHER, G.B. & CHRISTIAN, J.T. (2003). Reliability and Statistics in Geotechnical Engineering. John Wiley, Chichester, UK, 618 p.

BAUM, R.L., SAVAGE, W.Z. & GODT, J.W. (2002). TRIGRS - A Fortran Program for Transient Rainfall Infiltration and Grid-Based Regional Slope-Stability Analysis. Open-file Report 02-424, USGS, Colorado, USA, 61 p.

BENDA, L.E. & CUNDY, T.W. (1990). Predicting deposition of debris flows in mountain channels. *Can. Geotech. J.*, 27(1): 409-117.

BEVEN, K.J. & KIRKBY, M.J. (1979). A physically based, variable contributing area model of basin hydrology. *Bulletin of Hydrological Sciences*, 24(1): 43-69.

BIONDI, G., CASCONI, E., MAGUERI, M. & MOTTA, E. (2000). Seismic response of saturated cohesionless slopes. *Soil Dynam. Earthquake Eng.*, 20(1-4): 209-215.

BISHOP, A.W. (1955). The use of slip circle in the stability analysis. *Geotechnique*, 5(1): 7-17.

BISPO, P.C., ALMEIDA, C.M., VALERIANO, M.M., MEDEIROS, J.S. & CREPANI, E. (2009). Avaliação da suscetibilidade aos movimentos de massa na região de São Sebastião-SP com o uso de técnicas de geoprocessamento. XIV Simpósio Brasileiro de Sensoriamento Remoto, INPE, Natal, RN, 1: 3583-3590.

BLOOM, A.L. (1988). Superfície da Terra. Edgard Blücher, São Paulo, SP, 184 p.

BORGA, M., FONTANA, G.D., GREGORETTI, C. & MARCHI, L. (2002). Assessment of shallow landsliding by using a physically based model of hillslope stability. *Hydrological Processes*, 16(1): 2833-2851.

BOTERO, E.M. (2013). Modelo de alerta de escorregamentos deflagrados por chuvas usando redes neurais artificiais. Dissertação de mestrado, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 130 p.

BRAGA, L. (2013). Mapeamento de Feições Erosivas e Análise de Fatores Controladores da Bacia Hidrográfica do Ribeirão Contagem-DF. Monografia, Programa de Graduação em Geografia, Universidade de Brasília, Brasília, DF, 70 p.

BROMS, B.B. (1975). Landslides. *Foundation Engineering Handbook*, H.F. Winterkorn & H.-Y. Fang (eds.), Van Nostrand Reinhold Company, New York, USA, pp. 373-401.

CAINE, N. (1980). The rainfall intensity-duration control of shallow landslides and debris flows. *Geografiska Annaler*, 62A(1-2): 23-27.

CALVELLO, M., CASCINI, L. & MASTROIANNI, S. (2013). Landslide zoning over large areas from a sample inventory by means of scale-dependent terrain units. *Geomorphology*, 182(1): 33-48.

CÂMARA G., SOUZA, R.C.M., FREITAS, U.M., GARRIDO, J. & MITSUO, F. (1996). Spring: integrating remote sensing and gis by object-oriented data modeling. *Computers & Graphics*, 20(3): 395-403.

CARDINALI, M., GUZZETTI, F. & BRABB, E.E. (1990). Preliminary map showing landslide deposits and related features in New Mexico. Open File Report 90-293, U.S. Geological Survey, USA, 4 p.

CARRARA, A., CARDINALI, M., GUZZETTI, F. & REICHENBACH, P. (1995). GIS-based techniques for mapping landslide hazard. *Geographical Information Systems in Assessing Natural Hazards*, A. Carrara & F. Guzzetti (eds.), Kluwer Academic Publications, Dordrecht, The Netherlands, pp. 135-176.

CARRARA, A., CROSTA, G. & FRATTINI, P. (2003). Geomorphological and historical data in assessing landslide hazard. *Earth Surface Processes and Landforms*, 28(1): 1125-1142.

CARRARA, A., GUZZETTI, F., CARDINALI, M. & REICHENBACH, P. (1998). Current limitations in modeling landslide hazard. *Proceedings of IAMG'98*, A. Buccianti, G. Nardi & R. Potenza (eds.), De Frede, Naples, Italy, pp. 195-203.

CARRARA, A. & PIKE, R.J. (2008). GIS technology and models for assessing landslide hazard and risk. *Geomorphology*, 94(1): 257-260.

CASCINI, L. (2005). Risk assessment of fast landslide - from theory to practice. *Proceedings of the International Conference on "Fast Slope Movements - Prediction and Prevention for Risk Mitigation"*, L. Picarelli (ed.), Pàtron Editore, Naples, Italy, pp. 33-52.

CASCINI, L. (2008). Applicability of landslide susceptibility and hazard zoning at different scales. *Engineering Geology*, 102(1): 164-177.

- CASCINI, L., BONNARD, C., COROMINAS, J., JIBSON, R. & MONTERO-OLARTE, J. (2005). Landslide hazard and risk zoning for urban planning and development. State of Art Report, O. Hungr, R. Fell, R. Couture & E. Eberhardt (eds.), Taylor and Francis, Vancouver, Canada, pp. 199-235.
- CASCINI, L., CUOMO, S., PASTOR, M. & SORBINO, G. (2010). Modelling of rainfall-Induced shallow landslides of the flow-type. *Journal of Geotechnical and Geoenvironmental Engineering*, 136(1): 85-98.
- CASTRO, B.C. (2011). Modelo Geomecânico para os Principais Solos de Brasília. Dissertação de Mestrado, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 179 p.
- CERRI, L.E.S. & AMARAL, C.P. (1998). Riscos Geológicos. *Geologia de Engenharia*, A.M.S. Oliveira & S.N.A. Brito (eds.), Oficina de Textos, São Paulo, SP, pp. 301-310.
- CHACÓN, J., IRIGARAY, C., FERNÁNDEZ, T. & EL HAMDOUNI, R. (2006). Engineering geology maps: landslides and geographical information systems. *Bulletin of Engineering Geology & Environment*, 65(1): 341-411.
- CHEN, J.C., JAN, C.D. & LEE, M.H. (2007). Probabilistic analysis of landslide potential of an inclined uniform soil layer of infinite length: theorem. *Environ. Geol.*, 51(1): 1239–1248.
- CHENG, K.S., WEI, C. & CHANG, S.C. (2004). Locating landslides using multi-temporal satellite images. *Advances in Space Research*, 33(3): 296-301.
- CHRISTIAN, J.T. & BAECHER, G.B. (1999). Point-estimate method as numerical quadrature. *Journal of Geotechnical and Geoenvironmental Engineering*, 125(9): 779-786.
- CHRISTIAN, J.T., LADD, C.C. & BAECHER, G.B. (1994). Reliability applied to slope stability analysis. *Journal of Geotechnical and Geoenvironmental Engineering*, 120(12): 2180–2207.
- CLAESSENS, L., SCHOORL, J.M. & VELDKAMP, A. (2007). Modelling the location of shallow landslides and their effects on landscape dynamics in large watersheds: an application for Northern New Zealand. *Geomorphology*, 87(1): 16-27.
- COLLINS, B.D. & ZNIDARCIC, D. (2004). Stability Analyses of Rainfall Induced Landslides. *Journal of Geotechnical and Geoenvironmental Engineering*, 130(4): 362-372.
- CONTE, E. & TRONCONE, A. (2012). A method for the analysis of soil slips triggered by rainfall. *Geotechnique*, 62(3): 187-192.
- COROMINAS, J. & MOYA, J. (2008). A review of assessing landslide frequency for hazard zoning purposes. *Engineering Geology*, 102(1): 193-213.
- CORONADO, J.A. (2006). Avaliação da suscetibilidade a deslizamentos por meio do uso de sistema de informações geográficas: aplicação em uma área do eixo cafeiteiro colombiano. Dissertação de mestrado, Universidade de Brasília, Brasília, DF, 92 p.

- COSTA, E.A. (2005). Avaliação de ameaças e risco geotécnico aplicados à estabilidade de taludes. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 160 p.
- COUTINHO, L.A. (2010). Banco de dados geográfico de desastres naturais: projecto conceitual, inventariação e proposta para difusão dos dados. Dissertação de Mestrado, Universidade Nova de Lisboa, Lisboa, Portugal, 99 p.
- CRAIG, R.F. (2004). Craig's Soil Mechanics. Spon Press, London, England, 458 p.
- CROSTA, G. (1998). Regionalization of rainfall threshold: an aid for landslide susceptibility zonation. *Environmental Geology*, 35(2-3): 131-145.
- CROSTA, G. & FRATTINI, P. (2003). Distributed Modeling of Shallow Landslides Triggered by Intense Rainfall. *Natural Hazard and Earth System Sciences*, 3(1): 81-93.
- CRUDEN, D.M. & VARNES, D.J. (1996). Landslide types and processes. *Landslides: Investigation and Mitigation*, A.K. Turner & R.L. Schuster (eds.), National Academy Press, Washington D.C., USA, pp. 36-75.
- DAI, F.C., LEE, C.F. & NGAI, Y.Y. (2002). Landslide risk assessment and management: an overview. *Engineering Geology*, 64(1): 65-87.
- DE GRAAFF, L.W.S., DE JONG, M.G.G., RUPKE, J. & VERHOFSTAD, J. (1987). A geomorphological mapping system at scale 1:10,000 for mountainous areas. *Zeitschrift für Geomorphologie*, 13(1): 229-242.
- DIAS, L.S.O. (2006). Identificação de áreas suscetíveis a deslizamento de terra utilizando Sistema de Informações Geográficas. Dissertação de Mestrado, Escola Politécnica, Universidade Federal da Bahia, Salvador, BA, 87 p.
- D'ODORICO, P. & FAGHERAZZI, S. (2003). A probabilistic model of rainfall-triggered shallow landslides in hollows: A long term analysis. *Water Resource Research*, 39(9): 1-14.
- DUNCAN, J.M. (2000). Factors of safety and reliability in geotechnical engineering. *Journal of Geotechnical and Geoenvironmental Engineering*, 126(4): 307-316.
- EINSTEIN, H. (2003). Uncertainty in Rock Mechanics and Rock Engineering - Then and Now. 10th ISRM Congress, ISRM, Sandton, South Africa, 1: 281-293.
- ELKATEB, T., CHALATURNYK, R. & ROBERTSON, P.K. (2003). An overview of soil heterogeneity: quantification and implications on geotechnical field problems. *Canadian Geotechnical Journal*, 40(1): 1-15.
- EL-RAMLY, H., MORGENSTERN, N.R. & CRUDEN, D.M. (2005). Probabilistic assessment of stability of a cut slope in residual soil. *Géotechnique*, 55(1): 77-84.

ENTRALGO, J.F.T. (2013). Análise espacial e estatística dos movimentos de massa deflagrados pelas chuvas dos dias 11 e 12 de janeiro de 2011 na região serrana do estado do Rio de Janeiro, Brasil. Dissertação de Mestrado, Universidade de Brasília, Brasília, DF, 116 p.

FARIAS, M.M. & ASSIS, A.P. (1998). Uma comparação entre métodos probabilísticos aplicados à estabilidade de taludes. XI Congresso Brasileiro de Mecânica dos Solos e Engenharia Geotécnica, ABMS, Brasília, DF, 2: 1305-1313.

FELICÍSIMO, A., CUARTERO, A., REMONDO, J. & QUIRÓS, E. (2013). Mapping landslide susceptibility with logistic regression, multiple adaptive regression splines, classification and regression trees, and maximum entropy methods: a comparative study. *Landslides*, 10(2): 175-189.

FELL, R., COROMINAS, J., BONNARD, C., CASCINI, L., LEROI, E. & SAVAGE, W.Z. (2008a). Guidelines for landslide susceptibility, hazard and risk zoning for land use planning. *Engineering Geology*, 102(1): 85-98.

FELL, R., COROMINAS, J., BONNARD, C., CASCINI, L., LEROI, E. & SAVAGE, W.Z. (2008b). Commentary of guidelines for landslide susceptibility, hazard and risk zoning for land use planning. *Engineering Geology*, 102(1): 99-111.

FELL, R. & HARTFORD, D. (1997). Landslide risk management. *Landslide risk management*, D.M. Cruden & R. Fell (eds.), A.A. Balkema Publishers, Rotterdam, Netherlands, pp. 51-109.

FELL, R., HO, K.K.S., LACASSE, S. & LEROI, E. (2005). A framework for landslide risk assessment and management. *Landslide risk management, Proceedings of the International Conference on Landslide Risk Management*, O. Hungr, R. Fell, R. Couture & E. Eberhardt (eds.), Taylor and Francis, Londres, Inglaterra, pp. 3-25.

FERNANDES, N.F. & AMARAL, C.P. (1996). Movimentos de massa: uma abordagem geológica-geomorfológica. *Geomorfologia e Meio Ambiente*, A.J.T. Guerra & S.B. Cunha (eds.), Bertrand, Rio de Janeiro, RJ, pp. 123-194.

FERNANDES, N.F., GUIMARÃES, R.F., GOMES, R.A.T., VIEIRA, B.C., MONTGOMERY, D.R. & GREENBERG, H. (2004). Topographic controls of landslides in Rio de Janeiro: field evidence and modeling. *Catena*, 55(1): 163-181.

FLORES, E.A.F. (2008). Análises Probabilísticas da Estabilidade de Taludes Considerando a Variabilidade Espacial do Solo. Dissertação de Mestrado, PUC, Rio de Janeiro, RJ, 178 p.

FRANCÉS, F., VÉLEZ, J.I. & VÉLEZ, J.J. (2007). Split-parameter structure for the automatic calibration of distributed hydrological models. *Journal of Hydrology*, 332(1): 226-240.

FREDLUND, D.G., XING, A., FREDLUND, M.D. & BARBOUR, S.L. (1996). The relationship of the unsaturated soil shear strength to the soil-water characteristic curve. *Canadian Geotechnical Journal*, 33(3): 440-448.

FREEMAN, T.G. (1991). Calculating catchment area with divergent flow based on a regular grid. *Computer and Geosciences*, 17(3): 413-422.

GARCÍA, H.A.G. (2004). Avaliação da susceptibilidade aos movimentos de massa usando Sistemas de Informação Geográfica e Redes Neurais Artificiais. Dissertação de Mestrado, Universidade de Brasília, Brasília, Distrito Federal, 140 p.

GE, H., TU, J. & QIN, F. (2011). Analysis of Slope Stability with First Order Second Moment Method. *International Journal of Digital Content Technology and its Applications*, 5(12): 445-451.

GERENTE, J., LUIZ, E.L. & DAL SANTO, M.A. (2014) Utilização do SHALSTAB como ferramenta para o diagnóstico preliminar da suscetibilidade a deslizamentos rasos no baixo vale do rio Luís Alves – SC. *Ambiência Guarapuava*, 10(1): 333-350.

GIANNECCHINI, R., GALANTI, Y. & D'AMATO AVANZI, G. (2012). Critical rainfall thresholds for triggering shallow landslides in the Serchio River Valley (Tuscany, Italy). *Nat. Hazards Earth Syst. Sci.*, 12(1): 829-842.

GITIRANA, J.F.N. (2005). Weather-Related Geo-Hazard Assessment Model for Railway Embankment Stability. Ph.D. Thesis, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, 411 p.

GLADE, T. (2003). Landslide occurrence as a response to land use change: a review of evidence from New Zealand. *CATENA*, 51(3-4): 297-314.

GODT, J.W., BAUM, R.L. & CHLEBORAD, A.F. (2006). Rainfall characteristics for shallow landsliding in Seattle, Washington, USA. *Earth Surface Processes and Landforms*, 3(1): 97-110.

GODT, J.W., BAUM, R.L., SAVAGE, W.Z., SALCIARINI, D., SCHULZ W.H. & HARP, E.L. (2008). Transient deterministic shallow landslide modeling: Requirements for susceptibility and hazard assessments in a GIS framework. *Engineering Geology*, 102(1): 214-226.

GOSTELOW, T.P. (1991). Rainfall and landslides. Prevention and control of landslides and other mass movements, M.E. Almeida-Teixeira, R. Fantechi, R. Oliveira & A. Gomes Coelho (eds.), European Communities, Bruxelas, Bélgica, pp. 139-161.

GRAHAM, J. (1984). Methods of Stability Analysis. Slope Instability, D. Brunsten & D.B. Prior (eds.), John Wiley & Sons Ltd., Chichester, United Kingdom, pp. 171-215.

GRIFFITHS, D.V., FENTON, G.A. & TVETEN, D.E. (2002). Probabilistic geotechnical analysis: How difficult does it need to be? Proceedings of the International Conference on Probabilistics in Geotechnics: Technical and Economic Risk Estimation, R. Pottler, H. Klapperich & H. Schweiger (eds.), United Engineering Foundation, New York, USA, pp. 3-20.

GRIFFITHS, D.V., HUANG, J. & FENTON, G.A. (2011). Probabilistic infinite slope analysis. *Computers and Geotechnics*, 38(4): 577-584.

GRUBER, S. & PECKHAM, S. (2009). Land-Surface Parameters and Objects in Hydrology. *Geomorphometry: Concepts, Software, Applications*, T. Hengl & H. Reuter (eds.), Elsevier, Amsterdam, The Netherlands, pp. 171-194.

GUIMARÃES, R.F., GOMES, R.A.T., CARVALHO JÚNIOR, O.A., MARTINS, E.S., OLIVEIRA, S.N. & FERNANDES, N.F. (2009). Análise temporal das áreas susceptíveis a escorregamentos rasos no Parque Nacional da Serra dos Órgãos (RJ) a partir de dados pluviométricos. *Revista Brasileira de Geociências*, 39(1): 190-198.

GUSTAVSSON, M., KOLSTRUP, E. & SEIJMONSBERGEN, A.C. (2006). A new symbol-and-GIS based detailed geomorphological mapping system: renewal of a scientific discipline for understanding landscape development. *Geomorphology*, 77(1-2): 90-111.

GUZZETTI, F. (2000). Landslide fatalities and evaluation of landslide risk in Italy. *Engineering Geology*, 58(1): 89-107.

GUZZETTI, F., CARDINALI, M., REICHENBACH, P. & CARRARA, A. (1999). Comparing landslide maps: A case study in the upper Tiber River Basin, central Italy. *Environmental Management*, 25(1): 247-363.

GUZZETTI, F., PERUCCACCI S., ROSSI, M. & STARK, C.P. (2008). The rainfall intensity-duration control of shallow landslides and debris flows: an update. *Landslides*, 5(1): 3-17.

HACHICH, W. (1998). Segurança das fundações e escavações. *Fundações: Teoria e prática*, W. Hachich, F.F. Falconi, J.L. Saes, R.G.Q. Frota, C.S. Carvalho & S. Niyama (eds.), Pini, São Paulo, SP, pp. 197-208.

HAMMERSLEY, J.M. & HANDSCOMB, D.C. (1964). Monte Carlo Methods, Methuen's Monographs on Applied Probability and Statistics. Methuen, London, UK, 178 p.

HAMMOND, C., HALL, D., MILLER, S. & SWETIK, P. (1992). Level I Stability Analysis (LISA) Documentation for Version 2.0. General Technical Report INT-285, U.S. Department of Agriculture, Ogden, USA, 190 p.

HARR, M.E. (1987). Reliability-Based Design in Civil Engineering. McGraw-Hill Book Company, New York, EUA, 290 p.

HASSAN, A.M. & WOLFF, T.R. (2000). Effect of deterministic and probabilistic models on slope reliability index. *Slope Stability 2000*, ASCE, Denver, Colorado, 1: 194-208.

HERATH, S. & WANG, Y. (2009). Case Studies and National Experiences. *Landslides - Disaster Risk Reduction*. K. Sassa & P. Canuti (eds.), Springer, Berlin, Germany, pp. 475-497.

HOLMGREN, P. (1994). Multiple flow direction algorithms for runoff modeling in grid based elevation models - an empirical-evaluation. *Hydrological Processes*, 8(4): 327-334.

HUNGR, O., LEROUEIL, S. & PICARELLI, L. (2014). The Varnes classification of landslide types, an update. *Landslide*, 11(2): 167-194.

HUTCHINSON, J.N. (1986). A sliding-consolidation model for flow slides. *Can. Geotech. J.*, 23(2): 115-126.

HUTCHINSON, J.N. (1988). Morphological and Geotechnical Parameters of Landslides in Relation to Geology and Hydrogeology. 5th International Conference on Landslides, C. Bonnard (ed.), Balkema, Rotterdam, Netherlands, pp. 3-35.

IAEG (1990). Suggested Nomenclature for Landslides. Bulletin N° 41, International Association of Engineering Geology, Paris, França, 4 p.

IGOS (2004). Geohazards. Theme Report, IGOS, Paris, França, 55 p.

IIDA, T. (1999). A stochastic hydro-geomorphological model for shallow landsliding due to rainstorm. *Catena*, 34(1): 293-313.

IIRITANO, G., VERSACE, P. & SIRANGELO, B. (1998). Real-time estimation of hazard for landslides triggered by rainfall. *Environmental Geology*, 35(2-3): 175-183.

IVERSON, R.M. (2000). Landslide triggering by rain infiltration. *Water Resources Research*, 36(7): 1897-1910.

JAMES, F. (1980). Monte Carlo theory and practice. *Rep. Prog. Phys.*, 43(1): 1147-1189.

JORDAN, C.A. (2011). Combined hydrology and slope stability assessment of the olympic region of Washington State. Thesis submitted for a master's degree, Washington State University, Washington D.C., USA, 83 p.

JULIÃO, P.R., NERY, F., RIBEIRO, J.L., BRANCO, M.C. & ZÊZERE, J.L. (2009). Guia metodológico para a produção de cartografia municipal de risco e para a criação de sistemas de informação geográfica (SIG) de base municipal. Autoridade Nacional de Proteção Civil, Portugal, 92 p.

KARAM, K.S. (2005). Landslide Hazards Assessment and Uncertainties. Ph.D. Thesis, Massachusetts Institute of Technology, Boston, USA, 751 p.

KORVER, B. (1994). The Monte Carlo Method and Software Reliability Theory. Technical Report 94-1, Portland State University Computer Science Department, Portland, Oregon, 27 p.

KRYNINE, D.P. & JUDD, W.R. (1957). Principles of Engineering Geology and Geotechnics. McGraw-Hill Book Company, Tokio, Japan, 699 p.

KUBOTA, J. & SIVAPALAN, M. (1995). Towards a catchment-scale model of subsurface runoff generation based on synthesis of small-scale process-based modeling and field studies. *Hydrological Processes*, 9(1): 541-554.

LACASSE, S. & NADIM, F. (1996). Uncertainties in Characterizing Soil Properties. Uncertainty in the Geologic Environment: from theory to practice, C.D. Shackelford, P.P. Nelson & M.J.S. Roth (eds.), ASCE, New York, USA, pp. 49-75.

- LAI, T. & DRAGIĆEVIĆ, S. (2011). Development of an urban landslide cellular automata model: a case study of North Vancouver, Canada. *Earth Sci. Inform*, 6(1): 69-80.
- LEONE, F., ASTÉ, J.P. & LEROI, E. (1996). Vulnerability assessment of elements exposed to mass-movements: working toward a better risk perception. *Landslides - Glissements de Terrain*, K. Senneset (ed.), Balkema, Rotterdam, The Netherland, pp. 263-270.
- LEROUEIL, S., LOCAT, J., VAUNAT, J., PICARELLI, L., LEE, H. & FAURE, R. (1996). Geotechnical characterization of slope movements. *Landslides: Proceedings of the 7th International Symposium on Landslides*, K. Senneset (ed.), Balkema, Rotterdam, Netherlands, pp. 53-74.
- LI, A.G., YUE, L.G., THAM, L.G., LEE, C.F. & LAW, K.T. (2005). Field-monitored variations of soil moisture and matric suction in a saprolite slope. *Canadian Geotechnical Journal*, 42(1): 13-26.
- LI, Z., NADIM, F., HUANG, H., UZIELLI, M. & LACASSE, S. (2010). Quantitative vulnerability estimation for scenario-based landslide hazards. *Landslides*, 7(1): 125-134.
- LI, X., ZHANG, L.M. & FREDLUND, D.G. (2009). Wetting front advancing column test for measuring unsaturated hydraulic conductivity. *Canadian Geotechnical Journal*, 46(12): 1431-1445.
- LIAO, Z., HONG, Y., WANG, J., FUKUOKA, H., SASSA, K. KARNAWATI, D. & FATHANI, F. (2010). Prototyping an experimental early warning system for rainfall-induced landslides in Indonesia using satellite remote sensing and geospatial datasets. *Landslides*, 7(1): 317-324.
- LÔBO, B.R.H. (2012). Efeito do avanço da frente de umidade na redução da sucção e da resistência ao cisalhamento de um perfil de solo coluvionar em Medellín-Colômbia. *Dissertação de Mestrado, Universidade de Brasília, Brasília, DF, 113 p.*
- MAIA, J.A.C. (2003). Métodos Probabilísticos Aplicados à Estabilidade de Taludes e Cavidades em Rochas. *Dissertação de Mestrado, Universidade de Brasília, Brasília, DF, 228 p.*
- MALAMUD, B.D., TURCOTTE, D.L., GUZZETTI, F. & REICHENBACH, P. (2004). Landslide inventories and their statistical properties. *Earth Surface Processes and Landforms*, 29(1): 687-711.
- MARCELINO, E.V. (2007). *Desastres naturais e geoteconologias: conceitos básicos*. INPE, Santa Maria, RS, 20 p.
- MARINILLI, A. & CERROLAZA, M. (1999). Computational stochastic analysis of earth structure settlements. *Computers and Geotechnics*, 25(1): 107-121.
- MATHEW, J., GIRI BABU, D., KUNDU, S., VINOD KUMAR, K. & PANT, C.C. (2014). Integrating intensity-duration-based rainfall threshold and antecedent rainfall-based probability estimate towards generating early warning for rainfall-induced landslides in parts of the Garhwal Himalaya, India. *Landslides*, 11(4): 575-588.

- MATLAN, S.J. (2009). Back analysis of rainfall induced landslide in Sabah by Perisi Model. Thesis for the award of the degree of Master of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 93 p.
- METTERNICHT, G., HURNI, L. & GOGU, R. (2005). Remote sensing of landslides: an analysis of the potential contribution to geo-spatial systems for hazard assessment in mountainous environments. *Remote Sensing of Environment*, 98(23): 284-303.
- MILIAREISIS, G.C. (2001). Geomorphometric mapping of Zagros Ranges at regional scale. *Computers & Geosciences*, 27(1): 775-786.
- MILLER, S.M., WHYATT, J.K. & MCHUGH, E.L. (2004). Applications of the point estimation method for stochastic rock slope engineering. 6th North American Rock Mechanics Conference, ARMA, Houston, Texas, 1: 1-12.
- MIRANDA, M.F. (2005). Avaliação da estabilidade de contenções através de métodos probabilísticos. Dissertação de mestrado, Universidade Estadual do Norte Fluminense Darcy Ribeiro, Campos dos Goytacazes, RJ, 175 p.
- MONTGOMERY, D.R. & DIETRICH, W.E. (1994). A physically based model for the topographic control on shallow landsliding. *Water Resources Research*, 30(4): 1153-1171.
- MONTOYA, C.A.H. (2013). Incertezas, vulnerabilidade e avaliação de risco devido a deslizamento em estradas. Tese de Doutorado, Universidade de Brasília, Brasília, DF, 273 p.
- MONTOYA, C.A.H. & ASSIS, A.P. (2011a). Tools for reliability analysis in geotechnical engineering: theory (in Spanish). *Revista Ingenierías Universidad de Medellín*, 10(18): 69-78.
- MONTOYA, C.A.H. & ASSIS, A.P. (2011b). Tools for reliability analysis in geotechnical engineering: application (in Spanish). *Revista Ingenierías Universidad de Medellín*, 10(18): 79-86.
- MONTRASIO, L. & VALENTINO, R. (2008). A model for triggering mechanisms of shallow landslides. *Natural Hazards and Earth System Sciences*, 8(1): 1149-1159.
- MOORE, J. & WEATHERFORD, L.R. (2005). Tomada de decisão em administração com planilhas eletrônicas. Bookman, Porto Alegre, Rio Grande do Sul, 643 p.
- MORGENSTERN, N.R. (1995). Managing Risk in Geotechnical Engineering. Proc. 10th Panamerican Conf. on Soil Mechanics and Foundation Engineering, SMMS, Guadalajara, México, 4: 2-126.
- MOTAMEDI, M. & LIANG, R.Y. (2014). Probabilistic landslide hazard assessment using Copula modeling technique. *Landslide*, 11(4): 565-573.
- OLIVARES, L., DAMIANO, E., MERCOGLIANO, P., PICARELLI, L., NETTI, N., SCHIANO, P., SAVASTANO, V., COTRONEO, F. & MANZI, M.P. (2014). A simulation chain for early prediction of rainfall-induced Landslides. *Landslides*, 11(5): 765-777.

O'LOUGHLIN, E. M. (1986). Prediction of surface saturation zones in natural catchments by topographic analysis. *Water Resources Research*, 22(1): 794-804.

OTÁLVARO, I.F. & CORDÃO NETO, M.P. (2011). Avaliação da estabilidade em maciços de solo não saturado: um enfoque probabilístico. VII Simpósio Brasileiro de Solos Não Saturados, ABMS, Pirenópolis, Goiás, 1: 215-220.

PACK, R.T., TARBOTON, D.G. & GOODWIN, C.N. (1998). The SINMAP Approach to Terrain Stability Mapping. 8th Congress of the International Association of Engineering Geology, IAEG, Vancouver, Canada, 1: 21-25.

PAPA, M.N., MEDINA, V. & BATEMAN, A. (2011). Derivation of critical rainfall thresholds for debris flow warnings through mathematical and numerical modeling. Fifth International Conference on Debris-flow Hazards Mitigation: Mechanics, Prediction and Assessment, R. Genevois, D.L. Hamilton & A. Prestininzi (eds.), Università La Sapienza, Roma, Itália, pp. 495-502.

PEREIRA, T.S. (2013). Avaliação do desempenho de diferentes métodos de análise de análise de estabilidade de taludes em barragens de terra. Dissertação de mestrado, Universidade Federal de Santa Maria, Santa Maria, RS, 78 p.

PHOON, K-K. & KULHAWY, F.H. (1999). Characterization of geotechnical variability. *Canadian Geotechnical Journal*, 36(4): 612-624.

PIKE, R.J. (2000). Geomorphometry - diversity in quantitative surface analysis. *Progress in Physical Geography*, 24(1): 1-20.

QIAO, S.F. & CLAYTON, C.R.I. (2013). Flow slides run-out prediction using a sliding-consolidation model. *Landslides*, 10(1): 831-842.

QUINN, P., BEVEN, K., CHEVALLIER, P. & PLANCHON, O. (1991). The prediction of hillslope paths for distributed hydrological modeling using digital terrain models. *Hydrological Processes*, 5(1): 59-79.

RADBRUCH-HALL, D.H., COLTON, R.B., DAVIES, W.E., LUCCHITTA, I., SKIPP, B.A. & VARNES, D.J. (1982). Landslide overview map of the conterminous United States. US Geological Survey Professional Paper 1183, United States Government Printing Office, Washington D.C, USA, 25 p.

RAGOZIN, A.L. & TIKHVINSKY, I.O. (2000). Landslide hazard, vulnerability and risk assessment. *Landslides in Research, Theory and Practice*, E. Bromhead, N. Dixon & M.L. Ibsen (eds.), Thomas Telford, Cardiff, País de Gales, pp. 1257-1262.

RAHARDJO, H., LEE, T., LEONG, E. & REZAUR, R. (2005). Response of a residual soil slope to rainfall. *Canadian Geotechnical Journal*, 42(2): 340-351.

RAHARDJO, H., LI, X.W., TOLL, D.G. & LEONG, E.C. (2001). The effect of antecedent rainfall on slope stability. *Geotechnical and Geological Engineering*, 19(1): 371-399.

RAHARDJO, H., ONG, T.H., REZAUR, R.B. & LEONG, E.C. (2007). Factors Controlling Instability of Homogeneous Soil Slopes under Rainfall. *Journal of Geotechnical and Geoenvironmental Engineering*, 133(12): 1532-1543.

RAMOS, J.A.S., SILVEIRA, C.S. & ROIG, H.L. (2007). Desenvolvimento de um algoritmo para a determinação do fluxo e da hierarquia de canais de drenagem. XIII Simpósio Brasileiro de Sensoriamento Remoto, INPE, Florianópolis, Santa Catarina, 1: 3027-3034.

RAMOS, V.M., GUIMARÃES, R.F., REDIVO, A.L., GOMES, R.A.T., FERNANDES, N.F. & CARVALHO JÚNIOR, O.A. (2002). Aplicação do modelo SHALSTAB, em ambiente Arcview, para o mapeamento de áreas susceptíveis a escorregamento raso na região do quadrilátero ferrífero (MG). *Espaço & Geografia*, 5(1): 49-67.

REATTO, A., MARTINS, E.S., FARIAS, M.F.R., SILVA, A.V. & CARVALHO JR., O.A.C. (2004). Mapa Pedológico Digital – SIG Atualizado do Distrito Federal Escala 1:100.000 e uma Síntese do Texto Explicativo. Embrapa, Planaltina, DF, 29 p.

REGINATTO, G.M.P., MACCARINI, M., KOBAYAMA, M., HIGASHI, R.A.R., GRANDO, A., CORSEUIL, C.W. & CARAMÉZ, M.L. (2012). Shalstab Application to Identify the Susceptible Areas of Shallow Landslides in Cunha River Watershed, Rio dos Cedros City, SC, Brazil. Fourth International Conference on Geographic Object-Based Image Analysis (GEOBIA), INPE, Rio de Janeiro, RJ, 1: 108-113.

REIS, R.C. (2010). Estudo de estabilidade de taludes da mina de Tapira-MG. Dissertação de Mestrado, Programa de Pós-Graduação em Geotecnia, Universidade Federal de Ouro Preto, Ouro Preto, MG, 145 p.

REIS E., ZÉZERE J.L., VIEIRA G.T. & RODRIGUES M.L. (2003). Integração de Dados Espaciais em SIG Para Avaliação da Susceptibilidade à Ocorrência de Deslizamentos. *Finisterra*, 38(76): 3-34.

RESTREPO, P.A.I. (2011). Avaliação do Risco Aceitável por Escorregamentos de Encostas em Regiões Tropicais Montanhosas: Caso de Aplicação, Medellín-Colômbia. Dissertação de Mestrado, Universidade de Brasília, Brasília, DF, 204 p.

RIBEIRO JUNIOR, C.A. (2011). Análise Numérica de Escorregamento em Encostas. Dissertação de Mestrado, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, RJ, 130 p.

RILHO, A.S. (2011). Aplicação de métodos probabilísticos e análise da decisão em projetos de fundações profundas. Trabalho de diplomação, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 76 p.

RÍOS, M.A. & CARVAJAL, H.E.M. (2011). Análise Probabilística de Estabilidade de Taludes em Escalas Sub-regionais baseado no Método de Confiabilidade de Primeira Ordem (FORM). VI Simpósio Brasileiro de Informática em Geotecnia, ABMS, Brasília, DF, 1: 81-87.

RODRÍGUEZ, A.J.M. (2010). Stochastic modeling of geotechnical engineering problems (in Spanish). Tese de Doutorado, Universidad Central de Las Villas, Santa Clara, Cuba, 169 p.

ROQUE, W.V. & SOUZA, N.M. (2008). Análise da Estabilidade de Taludes pelo Método Racional na Região do Taquari (DF). 12º Congresso Brasileiro de Geologia de Engenharia e Ambiental, ABGE, Porto de Galinhas, Pernambuco, 1: 1-9.

ROSENBLUETH, E. (1975). Point estimates for probability moments. Proceedings of the National Academy of Sciences of the United of America, 72(10): 3812-3814.

ROSENBLUETH, E. (1981). Two-point estimates in probabilities. Appl. Math. Modelling, 5(2): 329-335.

SALCIARINI, D., GODT, J.W., SAVAGE, W.Z., BAUM R.L. & CONVERSINI, P. (2008). Modeling landslide recurrence in Seattle, Washington, USA. Engineering Geology, 102(1): 227-237.

SALDIVAR-SALI, A. & EINSTEIN, H.H. (2007). A Landslide Risk Rating System for Baguio, Philippines. Engineering Geology, 91(1): 85-99.

SANDRONI, S.S. & SAYÃO, A.S.F. (1993). The use of relative probability of failure in the design of open pit mine slopes. Innovative Mine Design for the 21st Century, W.F. Bawden & J.F. Archibald (eds.), Balkema, Rotterdam, The Netherlands, pp. 21-24.

SANTOS, M.S. (2008). Inferência Bayesiana na avaliação da segurança de fundações em estacas de deslocamento. Dissertação de Mestrado, Universidade de São Paulo, São Paulo, SP, 128 p.

SCHUSTER, R.L. (1996). Socioeconomic significance of landslides. Landslides: Investigation and Mitigation, A.K. Turner & R.L. Schuster (eds.), National Academy Press, Washington, USA, pp. 12-35.

SERNA, M.A.L., FARIAS, M.M. & CARVAJAL, H.E.M. (2015). Numerical modeling of Alto Verde landslide using the Material Point Method (in Spanish). Dyna, (In Press), 1-11.

SHARMA, S. (1996). Slope Stability Concepts. Slope Stability and Stabilization Methods, L.W. Abramson, T.S. Lee, S. Sharman & G.M. Boyce (eds.), John Wiley & Sons Inc., New York, USA, pp. 337-440.

SIDLER, R.C. & OCHIAI, H. (2006). Landslides - processes, prediction and land use. American Geophysical Union, Washington, USA, 312 p.

SILVA, B.A. (2005). Análise de estabilidade de taludes ao longo da rota de um duto submarino na Bacia de Campos, RJ. Dissertação de Mestrado, PUC, Rio de Janeiro, RJ, 117 p.

SILVA, C.P.L., ALENCAR, C.L.S. & SOUZA, N.M. (2007). Utilização de geoprocessamento e cartografia geotécnica em estudos de susceptibilidade a escorregamentos na vila Varjão (DF). Proc. 6º Simpósio Brasileiro de Cartografia Geotécnica e Geoambiental, ABGE, Uberlândia, Minas Gerais, 1: 169-183.

SILVA, D.C.O. (2006). Aplicação do Modelo Shalstab na Previsão de Deslizamentos em Petrópolis. Dissertação de Mestrado, COPPE, UFRJ, Rio de Janeiro, RJ, 132 p.

SILVA, E.M. (2006). Análise de Estabilidade de Taludes em Solos de Alteração de Rochas Metamórficas do Quadrilátero Ferrífero. Dissertação de Mestrado, Universidade Federal de Viçosa, Viçosa, Minas Gerais, 143 p.

SILVEIRA, C.S. & RAMOS, J.A.S. (2007). Análise espacial com SIG de parâmetros ambientais e comportamento hidrológico (chuva-vazão) de uma bacia de drenagem montanhosa na serra dos órgãos: bacia do Paquequer, município de Teresópolis, RJ. *Revista Brasileira de Geomorfologia*, 8(1): 23-34.

SINGH, G. & CHUNG, C.M. (1991). Microcomputer-based reliability analysis in geotechnics. *Computers & Structures*, 41(6): 1397-1402.

SKEMPTON, A.W. & HUTCHINSON, J.N. (1969). Stability of natural slopes and embankment foundations. 7th. International conference of soil mechanics and foundation engineering, Mexico City, Mexico, State of the Art volume: 291-340.

SOBOL, L.M.A. (1994). *Primer for the Monte Carlo Method*. CRC Press Florida, Boca Raton, Florida, 128 p.

SOETERS, R. & VAN WESTEN, C.J. (1996). Slope instability recognition, analysis and zonation. *Landslides: Investigation and Mitigation*, A.K. Turner & R.L. Schuster (eds.), National Academic Press, Washington D.C., USA, pp. 129-177.

SOUZA, C.R.G, FARIA, D.G.M., SANTORO, J., TOMINAGA, L.K., TAVARES, R., MENDES, R.M., RIBEIRO, R.R., AMARAL, R. & SALLUN FILHO, W. (2009). *Desastres Naturais: conhecer para prevenir*. Instituto Geológico, São Paulo, SP, 196 p.

SOUZA, F.T. (2004). Predição de escorregamentos das encostas do município do Rio de Janeiro através de técnicas de mineração de dados. Tese de Doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 115 p.

SOUZA, N.M. (2004). *Cartografia Geotécnica*. Apostila, Universidade de Brasília, Brasília, DF, 40 p.

SPENCE, R.J.S., KELMAN, I., CALOGERO, E., TOYOS, G., BAXTER, P.J. & KOMOROWSKI, J.C. (2005). Modelling expected physical impacts and human casualties from explosive volcanic eruptions. *Natural Hazards and Earth System Sciences*, 5(1):1003-1015.

STERLACCHINI, S., FRIGERIO, S., GIACOMELLI, P. & BRAMBILLA, M. (2007). Landslide risk analysis: A multi-disciplinary methodological approach. *Natural Hazards Earth System Science*, 7(1): 657-675.

SUCHOMEL, R. & MAŠIN, D. (2009). Comparison of different probabilistic methods for predicting stability of a slope in spatially variable $c-\phi$ soil. *Computers and Geotechnics*, 37(1): 132-140.

- TAN, C.H., KU, C.Y., CHI, S.Y., CHEN, Y.H., FEI, L.Y., LEE, J.F. & SU, T.W. (2008). Assessment of regional rainfall-induced landslides using 3S-based hydro-geological model. 10th International Symposium on Landslides and Engineered Slopes, CISMGE-CCES, Xi'an, China, 1: 1639- 1645.
- TANG, W.H. (1993). Recent developments in geotechnical reliability. Probabilistic methods in geotechnical engineering, K.S. Li & S-C.R. Lo (eds.), A.A. Balkema, Rotterdam, Holanda, pp. 3-28.
- TATIZANA, C., OGURA, A.T., CERRI, L.E.S. & ROCHA, M.C.M. (1987). Análise de Correlação entre Chuvas e Deslizamentos - Serra do Mar, município de Cubatão. 5º Congresso Brasileiro de Geologia de Engenharia, ABGE, São Paulo, SP, 2: 225-236.
- TERLIEN, M.T.J. (1998). The determination of statistical and deterministic hydrological landslide - triggering thresholds. *Environmental Geology*, 35(2-3): 124-130.
- TERZAGHI, K. (1936). The shearing resistance of saturated soils and the angle between the planes of shear. Proc. 1st Int. Conf. Soil Mech. and Found. Engg., Harvard, Cambridge, Massachusetts, 1: 54-56.
- TERZAGHI, K. (1950). Mechanics of landslides. Application of Geology to Engineering Practice, Geological Society of America (ed.), Geological Society of America, New York, EUA, pp. 83-124.
- THANH, L.N. & DE SMEDT, F. (2014). Slope stability analysis using a physically based model: a case study from A Luoi district in Thua Thien-Hue Province, Vietnam. *Landslides*, 11(5): 897-907.
- THIEBES, B., BELL, R., GLADE, T., JÄGER, S., MAYER, J., ANDERSON, M. & HOLCOMBE, L. (2014). Integration of a limit-equilibrium model into a landslide early warning system. *Landslides*, 11(5): 859-875.
- TOLL, D.G., TSAPARAS, I. & RAHARDJO, H. (2001). The influence of rainfall sequences on negative pore-water pressures within slopes. 15th International Conference on Soil Mechanics and Geotechnical Engineering, ICSMGE, Istanbul, Turkey, 2: 1269-1272.
- TOMINAGA, L.K. (2007). Avaliação de Metodologias de Análise de Risco a Escorregamentos: Aplicação de um Ensaio em Ubatuba, SP. Tese de Doutorado, Universidade de São Paulo, São Paulo, SP, 220 p.
- TRIBE, S. & LEIR, M. (2004). The role of aerial photograph interpretation in natural hazard and risk assessment. Proceedings of the International Pipeline Conference, PRCI, Calgary, Canada, 1: 1-6.
- TURNER, A.K. (1996) Socioeconomic significance of landslides. *Landslides: Investigation and Mitigation*, A.K. Turner & R.L. Schuster (eds.), National Academy Press, Washington D.C., USA, pp. 12-35.

- UZIELLI, M., NADIM, F., LACASSE, S. & KAYNIA, A.M. (2008). A conceptual framework for quantitative estimation of physical vulnerability to landslides. *Engineering Geology*, 102(1): 251-256.
- VALLEY, B. & DUFF, D. (2011). Probabilistic analyses in Phase2 8.0. RocNews Articles, Center for Excellence in Mining Innovation, Sudbury, Canada, 6 p.
- VAN BEEK, L.P.H. & VAN ASCH, T.W.J. (2004). Regional assessment of the effects of land-use change and landslide hazard by means of physically based modeling. *Natural Hazards*, 30(3): 289-304.
- VANMARCKE, E.H. (1977). Probabilistic modeling of soil profiles. *Journal of the Geotechnical Engineering Division*, 103(11): 1227-1246.
- VAN WESTEN, C.J., ASCH, T.W.J. & SOETERS, R. (2005). Landslide hazard and risk zonation – why is it still so difficult? *Bulletin of Engineering Geology and the Environment* 65(2): 167-184.
- VAN WESTEN, C.J., CASTELLANOS, E. & KURIAKOSE, S.L. (2008). Spatial data for landslide susceptibility, hazard, and vulnerability assessment: An overview. *Engineering Geology*, 102(1): 112-131.
- VARNES, D.J. (1978). Slope movement types and processes. *Landslides, Analysis and Control*, R. L. Schuster & R. J. Krizek (eds.), Transportation Research Board – Commission on Sociotechnical Systems - National Research Council, National Academy of Sciences, Washington, USA, pp. 11-33.
- VÉLEZ, J.I. (2001). Desarrollo de un modelo hidrológico conceptual distribuido orientado a la simulación de crecidas. Tesis Doctoral, Universidad Politécnica de Valencia, Valencia, España, 266 p.
- WANG, G. & SASSA, K. (2003). Pore-pressure Generation and Movement of Rainfall-induced landslides: Effects of Grain Size and Fine-particle Content. *Engineering Geology*, 69(1): 109-125.
- WANG, H.B. & SASSA, K. (2006). Rainfall-induced Landslide Hazard Assessment Using Artificial Neural Networks. *Earth Surface Processes and Landforms*, 31(2): 235-247.
- WHITMAN, R.V. (1984). Evaluating calculated risk in geotechnical engineering. *J. of Geot. Eng.*, 110(2): 143-188.
- WHITMAN, V.W. (1996). Organizing and Evaluating Uncertainty in Geotechnical Engineering. *Uncertainty in the Geologic Environment: from theory to practice*, C.D. Shackelford, P.P. Nelson & M.J.S. Roth (eds.), ASCE, New York, USA, pp. 1-28.
- WIECZOREK, G.F. (1984). Preparing a detailed landslide-inventory map for hazard evaluation and reduction. *Bulletin Association of Engineering Geologists*, 21(1): 337-342.

WIECZOREK, G.F. (1987). Effect of rainfall intensity and duration on debris flows in central Santa Cruz Mountains, California. *Debris Flows/Avalanches: Processes, Recognition and Mitigation*, J.E. Costa & G. Wieczorek (eds.), The Geological Society of America, Boulder, Colorado, pp. 93-104.

WIECZOREK, G.F. & GLADE, T. (2005). Climatic Factors Influencing Occurrence of Debris Flows. *Debris-flow Hazards and Related Phenomena*, M. Jakob & O. Hungr (eds.), Springer, Chichester, UK, pp. 325-362.

WIECZOREK, G.F., MORGAN, B.A. & CAMPBELL, R.H. (2000). Debris-flow hazards in the Blue Ridge of central Virginia. *Environmental and Engineering Geoscience*, 6(1): 3-23.

WU, T.H., TANG, W.H. & EINSTEIN, H.H. (1996). Landslide hazard and risk assessment. *Landslides: Investigation and Mitigation*, A.K. Turner & R.L. Schuster (eds.), National Academy Press, Washington, USA, pp. 106-118.

YALCIN, A. (2011). A geotechnical study on the landslides in the Trabzon Province, NE, Turkey. *Applied Clay Science*, 52(1): 11-19.

YIN, Y. (2009). Landslide mitigation strategy and implementation in China. *Landslides - disaster risk reduction*, K. Sassa & P. Canuti (eds.), Springer, Berlin, Germany, pp. 482-484.

YUAN, C.C., CHIEN, C.T., CHIEH, Y.F. & CHI, L.S. (2005). Analysis of time-varying rainfall infiltration induced landslide. *Environmental Geology*, 48(1): 466-479.

ZAIDAN, R.T. & FERNANDES, F.F. (2009). Modelagem de Susceptibilidade a Escorregamentos na Bacia de Drenagem Urbana do Córrego do Independência: Juiz de fora - MG. *Revista Brasileira de Geomorfologia*, 10(1): 57-76.

ZHANG, J., HUANG, H.W., ZHANG, L.M., ZHU, H.H. & SHI, B. (2014). Probabilistic prediction of rainfall-induced slope failure using a mechanics-based model. *Engineering Geology*, 168(1): 129-140.

ZHANG, J., ZHANG, L.M. & TANG, W.H. (2011). Slope reliability analysis considering site-specific performance information. *Journal of geotechnical and geoenvironmental engineering*, 137(3): 227-238.

APÊNDICE A

LISTAGEM DAS ROTINAS COMPUTACIONAIS

A.1 FUNÇÕES PARA MODELO SHIA_LANSLIDE PROBABILÍSTICO

A1.1 INTERFACE GRÁFICA

```
function varargout = SHIA_COMPL2(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @SHIA_COMPL2_OpeningFcn, ...
                  'gui_OutputFcn',  @SHIA_COMPL2_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function SHIA_COMPL2_OpeningFcn(hObject, eventdata, handles, varargin)
global FS_crit;
global dado_CV1;
global dado_CV2;
global dado_CV3;

handles.output = hObject;

guidata(hObject, handles);
% inicialmente o fator de segurança crítico é 1
FS_crit = 1;
% inicialmente, os dados de CV são aqueles padrões
dado_CV1 = 40;
dado_CV2 = 10;
dado_CV3 = 3;

function varargout = SHIA_COMPL2_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function edit89_Callback(hObject, eventdata, handles)
global dado_col2;

dado_col2 = str2double(get(hObject, 'String'));

function edit89_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit90_Callback(hObject, eventdata, handles)
global dado_PtoCont2;

dado_PtoCont2 = str2double(get(hObject, 'String'));

function edit90_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit91_Callback(hObject, eventdata, handles)
global dado_PtoCont1;

dado_PtoCont1 = str2double(get(hObject, 'String'));

function edit91_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit92_Callback(hObject, eventdata, handles)
global dado_row2;

dado_row2 = str2double(get(hObject, 'String'));
```

```

function edit92_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupmenu3_Callback(hObject, eventdata, handles)
global dado_CV1;
global dado_CV2;
global dado_CV3;

contents_menu3 = get(hObject,'Value');

    if contents_menu3 == 1

        set(handles.edit86, 'enable', 'on','String','40');
        set(handles.edit87, 'enable', 'on','String','10');
        set(handles.edit88, 'enable', 'on','String','3');
        dado_CV1 = 40;
        dado_CV2 = 10;
        dado_CV3 = 3;

    elseif contents_menu3 == 2

        set(handles.edit86, 'enable', 'on','String','');
        set(handles.edit87, 'enable', 'on','String','');
        set(handles.edit88, 'enable', 'on','String','');
    end % end if

function popupmenu3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit86_Callback(hObject, eventdata, handles)
global dado_CV1;

dado_CV1 = str2double(get(hObject,'String'));

function edit86_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit87_Callback(hObject, eventdata, handles)
global dado_CV2;

dado_CV2 = str2double(get(hObject,'String'));

function edit87_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit88_Callback(hObject, eventdata, handles)
global dado_CV3;

dado_CV3 = str2double(get(hObject,'String'));

function edit88_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function pushbutton57_Callback(hObject, eventdata, handles)
global dado_Chuva;
[filename_Chuva pathname_Chuva] = uigetfile({'*.*'},'File Selector');
fullpathname_Chuva = strcat(pathname_Chuva,filename_Chuva);
set(handles.text343,'String',fullpathname_Chuva);

dado_Chuva = fullfile(pathname_Chuva,filename_Chuva);

%
% -----
% BOTÕES REFERENTES AOS MAPAS
% --- Executes on button press in pushbutton45.

```

```

function pushbutton45_Callback(hObject, eventdata, handles)
global dado_Dire;

[filename_Dire pathname_Dire] = uigetfile({'*.*'}, 'File Selector');
fullpathname_Dire = strcat(pathname_Dire, filename_Dire);
set(handles.text329, 'String', fullpathname_Dire);

dado_Dire = fullfile(pathname_Dire, filename_Dire);

% --- Executes on button press in pushbutton46.
function pushbutton46_Callback(hObject, eventdata, handles)
global dado_Dec;

[filename_Dec pathname_Dec] = uigetfile({'*.*'}, 'File Selector');
fullpathname_Dec = strcat(pathname_Dec, filename_Dec);
set(handles.text330, 'String', fullpathname_Dec);

dado_Dec = fullfile(pathname_Dec, filename_Dec);

% --- Executes on button press in pushbutton47.
function pushbutton47_Callback(hObject, eventdata, handles)
global dado_Area;

[filename_Area pathname_Area] = uigetfile({'*.*'}, 'File Selector');
fullpathname_Area = strcat(pathname_Area, filename_Area);
set(handles.text331, 'String', fullpathname_Area);

dado_Area = fullfile(pathname_Area, filename_Area);

% --- Executes on button press in pushbutton48.
function pushbutton48_Callback(hObject, eventdata, handles)
global dado_Ks;

[filename_Ks pathname_Ks] = uigetfile({'*.*'}, 'File Selector');
fullpathname_Ks = strcat(pathname_Ks, filename_Ks);
set(handles.text332, 'String', fullpathname_Ks);

dado_Ks = fullfile(pathname_Ks, filename_Ks);

% --- Executes on button press in pushbutton49.
function pushbutton49_Callback(hObject, eventdata, handles)
global dado_Kp;

[filename_Kp pathname_Kp] = uigetfile({'*.*'}, 'File Selector');
fullpathname_Kp = strcat(pathname_Kp, filename_Kp);
set(handles.text333, 'String', fullpathname_Kp);

dado_Kp = fullfile(pathname_Kp, filename_Kp);

% --- Executes on button press in pushbutton50.
function pushbutton50_Callback(hObject, eventdata, handles)
global dado_S1m;

[filename_S1m pathname_S1m] = uigetfile({'*.*'}, 'File Selector');
fullpathname_S1m = strcat(pathname_S1m, filename_S1m);
set(handles.text334, 'String', fullpathname_S1m);

dado_S1m = fullfile(pathname_S1m, filename_S1m);

% --- Executes on button press in pushbutton51.
function pushbutton51_Callback(hObject, eventdata, handles)
global dado_S3m;

[filename_S3m pathname_S3m] = uigetfile({'*.*'}, 'File Selector');
fullpathname_S3m = strcat(pathname_S3m, filename_S3m);
set(handles.text335, 'String', fullpathname_S3m);

dado_S3m = fullfile(pathname_S3m, filename_S3m);

% --- Executes on button press in pushbutton52.
function pushbutton52_Callback(hObject, eventdata, handles)
global dado_Zs;

[filename_Zs pathname_Zs] = uigetfile({'*.*'}, 'File Selector');
fullpathname_Zs = strcat(pathname_Zs, filename_Zs);
set(handles.text336, 'String', fullpathname_Zs);

dado_Zs = fullfile(pathname_Zs, filename_Zs);

% --- Executes on button press in pushbutton53.
function pushbutton53_Callback(hObject, eventdata, handles)
global dado_Y;

```

```

[filename_Y pathname_Y] = uigetfile({'*.*'}, 'File Selector');
fullpathname_Y = strcat(pathname_Y, filename_Y);
set(handles.text337, 'String', fullpathname_Y);

dado_Y = fullfile(pathname_Y, filename_Y);

% --- Executes on button press in pushbutton54.
function pushbutton54_Callback(hObject, eventdata, handles)
global dado_phi;

[filename_phi pathname_phi] = uigetfile({'*.*'}, 'File Selector');
fullpathname_phi = strcat(pathname_phi, filename_phi);
set(handles.text338, 'String', fullpathname_phi);

dado_phi = fullfile(pathname_phi, filename_phi);

% --- Executes on button press in pushbutton55.
function pushbutton55_Callback(hObject, eventdata, handles)
global dado_C;

[filename_C pathname_C] = uigetfile({'*.*'}, 'File Selector');
fullpathname_C = strcat(pathname_C, filename_C);
set(handles.text339, 'String', fullpathname_C);
dado_C = fullfile(pathname_C, filename_C);

% --- Executes on button press in pushbutton56.
function pushbutton56_Callback(hObject, eventdata, handles)
global dado_EVP;

[filename_EVP pathname_EVP] = uigetfile({'*.*'}, 'File Selector');
fullpathname_EVP = strcat(pathname_EVP, filename_EVP);
set(handles.text340, 'String', fullpathname_EVP);

dado_EVP = fullfile(pathname_EVP, filename_EVP);

function edit51_Callback(hObject, eventdata, handles)
global dado_iv2;

dado_iv2 = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit51_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit52_Callback(hObject, eventdata, handles)
global dado_iv3;

dado_iv3 = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit52_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit53_Callback(hObject, eventdata, handles)
global dado_iv4;

dado_iv4 = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit53_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit54_Callback(hObject, eventdata, handles)
global dado_iv5;

dado_iv5 = str2double(get(hObject, 'String'));

function edit54_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit55_Callback(hObject, eventdata, handles)
global dado_sil;

```

```

dado_si1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit55_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit56_Callback(hObject, eventdata, handles)
global dado_si2;

dado_si2 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit56_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit57_Callback(hObject, eventdata, handles)
global dado_si3;

dado_si3 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit57_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit58_Callback(hObject, eventdata, handles)
global dado_si4;

dado_si4 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit58_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit59_Callback(hObject, eventdata, handles)
global dado_si5;

dado_si5 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit59_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit37_Callback(hObject, eventdata, handles)
global dado_R1;

dado_R1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit37_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit38_Callback(hObject, eventdata, handles)
global dado_R2;

dado_R2 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit38_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit39_Callback(hObject, eventdata, handles)
global dado_R3;

dado_R3 = str2double(get(hObject,'String'));

```

```

% --- Executes during object creation, after setting all properties.
function edit39_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit40_Callback(hObject, eventdata, handles)
global dado_R4;

dado_R4 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit40_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit41_Callback(hObject, eventdata, handles)
global dado_R5;

dado_R5 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit41_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit42_Callback(hObject, eventdata, handles)
global dado_R13;

dado_R13 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit42_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit43_Callback(hObject, eventdata, handles)
global dado_R6;

dado_R6 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit43_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit44_Callback(hObject, eventdata, handles)
global dado_R7;

dado_R7 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit44_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit45_Callback(hObject, eventdata, handles)
global dado_R8;

dado_R8 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit45_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit46_Callback(hObject, eventdata, handles)
global dado_R9;

dado_R9 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit46_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit47_Callback(hObject, eventdata, handles)
global dado_R10;
dado_R10 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit47_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit48_Callback(hObject, eventdata, handles)
global dado_R11;

dado_R11 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit48_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit49_Callback(hObject, eventdata, handles)
global dado_R12;

dado_R12 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit49_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton17.
function pushbutton17_Callback(hObject, eventdata, handles)
% BOTÃO de SAÍDA
delete(get(hObject, 'parent'));

% --- Executes on button press in pushbutton18.
function pushbutton18_Callback(hObject, eventdata, handles)

global dado_Dire;
global dado_Dec;
global dado_Area;
global dado_Ks;
global dado_Kp;
global dado_S1m;
global dado_S3m;
global dado_Zs;
global dado_Y;
global dado_phi;
global dado_C;
global dado_EVP;

global dado_hillslope;
global dado_channel;
global dado_iv2;
global dado_iv3;
global dado_iv4;
global dado_iv5;
global dado_si1;
global dado_si2;
global dado_si3;
global dado_si4;
global dado_si5;
global dado_R1;
global dado_R2;
global dado_R3;
global dado_R4;
global dado_R5;
global dado_R13;
global dado_R6;
global dado_R7;
global dado_R8;
global dado_R9;
global dado_R10;
global dado_R11;
global dado_R12;

```

```

global dado_row2;
global dado_col2;
global dado_PtoCont1;
global dado_PtoCont2;
global dado_CV1;
global dado_CV2;
global dado_CV3;
global opc_PR;
global dado_T_PR;
global dado_D_rain;
global dado_Pevent;
global dado_Chuva;
global FS_crit;

global dado_dx;

global ctrol_per;

%
%-----
%Input_Grades;

% Direção de fluxo => Variável: Dire
Dire = load(dado_Dire);

nrows = size(Dire,1);
ncols = size(Dire,2);

% Declividade => Variável: MatSlope
MatSlope = load(dado_Dec); % em graus

% Área Acumulada => Variável: AreaA
AreaA = load(dado_Area); % em m^2

% ks saturado => Variável: Matks
MatKs = load(dado_Ks); % em cm/h

% Permeabilidade saturada do saprólito => Variável: Matkp
MatKp = load(dado_Kp); % em cm/h

% Ângulo de atrito => Variável: MatFriction
MatFriction = load(dado_phi); % em rad

% Peso específico do solo saturado => Variável: MatGammaS
MatGammaS = load(dado_Y); % em kN/m^3

% Profundidade de superfície de ruptura => Variável: MatZs
MatZs = load(dado_Zs); % em kN/m^3

% Coesão => Variável: MatCohesion
MatCohesion = load(dado_C); % em kN/m^2

% Capacidade máxima de armazenamento estático S1max => Variável: MatS1max
MatS1max = load(dado_S1m); % em mm

% Capacidade máxima S3 S3max => Variável: MatS3max
MatS3max = load(dado_S3m); % em mm

% Evapotranspiração Potencial => Variável: Mat
MatEVP = load(dado_EVP); % em mm

%
%-----
%Input_Parametros;

% constantes
GammaW = 10;
fc = 0.0028; % converte cm/h para mm/s

% Condições iniciais
hillslope = dado_hillslope;
channel = dado_channel;
iv2 = dado_iv2;
iv3 = dado_iv3;
iv_4 = dado_iv4;
iv5 = dado_iv5;
si1 = dado_si1;
si2 = dado_si2;
si3 = dado_si3;
si4 = dado_si4;
si5 = dado_si5;
row2 = dado_row2;
col2 = dado_col2;
% Linha do Ponto de Controle

```

```

PtosCont(1,1) = dado_PtoCont1;
% Coluna do Ponto de Controle
PtosCont(2,1) = dado_PtoCont2;

Dados_dx = load(dado_dx);
% Dimensão do pixel
dx = Dados_dx(1);

% coordenadas do canto
x11 = Dados_dx(2);
y11 = Dados_dx(3);

% dados inválidos
nodata = Dados_dx(4);

% Variável que determina se o método probabilístico será realizado em
% apenas 2 análises (contr_passos = 0) ou também em passos intermediários (contr_passos = 1)
contr_passos = opc_PR;

if contr_passos == 1

    tempo_prob = load(dado_T_PR);

end % end if
c_t = 1;

% Parâmetros para calibração do modelo

R1 = dado_R1;
R2 = dado_R2;
R3 = dado_R3;
R4 = dado_R4;
R5 = dado_R5;
R13 = dado_R13;
R6 = dado_R6;
R7 = dado_R7;
R8 = dado_R8;
R9 = dado_R9;
R10 = dado_R10;
R11 = dado_R11;
R12 = dado_R12;

%
%-----
%Input_Rain;

% Arquivo com o intervalo de tempo, n° total de dados, n° de estações pluviométricas
% e variável que diz se serão usadas informações observadas (Qreal)
%arquivo só com 3 informações VETOR
ch = load(dado_D_rain);

% Intervalo entre os dados de chuva
dt = ch(1); % em segundos
% Número total de dados de precipitação
nreg = ch(2);
% Número de estações pluviométricas
npt = ch(3);

% Matriz com as coordenadas dos postos pluviométricos
% Matriz tem as seguintes dimensões: [n° de postos,2(X e Y)]
Pevento = load(dado_Pevent)';
% Dimensões [n° dados de chuva,n° de postos]
precip = load(dado_Chuva)';

%
%-----

%vetor CV com os coeficientes de variação: coesão (Cohesion),atrito
%(Friction), peso específico (GammaS)
CV = zeros(1,3);
CV(1) = dado_CV1;
CV(2) = dado_CV2;
CV(3) = dado_CV3;

SHIA_ACOPL

ctrol_per = 1;

% Mensagem indicando fim das análises de PERIGO
warndlg('Análise do perigo estabelecida com sucesso!','Operação realizada');

% --- Executes on button press in pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
global dado_T_PR;
% hObject handle to pushbutton16 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
[filename_T_PR pathname_T_PR] = uigetfile({'*.txt'},'File Selector');
fullpathname_T_PR = strcat(pathname_T_PR,filename_T_PR);

```

```

set(handles.text117,'String',fullpathname_T_PR);

dado_T_PR = fullfile(pathname_T_PR,filename_T_PR);

function edit24_Callback(hObject, eventdata, handles)
global dado_hillslope;

dado_hillslope = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit24_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit25_Callback(hObject, eventdata, handles)
global dado_channel;

dado_channel = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit25_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton59.
function pushbutton59_Callback(hObject, eventdata, handles)
global dado_dx;

[filename_dx pathname_dx] = uigetfile({'*.txt'},'File Selector');
fullpathname_dx = strcat(pathname_dx,filename_dx);
set(handles.text354,'String',fullpathname_dx);

dado_dx = fullfile(pathname_dx,filename_dx);

% --- Executes when selected object is changed in uipanel27.
function uipanel27_SelectionChangeFcn(hObject, eventdata, handles)

% --- Executes on button press in pushbutton60.
function pushbutton60_Callback(hObject, eventdata, handles)
global dado_Pevent;

[filename_Pevent pathname_Pevent] = uigetfile({'*.txt'},'File Selector');
fullpathname_Pevent = strcat(pathname_Pevent,filename_Pevent);
set(handles.text358,'String',fullpathname_Pevent);

dado_Pevent = fullfile(pathname_Pevent,filename_Pevent);

% --- Executes on button press in pushbutton61.
function pushbutton61_Callback(hObject, eventdata, handles)
global dado_D_rain;

[filename_D_rain pathname_D_rain] = uigetfile({'*.txt'},'File Selector');
fullpathname_D_rain = strcat(pathname_D_rain,filename_D_rain);
set(handles.text357,'String',fullpathname_D_rain);

dado_D_rain = fullfile(pathname_D_rain,filename_D_rain);

% --- Executes during object creation, after setting all properties.
function icone_shia_CreateFcn(hObject, eventdata, handles)

a1 = imread('imag_perigo.jpg');
image(a1);
axis off

% --- Executes when selected object is changed in uipanel30.
function uipanel30_SelectionChangeFcn(hObject, eventdata, handles)
global opc_PR;

% opção para PR padrão
if hObject == handles.radiobutton11

    opc_PR = 0;
    set(handles.pushbutton16, 'enable', 'off');

```

```

% opção opção para PR tempos definidos
elseif hObject == handles.radiobutton12
    opc_PR = 1;
    set(handles.pushbutton16, 'enable', 'on');
end %end if

function edit93_Callback(hObject, eventdata, handles)
global FS_crit;

FS_crit = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit93_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

A.1.2 ROTINA ESPECÍFICA DO MODELO SHIA_LANDSLIDE PROBABILÍSTICO

```

global PR;

%
%Tipo_Mapa;

MatLevel = zeros(nrows,ncols);

for i = 1:nrows
    for j = 1:ncols
        %Verifica se há dados na célula atual
        if AreaA(i,j) ~= nodata
            % Se há dados, faz a classificação
            if AreaA(i,j) < hillslope
                MatLevel(i,j) = 1;

                else if (AreaA(i,j) >= hillslope) && (AreaA(i,j) < channel)
                    MatLevel(i,j) = 2;

                else
                    MatLevel(i,j) = 3;

                end % end if

            end %end if

        else
            %Se não há dados, classifica como dados inválidos
            MatLevel(i,j) = nodata;

        end %end if
    end % end for
end % end

%
%Vetor_Bacia;

area = nrows*ncols;

vfinal = zeros(5,area);

vfinal(1,area) = 1;
vfinal(2,area) = 0;
vfinal(3,area) = row2;
vfinal(4,area) = col2;
vfinal(5,area) = 1;

tenia = 1;
cont2 = 1;
sum1 = 1;

    while col2 > 0
%
% posicion;

cont = 0;

for kf = 1:3
    for kc = 1:3

        c = col2 + kc -2;
        f = row2 + kf -2;

        if (c <= ncols && c>0) && (f <= nrows && f > 0)

```

```

        if Dire(f,c) == 3*(kf)-kc+1
            cont = cont+1;
            aux2 = zeros(2,cont);
            % Salva primeiro a linha e depois a coluna
            aux2(1,cont) = f;
            aux2(2,cont) = c;

            if cont > 1

                for z = 1:cont-1

                    aux2(1,z) = aux1(1,z);
                    aux2(2,z) = aux1(2,z);

                end % end for

            end % end if

            aux1 = zeros(size(aux2,1),size(aux2,2));
            aux1 = aux2;

            clear aux2;
        end % end if

    end % end if

end % end for
end % end for

res = aux1;

%
for i = 1:cont

    vfinal(1,area-tenia+1-i) = tenia+1;
    vfinal(2,area-tenia+1-i) = cont2;
    vfinal(3,area-tenia+1-i) = res(1,i);
    vfinal(4,area-tenia+1-i) = res(2,i);
    vfinal(5,area-tenia+1-i) = 0;

end % end for

    tenia = tenia + cont;

    % Trocadas as ordens
    row2 = vfinal(3,area - cont2);
    col2 = vfinal(4,area - cont2);

    cont2 = cont2 + 1;

end % end while

resultado = zeros(7,cont2-1);

for i = 2:cont2
    for j = 1:5
        resultado(j,cont2-i+1) = vfinal(j,area-i+2);
    end %end for

    Lugar = Dire(vfinal(3,area-i+2),vfinal(4,area-i+2));
    % Tamanho da célula: caminho efetivamente percorrido
    prueba = mod(Lugar,2);
    % Se o resto é zero (direção 2,4,6 ou 8) comprimento é dx
    if (prueba == 0)
        long = dx;
    else
        long = sqrt(2*dx^2);
    end % end if

    resultado(6,cont2-i+1) = long;

end % end for

for i = 1:cont2-1

    if (resultado(3,i) == PtosCont(1,1)) && (resultado(4,i) == PtosCont(2,1))
        resultado(5,i)=2;

        break;
    end % end if

```

```

end % end for

cont2 = cont2-1;
tam1 = 7;
Trows = cont2;
Tcols = tam1;

%
% Matriz;

% definição das variáveis
Level = zeros(1,Trows+1);
EVP = zeros(1,Trows);
Slmax = zeros(1,Trows);
S3max = zeros(1,Trows);
Ks = zeros(1,Trows);
Kp = zeros(1,Trows);
Slope = zeros(1,Trows);
Cohesion = zeros(1,Trows);
Friction = zeros(1,Trows);
Acum = zeros(1,Trows);
GammaS = zeros(1,Trows);
Zs = zeros(1,Trows);

for i = 1:Trows

    Level(i) = MatLevel(resultado(3,i),resultado(4,i)); % sem unidades

    Slmax(i) = MatSlmax(resultado(3,i), resultado(4,i)); % mm

    % se para alguma célula Slmax é nulo, atribui-se um valor de 1 mm
    if Slmax(i) == 0
        Slmax(i) = 1;
    end %end if

    S3max(i) = MatS3max(resultado(3,i), resultado(4,i)); % em mm

    if S3max(i) == 0
        S3max(i) = 1;
    end %end if

    EVP(i) = MatEVP(resultado(3,i),resultado(4,i)); % [mm]
    Ks(i) = MatKs(resultado(3,i),resultado(4,i)); % [cm/h]
    Kp(i) = MatKp(resultado(3,i),resultado(4,i)); % [cm/h]

    % Vetores com os parâmetros geotécnicos
    Friction(i) = MatFriction(resultado(3,i), resultado(4,i)); % [Rad]
    Cohesion(i) = MatCohesion(resultado(3,i), resultado(4,i)); % [kN/m^2]
    GammaS(i) = MatGammaS(resultado(3,i), resultado(4,i)); % [kN/m^3]
    Zs(i) = MatZs(resultado(3,i), resultado(4,i)); % [m]

    % Vetores com as variáveis do MNT
    Slope(i) = MatSlope(resultado(3,i), resultado(4,i)); % [Rad]
    Acum(i) = AreaA(resultado(3,i),resultado(4,i)); % m^2

    if Acum(i) == 0
        Acum(i) = dx^2; % [m^2]
    end %end if

end % end for

%
% Rainfall

% Part 1
% Lista de variáveis criadas pelos autores
% integer k,z,a,b,c,g,flag,flag1,flag2,cont1,cont2,cont3,tam2,prueba,n
% real Ymin(2),Ymax(2),Ymin1,Ymax1,Xmed(2),Xmed1,Ptri(2,3),mL,mR,xL,xR,m1,a1,m2,a2,ar,al
% real Ax,Ay,Bx,By,Cx,Cy, radio, dist,xmin1,xmin2,xmax1,xmax2,ymin2,ymax2,x,y,ccx,ccy
% real, allocatable :: Pt(:, :)
% integer, allocatable :: conex(:,:),conex2(:,:),Tri(:, :)
% real*8 punto(2,1),oriP,Ori1,Ori2,Ori3
% real, allocatable :: mapapert(:, :)
% integer, allocatable :: perte(:)
% real, allocatable :: tabla(:, :)

% Cria o triângulo que abrange os demais pontos

% nreg = n° total de dados de chuva
%npt = n° de estações pluviométricas
% Matriz 'precip' contém os dados de chuva; Dimensão = [npt,nreg]
% Matriz 'Pevento' contém a localização de cada estação; Dimensão = [2,npt]

Ymax1 = max(Pevento(2,:));
Ymin1 = min(Pevento(2,:));

```

```

    if Ymin1 == 0
        Ymin1 = -1;
    end % end if

% Valor médio para as células de x
Xmed1 = mean(Pevento(1,:));

% começa a fazer iterações buscando o triângulo que contenha todos os pontos
%Designa los puntos originales del triangulo imaginario

Ptri = zeros(2,3);

Ptri(1,1) = Xmed1;
Ptri(2,1) = Ymax1+0.5*Ymax1;
Ptri(1,2) = Xmed1+abs(2*Xmed1);
Ptri(2,2) = Ymin1-abs(0.5*Ymin1);
Ptri(1,3) = Xmed1-abs(2*Xmed1);
Ptri(2,3) = Ymin1-abs(0.5*Ymin1);

%Comienza a iterar el triangulo imaginario
flag1 = 0;

while flag1 == 0

    flag2 = 0;

    %Define la pendiente de ambos lados del triangulo

    mR=(Ptri(2,1)-Ptri(2,2))/(Ptri(1,1)-Ptri(1,2));
    aR=Ptri(2,1)-mR*Ptri(1,1);
    mL=(Ptri(2,1)-Ptri(2,3))/(Ptri(1,1)-Ptri(1,3));
    aL=Ptri(2,1)-mL*Ptri(1,1);

    % Busca si los puntos se encuentran dentro del triangulo
    for i=1:npt

        xR = (Pevento(2,i)-aR)/mR;
        xL = (Pevento(2,i)-aL)/mL;

        if Pevento(1,i) > xR || Pevento(1,i) < xL

            flag2 = flag2+1;

        end % end if
    end % end for

    %Si alguno de los puntos queda por fuera, crece el tamaño del triangulo
    if flag2 ~= 0
        Ptri(1,2) = Ptri(1,2) + abs(2*Xmed1);
        Ptri(1,3) = Ptri(1,3) - abs(2*Xmed1);
        flag = 0;
    else
        flag1 = 1;
    end % end if

end % end while

%
%Comienza a introducir puntos dentro del triangulo
%Crea un vector de tamaño n+3 y le asigna todo lo existente

Pt = zeros(2,npt+3);
conex = zeros(npt+3,npt+3);
Tri = zeros(3,npt*npt*2);

Tri(:,1:npt*npt) = -9999;

Pt(:,1:npt) = Pevento(:,1:npt);
Pt(:,npt+1:npt+3) = Ptri(:,1:3);

%Comienza la agregación de puntos
cont1 = 1;
cont2 = 1;
Tri(1,1)=npt+3;
Tri(2,1)=npt+2;
Tri(3,1)=npt+1;

for a=1:npt

    %Limpia los conexiones en cada iteración

    %Observa en que triangulos cae el pt introducido
    %tam1=size(tri,DIM=2)

    for b =1:cont2

```

```

if Tri(1,b) ~= -9999
    %Obtiene Ax, Ay, Bx, By, etc...
    Ax=Pt(1, Tri(1,b));
    Ay=Pt(2, Tri(1,b));
    Bx=Pt(1, Tri(2,b));
    By=Pt(2, Tri(2,b));
    Cx=Pt(1, Tri(3,b));
    Cy=Pt(2, Tri(3,b));

    %Calcula el centro en X
    CCx=(By*Ax^2 - Cy*Ax^2 - By^2*Ay + Cy^2*Ay + Bx^2*Cy + Ay^2*By + Cx^2*Ay - Cy^2*By -
Cx^2*By - Bx^2*Ay + By^2*Cy - Ay^2*Cy)/(2*(Ay*Cx+By*Ax-By*Cx-Ay*Bx-Cy*Ax+Cy*Bx));
    %Calcula el centro en Y
    CCy=(Ax^2*Cx+ Ay^2*Cx + Bx^2*Ax - Bx^2*Cx + By^2*Ax - By^2*Cx - Ax^2*Bx -Ay^2*Bx -
Cx^2*Ax + Cx^2*Bx - Cy^2*Ax + Cy^2*Bx)/(2*(Ay*Cx+By*Ax-By*Cx-Ay*Bx-Cy*Ax+Cy*Bx));

    %calcula el radio del circulo
    radio=sqrt((CCx-Pt(1, Tri(1,b)))^2+(CCy-Pt(2, Tri(1,b)))^2);
    %Observa si el punto queda dentro de la circunferencia
    dist=sqrt((CCx-Pt(1,a))^2+(CCy-Pt(2,a))^2);

    if (dist<radio)
        %Crea las conexiones pertenecientes al poligono de incernción
        conex(Tri(1,b), Tri(2,b))=conex(Tri(1,b), Tri(2,b))+1;
        conex(Tri(2,b), Tri(1,b))=conex(Tri(2,b), Tri(1,b))+1;
        conex(Tri(1,b), Tri(3,b))=conex(Tri(1,b), Tri(3,b))+1;
        conex(Tri(3,b), Tri(1,b))=conex(Tri(3,b), Tri(1,b))+1;
        conex(Tri(3,b), Tri(2,b))=conex(Tri(3,b), Tri(2,b))+1;
        conex(Tri(2,b), Tri(3,b))=conex(Tri(2,b), Tri(3,b))+1;

        %Destruye el triangulo si este cumple la regla
        Tri(1:3,b) = -9999;

    end % end if
end % end if
end % end for

%Selecciona los puntos donde existe conexión
for c = 1:size(conex,2)

    if sum(conex(c,:)) ~= 0

        conex(c,a)=1;

    end %end for
end %end for

%Obtiene los triangulos nuevos
for i = 1:npt+3-1
    if conex(i,a) == 1

        for j = i+1:npt+3
            if conex(j,a) == 1
                if conex(j,i) == 1
                    cont2 = cont2+1;
                    %Tri: matriz que indica que nodos pertenecen a que triangulo
                    Tri(1,cont2) = a;
                    Tri(2,cont2) = i;
                    Tri(3,cont2) = j;

                end %end if
            end % end if
        end %end for
    end % end if
end % end for
end % end for

%
%-----
%Borra los triangulos imaginarios
for i = 1:cont2
    if (Tri(1,i) > npt) || (Tri(2,i) > npt) || (Tri(3,i) > npt)
        Tri(1:3,i)=-9999;
    end %end if
end %end for
%
%-----

%Completa triangulos incompletos
%Limpia los conexiones en cada iteración
conex2 = zeros(npt,npt);

%Indica las conexiones dadas por los triangulos
for i = 1:cont2
    if Tri(1,i) ~= -9999
        conex2(Tri(1,i),Tri(2,i)) = 1;
        conex2(Tri(2,i),Tri(1,i)) = 1;
        conex2(Tri(1,i),Tri(3,i)) = 1;
    end
end

```

```

        conex2(Tri(3,i),Tri(1,i)) = 1;
        conex2(Tri(2,i),Tri(3,i)) = 1;
        conex2(Tri(3,i),Tri(2,i)) = 1;
    end %end if
end %end for

%Observa que conexiones faltan
for i = 1:npt
    for j=i+1:npt

        if (conex2(j,i)~=1)

            if (Pevento(2,i)~=Pevento(2,j))

                if (Pevento(1,i)~=Pevento(1,j))

                    %Calcula la pendiente e intercepto

                    m1 = (Pevento(2,i)-Pevento(2,j))/(Pevento(1,i)-Pevento(1,j));

                    a1 = Pevento(2,i)-m1*Pevento(1,i);

                    prueba = 0;

                    %Compara la linea con las demas a ver si se cruza
                    for z = 1:cont2

                        if Tri(1,z)~= -9999

                            a = 2;

                            for k = 1:2

                                for g = a:3

                                    %calcula pendiente e intercepto

                                    m2 = (Pevento(2,Tri(k,z))-Pevento(2,Tri(g,z)))/(Pevento(1,Tri(k,z))-
Pevento(1,Tri(g,z)));

                                    a2 = Pevento(2,Tri(k,z))-m2*Pevento(1,Tri(k,z));
                                    %Calcula el intercepto

                                    x = (a1-a2)/(m2-m1);
                                    y = (a1*m2-a2*m1)/(m2-m1);

                                    %Encuentra maximos y minimos
                                    xmin1 = min(Pevento(1,i), Pevento(1,j)); xmin2=min(Pevento(1,Tri(k,z)),
Pevento(1,Tri(g,z)));
                                    xmax1 = max(Pevento(1,i), Pevento(1,j)); xmax2=max(Pevento(1,Tri(k,z)),
Pevento(1,Tri(g,z)));
                                    ymin1 = min(Pevento(2,i), Pevento(2,j)); ymin2=min(Pevento(2,Tri(k,z)),
Pevento(2,Tri(g,z)));
                                    ymax1 = max(Pevento(2,i), Pevento(2,j)); ymax2=max(Pevento(2,Tri(k,z)),
Pevento(2,Tri(g,z)));

                                    %Observa si el intercepto se encuentra entre ambas linea
                                    if (x > xmin1) && (x < xmax1) && (y>ymin1) && (y<ymax1) && (x>xmin2) &&
(x<xmax2) && (y>ymin2) && (y<ymax2)

                                        prueba = prueba+1;

                                    end %end if

                                end %end for

                            end %end for

                            a = a+1;

                        end %end for
                    end %end if
                end % end for

                % Si no cruza ninguna linea se genera la nueva linea
                if prueba == 0
                    conex2(i,j)=1; conex2(j,i)=1;
                end %end if

            end %endif
        end %end if
    end % end if
end %end for
end %end for

%
%Selecciona finalmente los triangulos

```

```

%Vacía el vector
Tri(1:3,1:npt*npt) = -9999;

%Encuentra los triángulos
cont3 = 0;

for z = 1:npt
    for i = 1:npt
        if conex2(i,z) == 1
            for j = i+1:npt
                if conex2(j,z) == 1
                    if conex2(j,i) == 1

                        %Tri: matriz que indica que nodos pertenecen a que triángulo.
                        cont3 = cont3+1;
                        Tri(1,cont3) = z;
                        Tri(2,cont3) = i;
                        Tri(3,cont3) = j;

                    end %end if
                end %end if
            end %end for

            conex2(z,i) = 2;

        end %end if
    end %end for

    conex2(npt,z)=2;

end %end for

%Deja el vector listo pa ser impreso
Trif = zeros(3,cont3);

Trif(1:3,1:cont3) = Tri(1:3,1:cont3);
%
% Part 2: Asignación

%Lectura de variables
%Comienza a buscar el triángulo de pertenencia de cada uno de los puntos de la cuenca
n = Trows;
perte = zeros(1,n);

for i = 1:n

    %Inicializa una bandera que revisa si el punto se encuentra dentro de algún triángulo
    flag = 0;
    %Obtiene el punto a evaluar
    punto(1,1) = xll+resultado(3,i)*dx-0.5*dx;
    punto(2,1) = yll+(nrows-resultado(4,i))*dx+0.5*dx;

    %Comienza a buscar el triángulo de pertenencia
    for j=1:cont3
        %Calcula la orientación del triángulo
        oriP = (Pevento(1,Trif(1,j))-Pevento(1,Trif(3,j)))*(Pevento(2,Trif(2,j))-Pevento(2,Trif(3,j)))-
        (Pevento(2,Trif(1,j))-Pevento(2,Trif(3,j)))*(Pevento(1,Trif(2,j))-Pevento(1,Trif(3,j)));
        %Calcula la orientación de los triángulos resultantes de unir P con los vértices del Tri Ppal.
        %Ori1: A1A2P
        ori1=(Pevento(1,Trif(1,j))-punto(1,1))*(Pevento(2,Trif(2,j))-punto(2,1))-(Pevento(2,Trif(1,j))-
        punto(2,1))*(Pevento(1,Trif(2,j))-punto(1,1));

        %Ori2: A2A3P
        ori2=(punto(1,1)-Pevento(1,Trif(3,j)))*(Pevento(2,Trif(2,j))-Pevento(2,Trif(3,j)))-(punto(2,1)-
        Pevento(2,Trif(3,j)))*(Pevento(1,Trif(2,j))-Pevento(1,Trif(3,j)));

        %Ori3: A3A1P
        ori3 = (Pevento(1,Trif(1,j))-Pevento(1,Trif(3,j)))*(punto(2,1)-Pevento(2,Trif(3,j)))-
        (Pevento(2,Trif(1,j))-Pevento(2,Trif(3,j)))*(punto(1,1)-Pevento(1,Trif(3,j)));
        %Si las cuatro orientaciones son o positivas o negativas el punto se encuentra dentro

        if ((oriP>0) && (ori1>0) && (ori2>0) && (ori3>0)) || ((oriP<0) && (ori1<0) && (ori2<0) && (ori3<0))
            perte(i) = j;
            resultado(7,i) = j;
            flag=1;
        end %end if
    end %end for

    if flag == 0
        perte(i) = nodata;
        resultado(7,i) = 0;
    end %end if

end % end for
%
%Model2

```

```

R = zeros(Trows,5);
S1 = zeros(1,Trows);
S2 = zeros(1,Trows);
S3 = zeros(1,Trows);
S4 = zeros(1,Trows);
S5 = zeros(1,Trows);
vn = zeros(1,Trows);
%
-----
az = ones(2,Trows);
bz = ones(2,Trows);
cz = ones(2,Trows);
dz = ones(2,Trows);
coef = ones(1,Trows);

% MODIFICADO CRIADO
a3 = zeros(1,Trows);
b3 = zeros(1,Trows);
c3 = zeros(1,Trows);
coef1 = zeros(1,Trows);
E = zeros(1,Trows);
Z3 = zeros(1,Trows);
Areal = zeros(1,Trows);
E3 = zeros(1,Trows);
E2 = zeros(1,Trows);
E4 = zeros(1,Trows);
E5 = zeros(1,Trows);
drenaid = zeros(1,Trows);
Auxi = 1:1:Trows;
% Método PROBABILÍSTICO
FS_p = zeros(2,Trows);
Var = zeros(3,Trows);
%
-----
Zw = zeros(1,Trows);
Bo = zeros(1,Trows);
Est = zeros(1,Trows);
Stream_flow = zeros(1,nreg);
Subsur_level = zeros(1,nreg);
v2 = zeros(1,Trows);
v3 = zeros(1,Trows);
v4 = zeros(1,Trows);
v5 = zeros(1,Trows);
kpp = zeros(1,Trows);
Zmin = zeros(1,Trows);
Zwcrit = zeros(1,Trows);
Zmax = zeros(1,Trows);

if contr_passos == 1
    MatEst = zeros(nrows,ncols,2+size(tempo_prob,2));
else
    MatEst = zeros(nrows,ncols,2);

end % end if

%MatEst1 = zeros(ncols,nrows);
%MatEst2 = zeros(ncols,nrows);
%
-----
% Parâmetros para Geomorphological Kinematic Wave
es = 0.1667;
teta = 1.26;

% alfa1: Wb & Qb [rango: 0.34 a 0.55]
alfa1 = 0.5;
%alfa2: Wt & Qt [rango: 0.05 a 0.2]
alfa2 = 0.20;
% k1: Qb & Acum [rango: 0.5 a 0.75]
k1 = 0.6;
%Phi: Qb & Acum [rango: 0.65 a 0.8]
phi = 0.75;
%c1: Calibration coefficient [rango: 0.5 a 5.75]
c1 = 3.26;
%cd:
cd = 20;
%cn:
cn = 0.047;

% Parâmetros para overland flow velocity e subsurface flow velocity

epsi=0.038;
%e1:
e1=0.315;
%mn: manning
manning = 0.5;
%
-----
% Começo da balanço de água [mm]

```

```

entradas = 0;
salidas = 0;

% For calculating RMSE & NS
SE = 0;
Nash = 0;

if contr == 1
    mean1 = sum(Qreal)/nreg;
end % end if

% Calculando os coeficientes GWK e os expoentes
ey = 2/3-(es*teta);
ez = 1/(1+ey*alfa2);
K2 = (cn*cd^es*(c1^ey)*(k1^(ey*(alfa1-alfa2))))^(-ez);
w1=ey*ez*(1-alfa2);
w2=-ez*ey*fhi*(alfa1-alfa2);
w3=ez*(ey-1/6);

%To convert mm a m^3
conver=(dx^2)/1000; %[m2]*[m/mm]=[m3/mm]
% Inicializa-se esta matriz com os valores inválidos porque o zero
MatEst(:, :) = nodata;

%condições iniciais dos tanques para cada pixel
S1(:, :) = si1; %[mm]
S2(:, :) = si2; %[mm]
S3(:, :) = si3; %[mm]
S4(:, :) = si4; %[mm]
S5(:, :) = si5; %[mm]

% Processo de calibração

S1max(:, :) = S1max(:, :) * R1; %[mm]
EVP(:, :) = EVP(:, :) * R2; %[mm]
Ks(:, :) = Ks(:, :) * R3 * dt * fc; %[mm]
Kp(:, :) = Kp(:, :) * R4 * dt * fc; %[mm]
kpp(:, :) = Kp(:, :) * R5 * dt * fc; %[mm]
v2(:, :) = iv2 * R6; %[m/sec]
v3(:, :) = iv3 * R7; %[m/sec]
v4(:, :) = iv4 * R8; %[m/sec]
v5(:, :) = iv5 * R9; %[m/sec]
Cohesion(:, :) = Cohesion(:, :) * R10; %[KPa]
Friction(:, :) = Friction(:, :) * R11; %[Rad]
Zs(:, :) = Zs(:, :) * R12; %[m]
S3max(:, :) = S3max(:, :) * R13;

%
% MÉTODO FOSM

% Valor médio do FS
FS_p(1, :) =
(Cohesion(1, :)*GammaS(1, :)*Zs(1, :)*(cos(Slope(1, :)))^2*tan(Friction(1, :)))/(GammaS(1, :)*Zs(1, :)*sin(
Slope(1, :))*cos(Slope(1, :)));

% Cálculo das parcelas das variâncias

% Em relação a coesão
Var(1, :) = ( (1./(GammaS(1, :)*Zs(1, :)*cos(Slope(1, :))*sin(Slope(1, :)))) ^2
).*((CV(1)/100)*Cohesion(1, :))^2;

% Em relação ao atrito
Var(2, :) = ((cos(Slope(1, :))*tan(Friction(1, :))^2 + 1)/sin(Slope(1, :))
)^2).*((CV(2)/100)*Friction(1, :))^2;

% Em relação ao peso específico
Var(3, :) = ((cos(Slope(1, :))*tan(Friction(1, :)))/(GammaS(1, :)*sin(Slope(1, :))) -
(GammaS(1, :)*Zs(1, :)*tan(Friction(1, :))*cos(Slope(1, :)))^2 +
Cohesion(1, :))/(GammaS(1, :)^2*Zs(1, :)*cos(Slope(1, :))*sin(Slope(1, :))
)^2).*((CV(3)/100)*GammaS(1, :))^2;

% cálculo do desvio padrão
FS_p(2, :) = sqrt(sum(Var(:, :)));

% EQUAÇÃO DO FATOR DE SEGURANÇA
%FS = (c+(gama*Z-GamaW*Zw)*(cos(beta)^2*tan(Atrito)))/(gama*Z*sin(beta))*cos(beta)
%
% WaitBar para a análise de Susceptibilidade

h = waitbar(0, '1', 'Name', 'Análise de Susceptibilidade...', ...
'CreateCancelBtn', ...
'setappdata(gcf, 'canceling', 1)');
setappdata(h, 'canceling', 0)
% Parâmetro da Waitbar
steps = Trows;

```

```

for i = 1:Trows

    %balanço hidrico

    entradas = entradas + S1(i) + S2(i) + S3(i) + S4(i) + S5(i)/conver;

    % Toma os valores para a iteração da chuva

    if (resultado(7,i) == 0) %Just to be sure there is one trinagle for each pixel
        resultado(7,i) = 1;
    end % end if

    az(1,i) = Pevento(1,Trif(1,resultado(7,i)));
    az(2,i) = Pevento(2,Trif(1,resultado(7,i)));
    bz(1,i) = Pevento(1,Trif(2,resultado(7,i)));
    bz(2,i) = Pevento(2,Trif(2,resultado(7,i)));
    cz(1,i) = Pevento(1,Trif(3,resultado(7,i)));
    cz(2,i) = Pevento(2,Trif(3,resultado(7,i)));
    dz(1,i) = xll+resultado(3,i)*dx-0.5*dx;
    dz(2,i) = yll+(nrows-resultado(4,i))*dx+0.5*dx;
    coef(i) = (bz(1,i)-az(1,i))*(cz(2,i)-az(2,i))-(cz(1,i)-az(1,i))*(bz(2,i)-az(2,i));

    %To be sure S1(j) <= S1max(j)
    if (S1(i) > S1max(i))
        S1max(i) = S1(i);
    end %end if

% -----
% PRIMEIRA ANÁLISE DE ESTABILIDADE

    x = -10:0.01:(FS_crit-FS_p(1,i))/FS_p(2,i);

    Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

    MatEst(resultado(3,i),resultado(4,i),1) = trapz(x,Y);

    % Chamada da waitbar de Susceptibilidade
    waitbar(i / steps,h,sprintf('%0.1f ',i/steps*100))

end % end for i

delete(h)

% -----
% VERIFICAR SE TEM QUE ZERAR A MATRIZ DE RESULTADOS PARA INICIAR AS ITERAÇÕES COM AS CHUVAS
%MatEst(:, :) = nodata;

h = waitbar(0,'1','Name','Análise de Perigo...','...
    'CreateCancelBtn','...
    'setappdata(gcf,'canceling',1)');
setappdata(h,'canceling',0)

    steps = nreg

% Iteração para cada intervalo de tempo
for i = 1:nreg
    % cont = 0;

    %Getting the value where the pixel drainage
    dremaid(1,:) = resultado(1,:)-resultado(2,:) + Aux1;

    for j = 1:Trows
        a3(j) = precip(Trif(1,resultado(7,j)),i);
        b3(j) = precip(Trif(2,resultado(7,j)),i);
        c3(j) = precip(Trif(3,resultado(7,j)),i);
        % Getting the coefficient to obtain the rainfall

    end %end for

    coef1(1,:)=(cz(1,:)-az(1,:)).*(dz(2,:)-az(2,:)).*(b3(1,:)-a3(1,:))+ (bz(2,:)-
az(2,:)).*(c3(1,:)-a3(1,:)).*(dz(1,:)-az(1,:))- (b3(1,:)-a3(1,:)).*(cz(2,:)-az(2,:)).*(dz(1,:)-az(1,:))-
(dz(2,:)-az(2,:)).*(c3(1,:)-a3(1,:)).*(bz(1,:)-az(1,:));
    R(:,1) = a3(:)'-coef1(:)'/coef(:)'; % [mm]
    entradas = entradas+R(:,1); % [mm]

    R(:,2)=(R(:,1))' - min(R(:,1))' .*(1-(S1(1,:)/S1max(1,:)).^2, S1max(1,:)-S1(1,:))'; % [mm]
    S1(1,:) = min(S1(1,:)+R(:,1)'-R(:,2)', S1max(1,:)); % [mm]
    E(1,:) = min(EVP(1,:).*(S1(1,:)/S1max(1,:)).^0.6, S1(1,:)); % [mm]
    S1(1,:) = S1(1, :)-E(1, :);
    salidas = salidas + sum(E); % [mm]

    R(:,3) = min(R(:,2), Ks(1, :)); % [mm]

    R(:,4) = min(R(:,3), Kp(1, :)); % [mm]

    %T3 water level
    S3(1,:)=min(S3(1,:)+R(:,1)'-R(:,4)', S3max(1,:)); % [mm]
    %T3 excedence

```

```

Z3(1,:) = max(0,S3(1,:)+R(:,3)'-R(:,4)'+S3max(1,:));

%Subsurface outflow using the equation of Kubota & Sivapalan (1995) for mountain basins
covered by forests and b=2
for z =1:3
    Areal(1,:) = S3(1,:).*conver./(dx+v3(1,:)*dt); %[m^2]
    vn(1,:)=
R7.*((Ks(1,:)./(dt*1000)).*sin(Slope(1,:)).*(Areal(1,:)*dx).^2)/(3*((S3max(1,:)./1000).*(dx^2)).^2);
    v3(1,:)=(2*vn(1,:)+v3(1,:))./3;
end % end for

E3(1,:) = Areal(1,:).*v3(1,:)*dt/conver; %[mm]

%Subsurface outflow usin linear reservoir equation
%E3=(1-resultado(6,j)/(v3(j)*dt+resultado(6,j)))*S3(j) ![mm], resultado(6,j) =ddx(j)
%T3 level updating
S3(1,:)= S3(1,:) - E3(1,:); %[mm]

%Water balance

    salidas = salidas + E3(1,Trows);

%Second Tank T2 (Surface storage)
%T2 water level
S2(1,:)=S2(1,:)+R(2)-R(:,3)'+Z3(1,:); %[mm]

%Overland flow velocity
for z = 1:3
    Areal(1,:) = S2(1,:).*conver./(dx+v2(1,:)*dt); %[m^2]
    % Using an equation function of transversal section area ofthe flow over natural
surfaces
    vn(1,:) = R6.*(epsi.*Areal(1,:).^ (0.667*e1).*(atan(Slope(1,:)).^0.5))./manning;
    v2(1,:) = (2.*vn(1,:)+v2(1,:))./3;
end %end for

%Flow out
E2(1,:) = Areal(1,:).*v2(1,:).*dt./conver; %[mm]

%E2=(1-resultado(6,j)/(v2(j)*dt+resultado(6,j)))*S2(j) ![mm], resultado(6,j) =ddx(j)
%T2 level updating
S2(1,:)=S2(1,:)-E2(1,:); %[mm]

%Water balance

    salidas = salidas + E2(1,Trows);
%
%-----

%Four Tank T4 (Aquifer)

%Groundwater outflow
R(:,5) = min(R(:,4),kpp(1,:)); %[mm]
%T4 water level
S4(1,:) = S4(1,:)+R(:,4)'+R(:,5)'; %[mm]
%Flow out using a linear equation
E4(1,:) = (1-resultado(6,:)./(v4(1,:).*dt+resultado(6,:))).*S4(1,:); %[mm],
resultado(6,j) =ddx(j)
%T4 level updating
S4(1,:) = S4(1,:)-E4(1,:); %[mm]

%Water balance

    salidas = salidas + E4(1,Trows);
%
%-----

for z = 1:3
    Areal=S5(1,:)./(dx+(v5(1,:).*dt)); %[m2]
    vn(1,:)=R9*K2.*(Areal(1,:).^w1).*(Acum(1,:).^w2).*(atan(Slope(1,:)).^w3);
    v5(1,:)=(2*vn(1,:)+v5(1,:))/3;
end % end for

E5(1,:) = Areal(1,:).*v5(1,:)*dt; %[m3]
S5(1,:) = S5(1,:) - E5(1,:); %[m3]

salidas = salidas+E5(1,Trows)/conver;

%calculating perched water table
Zw(1,:) = (S3(1,:).*Zs(1,:))./S3max(1,:);

if (contr_passos == 1) && (c_t <= size(tempo_prob,2))
    if (tempo_prob(c_t) == i)

        FS_p(1,:) = (Cohesion(1,:)+(GammaS(1,:).*Zs(1,:)-
GammaW.*Zw(1,:)).*(cos(Slope(1,:)).^2.*tan(Friction(1,:)))./(GammaS(1,:).*Zs(1,:).*sin(Slope(1,:)).*cos(Sl
ope(1,:)));

        % Cálculo das parcelas das variâncias

```

```

% Em relação a coesão
Var(1,:) = ( (1./ (GammaS(1,:).*Zs(1,:).*cos(Slope(1,:)).*sin(Slope(1,:))))
.^2 ).*(CV(1)/100).*Cohesion(1,:).^2;

% Em relação ao atrito
Var(2,:) = (( cos(Slope(1,:)).*(GammaS(1,:).*Zs(1,:) -
GammaW.*Zw(1,:)).*(tan(Friction(1,:)).^2 + 1))./(GammaS(1,:).*Zs(1,:).*sin(Slope(1,:)))
).^2).*(CV(2)/100).*Friction(1,:).^2;

% Em relação ao peso específico
Var(3,:) = ((
cos(Slope(1,:)).*tan(Friction(1,:))./(GammaS(1,:).*sin(Slope(1,:))) -
(tan(Friction(1,:)).*(GammaS(1,:).*Zs(1,:) - GammaW.*Zw(1,:)).*cos(Slope(1,:)).^2 +
Cohesion(1,:))./(GammaS(1,:).^2.*Zs(1,:).*cos(Slope(1,:)).*sin(Slope(1,:)))
).^2).*(CV(3)/100).*GammaS(1,:).^2;

% Em relação à espessura do solo
%Var(4,:) = ((
cos(beta(1,:)).*tan(Friction(1,:))./(Zs(1,:).*sin(beta(1,:))) -
(GammaS(1,:).*Zs(1,:).*tan(Friction(1,:)).*cos(beta(1,:)).^2 +
Cohesion(1,:))./(GammaS(1,:).*Zs(1,:).^2*cos(beta(1,:)).*sin(beta(1,:)))
).^2).*(CV(4).*Zs(1,:).^2;
%cálculo do desvio padrão
FS_p(2,:) = sqrt(sum(Var(:,:)));

c_t = c_t+1;

end % end if

else
if i == nreg

FS_p(1,:) = (Cohesion(1,:)+(GammaS(1,:).*Zs(1,:)-
GammaW.*Zw(1,:)).*(cos(Slope(1,:)).^2.*tan(Friction(1,:)))./(GammaS(1,:).*Zs(1,:).*sin(Slope(1,:)).*cos(Sl
ope(1,:)));

% Cálculo das parcelas das variâncias

% Em relação a coesão
Var(1,:) = ( (1./ (GammaS(1,:).*Zs(1,:).*cos(Slope(1,:)).*sin(Slope(1,:))))
.^2 ).*(CV(1).*Cohesion(1,:).^2;

% Em relação ao atrito
Var(2,:) = (( cos(Slope(1,:)).*(GammaS(1,:).*Zs(1,:) -
GammaW.*Zw(1,:)).*(tan(Friction(1,:)).^2 + 1))./(GammaS(1,:).*Zs(1,:).*sin(Slope(1,:)))
).^2).*(CV(2).*Friction(1,:).^2;

% Em relação ao peso específico
Var(3,:) = ((
cos(Slope(1,:)).*tan(Friction(1,:))./(GammaS(1,:).*sin(Slope(1,:))) -
(tan(Friction(1,:)).*(GammaS(1,:).*Zs(1,:) - GammaW.*Zw(1,:)).*cos(Slope(1,:)).^2 +
Cohesion(1,:))./(GammaS(1,:).^2.*Zs(1,:).*cos(Slope(1,:)).*sin(Slope(1,:)))
).^2).*(CV(3).*GammaS(1,:).^2;

%cálculo do desvio padrão
FS_p(2,:) = sqrt(sum(Var(:,:)));

end % end if

end % end if

for j = 1:Trows

%Pass the water to the appropriate grid cell

if (resultado(2,j)~=0)
if (Level(drenaid(1,j))==1)
S2(drenaid(1,j)) = S2(drenaid(1,j)) + E5(j);

else

S5(drenaid(1,j)) = S5(drenaid(1,j))+E5(j); %[m3]

end % end if
end %end if

%Control points
if resultado(5,j) == 2

%cont = cont+1
Stream_flow(i)=E5(j)/dt; %[m3/s]
Subsur_level(i)=S3(j)*Zs(j)/S3max(j); %[mm]

end %end if

```

```

%SECOND STABILITY ANALYSES

```

```

if (contr_passos == 1) && (c_t <= size(tempo_prob,2))
    x = -10:0.01:(FS_crit -FS_p(1,j))/FS_p(2,j);
    Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);
    MatEst(resultado(3,j),resultado(4,j),c_t+1) = trapz(x,Y);
else if i == nreg
    x = -10:0.01:(FS_crit-FS_p(1,j))/FS_p(2,j);
    Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);
    MatEst(resultado(3,j),resultado(4,j),c_t+1) = trapz(x,Y);
    end % end if
end % end if
end % end for j

% Chamada da waitbar para Perigo
waitbar(i / steps,h,sprintf('%0.f ',i/steps*100))

end % end for i

PR = MatEst;
delete(h)

```

A.2 FUNÇÕES PARA MODELO DE CURVAS DE PROBABILIDADE DE RUPTURA

A.2.1 INTERFACE GRÁFICA

```

function varargout = CURVAS_NIVEL_IMPORTA(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @CURVAS_NIVEL_IMPORTA_OpeningFcn, ...
                  'gui_OutputFcn',  @CURVAS_NIVEL_IMPORTA_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function CURVAS_NIVEL_IMPORTA_OpeningFcn(hObject, eventdata, handles, varargin)
global opcao;

opcao = 1;

handles.output = hObject;

guidata(hObject, handles);

handles.icone_curva = imread('imag_perigo.jpg');
image(handles.icone_curva);
axis off

function varargout = CURVAS_NIVEL_IMPORTA_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function pushbutton2_Callback(hObject, eventdata, handles)
% BOTÃO SAIR

delete(get(hObject, 'parent'));

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
global ctrol_per;

global dado_coef;

```

```

global dado_pos_curva;
global dado_coef_a;
global dado_coef_b;
global dado_coef_c;
global dado_nivel_pr;
global dado_nivel;
global dado_pos_nivel;
global dado_chuva;
global dado_grade;
global opcao;
global num_passos;
global PR;

switch opcao
% OPÇÃO DE CURVAS DISCRETAS
case 1
    entrada = InputHazard(num_passos);
    entrada = InputChuva(entrada,dado_chuva);
    entrada = InputCurva(entrada,dado_coef,dado_pos_curva);

    %Criando e inicializando o objeto
    curva =
hazard(entrada.posicao_curvas,entrada.matrix_chuva,entrada.coef_curva,entrada.tamanho_area);

    curva = ProbRup(curva);
    PR = curva.Matrix_Pr;
    ctrol_per = 1;

    % Mensagem indicando fim das análises de PERIGO
    warndlg('Análise de ameaça estabelecida com sucesso!','Operação realizada');

% OPÇÃO DE CURVAS CONTÍNUAS
case 2
    entrada = InputHazard(num_passos);
    entrada = InputChuva(entrada,dado_chuva);
    entrada = InputCurva_Continua(entrada,dado_coef_a,dado_coef_b,dado_coef_c);

    %Criando e inicializando o objeto
    curva =
hazard_continuos(entrada.coef_a,entrada.coef_b,entrada.coef_c,entrada.matrix_chuva,entrada.tamanho_area);
    curva = ProbRup(curva);

    PR = curva.Matrix_Pr;

    ctrol_per = 1;

    % Mensagem indicando fim das análises de PERIGO
    warndlg('Análise de ameaça estabelecida com sucesso!','Operação realizada');

% OPÇÃO DE NÍVEIS DE PROBABILIDADE DE RUPTURA
case 3
    % barra de status
    h = waitbar(0,'1','Name','Análise de Ameaça...','...
    'CreateCancelBtn',...
    'setappdata(gcf,'canceling',1)');
    setappdata(h,'canceling',0)

    steps = 6;
    waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))

    entrada = InputHazard(num_passos);
    waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))
    entrada = InputChuva(entrada,dado_chuva);
    waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))
    entrada = InputNivelAutomatico(entrada,dado_pos_nivel,dado_nivel,dado_nivel_pr);
    waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))
    nivel =
hazard_nivel(entrada.posicao_nivel,entrada.matrix_chuva,entrada.matrix_intervalos,entrada.tamanho_area,entr
ada.matrix_nivel_pr);
    waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))
    nivel = ProbRup_Nivel(nivel);
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
    PR = nivel.Matrix_Pr;

    delete(h)

    ctrol_per = 1;

    % Mensagem indicando fim das análises de PERIGO
    warndlg('Análise de ameaça estabelecida com sucesso!','Operação realizada');

% OPÇÃO DE IMPORTAÇÃO DIRETA DA GRADE
case 4

```

```

        PR = load(dado_grade);
        ctrol_per = 1;
        % Mensagem indicando fim das análises de PERIGO
        warndlg('Análise de ameaça estabelecida com sucesso!', 'Operação realizada');

    end % end switch

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
global dado_grade;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text5, 'String', fullpathname);

dado_grade = fullfile(pathname, filename);

function edit1_Callback(hObject, eventdata, handles)
global num_passos;

num_passos = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
global dado_coef_a;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text31, 'String', fullpathname);

dado_coef_a = fullfile(pathname, filename);

% --- Executes on button press in pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
global dado_coef_b;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text33, 'String', fullpathname);
dado_coef_b = fullfile(pathname, filename);

% --- Executes on button press in pushbutton17.
function pushbutton17_Callback(hObject, eventdata, handles)
global dado_coef_c;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text35, 'String', fullpathname);

dado_coef_c = fullfile(pathname, filename);

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
global dado_nivel_pr;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text29, 'String', fullpathname);

dado_nivel_pr = fullfile(pathname, filename);

% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)

```

```

global dado_nivel;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text27, 'String', fullpathname);

dado_nivel = fullfile(pathname, filename);

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
global dado_pos_nivel;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text25, 'String', fullpathname);

dado_pos_nivel = fullfile(pathname, filename);

% --- Executes on button press in pushbutton20.
function pushbutton20_Callback(hObject, eventdata, handles)
global dado_chuva;

folder_name = uigetdir;

% CÓDIGO PARA ABRIR SOMENTE O ARQUIVO
% [filename pathname] = uigetfile({'*.*'}, 'File Selector');
% fullpathname = strcat(pathname, filename);
% set(handles.text36, 'String', fullpathname);
%
% dado_chuva = fullfile(pathname, filename);

fullpathname = strcat(folder_name);
set(handles.text36, 'String', fullpathname);

dado_chuva = fullpathname;

% --- Executes on button press in pushbutton18.
function pushbutton18_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton19.
function pushbutton19_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton21.
function pushbutton21_Callback(hObject, eventdata, handles)
global dado_coef;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text40, 'String', fullpathname);

dado_coef = fullfile(pathname, filename);

% --- Executes on button press in pushbutton22.
function pushbutton22_Callback(hObject, eventdata, handles)
global dado_pos_curva;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text39, 'String', fullpathname);

dado_pos_curva = fullfile(pathname, filename);

% --- Executes when selected object is changed in uipanel11.
function uipanel11_SelectionChangeFcn(hObject, eventdata, handles)
global opcao;

% opção das curvas discretas
if hObject == handles.op1
    opcao = 1;
    % ligados
    set(handles.pushbutton20, 'enable', 'on');
    set(handles.pushbutton21, 'enable', 'on');
    set(handles.pushbutton22, 'enable', 'on');
    set(handles.edit1, 'enable', 'on');

    % desligados
    set(handles.pushbutton12, 'enable', 'off');
    set(handles.pushbutton13, 'enable', 'off');
    set(handles.pushbutton14, 'enable', 'off');
    set(handles.pushbutton15, 'enable', 'off');
    set(handles.pushbutton16, 'enable', 'off');
    set(handles.pushbutton17, 'enable', 'off');

    set(handles.pushbutton4, 'enable', 'off');

```

```

%limpar demais campos
set(handles.text31,'String','');
set(handles.text33,'String','');
set(handles.text35,'String','');

set(handles.text25,'String','');
set(handles.text27,'String','');
set(handles.text29,'String','');

set(handles.text5,'String','');

% opção das curvas contínuas
elseif hObject == handles.radiobutton10
    opcao = 2;
% ligados
set(handles.pushbutton20,'enable','on');
set(handles.pushbutton15,'enable','on');
set(handles.pushbutton16,'enable','on');
set(handles.pushbutton17,'enable','on');
set(handles.edit1,'enable','on');

% desligados
set(handles.pushbutton21,'enable','off');
set(handles.pushbutton22,'enable','off');
set(handles.pushbutton12,'enable','off');
set(handles.pushbutton13,'enable','off');
set(handles.pushbutton14,'enable','off');

set(handles.pushbutton4,'enable','off');

%limpar demais campos
set(handles.text40,'String','');
set(handles.text39,'String','');

set(handles.text25,'String','');
set(handles.text27,'String','');
set(handles.text29,'String','');

set(handles.text5,'String','');

% opção dos níveis de PR
elseif hObject == handles.radiobutton8
    opcao = 3;
% ligados
set(handles.pushbutton20,'enable','on');
set(handles.pushbutton12,'enable','on');
set(handles.pushbutton13,'enable','on');
set(handles.pushbutton14,'enable','on');
set(handles.edit1,'enable','on');

% desligados
set(handles.pushbutton21,'enable','off');
set(handles.pushbutton22,'enable','off');
set(handles.pushbutton15,'enable','off');
set(handles.pushbutton16,'enable','off');
set(handles.pushbutton17,'enable','off');

set(handles.pushbutton4,'enable','off');

%limpar demais campos
set(handles.text40,'String','');
set(handles.text39,'String','');
set(handles.text31,'String','');
set(handles.text33,'String','');
set(handles.text35,'String','');
set(handles.text5,'String','');

% opção de importação da grade
elseif hObject == handles.radiobutton7
    opcao = 4;
% ligados
set(handles.pushbutton4,'enable','on');
% desligados
set(handles.pushbutton12,'enable','off');
set(handles.pushbutton13,'enable','off');
set(handles.pushbutton14,'enable','off');
set(handles.pushbutton15,'enable','off');
set(handles.pushbutton16,'enable','off');
set(handles.pushbutton17,'enable','off');
set(handles.pushbutton20,'enable','off');
set(handles.pushbutton21,'enable','off');
set(handles.pushbutton22,'enable','off');
%limpar demais campos
set(handles.text31,'String','');
set(handles.text33,'String','');
set(handles.text35,'String','');
set(handles.text40,'String','');

```

```

set(handles.text39,'String','');
set(handles.text25,'String','');
set(handles.text27,'String','');
set(handles.text29,'String','');
set(handles.text36,'String','');
set(handles.edit1,'String','','enable','off');

end % end if

% --- Executes during object creation, after setting all properties.
function icone_curva_CreateFcn(hObject, eventdata, handles)

% Hint: place code in OpeningFcn to populate icone_curva
cur_fig = imread('imag_perigo.jpg');
image(cur_fig);
axis off

% --- Executes on button press in pushbutton23.
function pushbutton23_Callback(hObject, eventdata, handles)

```

A.2.2 CLASSES ESPECÍFICAS PARA ENTRADA DE DADOS E EXECUÇÃO DO MODELO

```

classdef InputHazard

    properties
        %Dados para importação de chuva
        coef_a;
        coef_b;
        coef_c;
        num_chuva;
        string_nome;
        string_complem;
        matrix_chuva;
        numero_arquivo;
        nome_arquivo;
        tamanho_area;
        %Dados referentes ao métodos das curvas
        coef_curva;
        posicao_curvas;
        %Dados referentes ao métodos dos níveis
        posicao_nivel;
        matrix_intervalos;
        matrix_nivel_pr;
        %Dados referentes ao método analítico
        unidade_chuva;
        delta_t;
        matrix_chuva_metros;
    end

    methods

        function obj = InputHazard(num_chuva)
            obj.num_chuva = num_chuva;
        end % end function

        %Método para importação dos dados de chuva
        function obj = InputChuva(obj,dado_chuva)
            obj.string_nome = 'input_chuva';
            obj.string_complem = '.txt';

            for i =1:obj.num_chuva
                obj.numero_arquivo = mat2str(i);
                obj.nome_arquivo =
strcat(dado_chuva,'\',obj.string_nome,obj.numero_arquivo,obj.string_complem);
                obj.matrix_chuva(:, :,i) = load(obj.nome_arquivo);
            end %end for

            obj.tamanho_area = size(obj.matrix_chuva);

            if obj.num_chuva == 1
                obj.tamanho_area(1,3) = 1;
            end %end if

        end % end function

        %Dados para o método que utiliza as curvas de PR
        function obj = InputCurva(obj,dado_coef,dado_pos_curva)

            obj.coef_curva = load(dado_coef);
            obj.posicao_curvas = load(dado_pos_curva);

        end % end function
    end
end

```

```

% Dados de entrada para a opção de curvas contínuas
function obj = InputCurva_Continua(obj,coef_a,coef_b,coef_c)

    obj.coef_a = load(coef_a);
    obj.coef_b = load(coef_b);
    obj.coef_c = load(coef_c);

end% end function

%Dados para o método que utiliza o cálculo do Fator de Segurança Chuva
%em horas ou dias
function obj = InputFS(obj,unidade_chuva)

    for i = 1:obj.num_chuva
        obj.matrix_chuva_metros(:, :, i) = obj.matrix_chuva(:, :, i)/1000;

    end %end for

    obj.unidade_chuva = unidade_chuva;

    switch unidade_chuva

        %Intensidade em horas
        case 1
            obj.delta_t = 3600;

        %Intensidade em dias
        case 2
            obj.delta_t = 86400;

    end %end switch
end %end function

% Função para o caso que se está utilizando uma precipitação
% genérica (por exemplo, W mm de chuva por X min)
function obj = InputFS_MIN(obj,unidade_chuva,tempo)

    for i = 1:obj.num_chuva
        obj.matrix_chuva_metros(:, :, i) = obj.matrix_chuva(:, :, i)/1000;

    end %end for

    obj.unidade_chuva = unidade_chuva;

    %Intensidade em min
    obj.delta_t = (tempo/60)*3600;
end %end function

function obj = InputNivelAutomatico(obj,dado_pos_nivel,dado_nivel,dado_nivel_pr)
    obj.posicao_nivel = load(dado_pos_nivel);
    obj.matrix_intervalos = load(dado_nivel);
    obj.matrix_nivel_pr = load(dado_nivel_pr);

end %end function
end % end methods
end %end class

classdef hazard
    %HAZARD Summary of this class goes here
    % Detailed explanation goes here

    properties (GetAccess = 'public',SetAccess = 'public')
        curva_atual;
        Matrix_Pr;
        Matrix_curva;
        Matrix_posicao;
        Matrix_chuva;
        tam;
    end

    methods
        %Inicializa o objeto
        function obj= hazard(posicao,chuva,matrix_curva,tam_matrix)

            obj.Matrix_posicao = posicao;

```

```

        obj.Matrix_chuva = chuva;
        obj.Matrix_curva = matrix_curva;
        obj.tam = tam_matrix;
    end

    % Probabilidade de ruptura para chuva acumulada
    function obj = ProbRup(obj)

        % barra de status
        h = waitbar(0,'1','Name','Análise de Perigo...',...
            'CreateCancelBtn',...
            'setappdata(gcf,'canceling',1)');
        setappdata(h,'canceling',0)

        steps = obj.tam(1)*obj.tam(3);

        for k = 1:obj.tam(3)
            for i = 1:obj.tam(1)
                for j = 1:obj.tam(2)

                    % carrega curva_atual com os coeficientes da célula atual
                    obj.Matrix_Pr(i,j,k) = obj.Matrix_curva(obj.Matrix_posicao(i,j),1)*exp(-
((obj.Matrix_chuva(i,j,k)-
obj.Matrix_curva(obj.Matrix_posicao(i,j),2))/obj.Matrix_curva(obj.Matrix_posicao(i,j),3))^2);

                    end %end for
                end %end for

                waitbar((i+(k-1)*obj.tam(1))/steps,h,sprintf('%0.f%% ',(i+(k-1)*obj.tam(1))/steps*100))

            end % end for

            delete(h)

        end %end function
    end % end methods
end %end class

classdef hazard_continuos
    %HAZARD Summary of this class goes here
    % Detailed explanation goes here

    properties (GetAccess = 'public',SetAccess = 'public')
        curva_atual;
        Matrix_Pr;
        Matrix_chuva;
        tam;
        coef_1;
        coef_2;
        coef_3;
    end

    methods
        %Inicializa o objeto
        function obj= hazard_continuos(coef_1,coef_2,coef_3,chuva,tam_matrix)

            obj.Matrix_chuva = chuva;
            obj.tam = tam_matrix;
            obj.coef_1 = coef_1;
            obj.coef_2 = coef_2;
            obj.coef_3 = coef_3;
        end

        % Probabilidade de ruptura para chuva acumulada
        function obj = ProbRup(obj)

            % barra de status
            h = waitbar(0,'1','Name','Análise de Ameaça...',...
                'CreateCancelBtn',...
                'setappdata(gcf,'canceling',1)');
            setappdata(h,'canceling',0)

            steps = obj.tam(3);

            for k = 1:obj.tam(3)

                obj.Matrix_Pr(:, :, k) = obj.coef_1.*exp(-((obj.Matrix_chuva(:, :, k)-
obj.coef_2)./obj.coef_3).^2);

                waitbar(k/steps,h,sprintf('%0.f%% ',k/steps*100))

            end % end for

            delete(h)

        end %end function
    end % end methods
end

```

```
end %end class
```

A.3 FUNÇÕES PARA MODELO SLIDE PROBABILÍSTICO

A.3.1 INTERFACE GRÁFICA

```
function varargout = MODELO_SLIDE(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @MODELO_SLIDE_OpeningFcn, ...
                  'gui_OutputFcn',  @MODELO_SLIDE_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before MODELO_SLIDE is made visible.
function MODELO_SLIDE_OpeningFcn(hObject, eventdata, handles, varargin)
global FS_crit;
global opcao_ch;
global opcao_cv;
global opcao_m;
global opcao_met;
global v;
global opcao_analise;

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

opcao_ch = 1;
opcao_cv = 1;
opcao_m = 1;
opcao_met = 1;
v = 1;
opcao_analise = 1;
FS_crit = 1;

% --- Outputs from this function are returned to the command line.
function varargout = MODELO_SLIDE_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

% --- Executes on button press in pushbutton19.
function pushbutton19_Callback(hObject, eventdata, handles)
% BOTÃO SAIR
delete(get(hObject, 'parent'));

% EXECUÇÃO DO PROGRAMA
% --- Executes on button press in pushbutton20.
function pushbutton20_Callback(hObject, eventdata, handles)

global ctrol_per;
global grade_cv_c;
global grade_cv_phi;
global grade_cv_kt;
global grade_cv_chuva;
global dado_Kt;
global dado_decl;
global dado_H;
global dado_phi;
global dado_Gs;
global dado_n;
global dado_S;
global dado_A;
global dado_lamb;
global dado_alph;
global dado_c;
global iterac;
global opcao_ch;
global opcao_cv;
global opcao_m;
global opcao_met;
```

```

global min;
global m_ini;
global dado_chuva;
global v;
global cv_c;
global cv_phi;
global cv_kt;
global cv_chuva;
global PR;
global opcao_analise;
global dado_quant;
global num_passos;
global FS_crit;

% BARRA DE PROGRESSO
h = waitbar(0, '1', 'Name', 'Análise pelo FOSM...', ...
           'CreateCancelBtn', ...
           'setappdata(gcf, 'canceling', 1)');
setappdata(h, 'canceling', 0)

% BARRA DE PROGRESSO
steps = 7;
waitbar(1/steps, h, sprintf('%0.f%% ', 1/steps*100))

entrada = InputHazard(num_passos);
entrada = InputChuva(entrada, dado_chuva);

% BARRA DE PROGRESSO
waitbar(2/steps, h, sprintf('%0.f%% ', 2/steps*100))

if opcao_ch ~= 3
    entrada = InputFS(entrada, opcao_ch);
else

    entrada = InputFS_MIN(entrada, opcao_ch, min);

end % end if

% BARRA DE PROGRESSO
waitbar(3/steps, h, sprintf('%0.f%% ', 3/steps*100))

% Valor inicial para m
if opcao_m == 2

    m_inicial = m_ini;

elseif opcao_m == 1

    m_inicial = 0;

end %end if

% BARRA DE PROGRESSO
waitbar(4/steps, h, sprintf('%0.f%% ', 4/steps*100))

switch opcao_met
% escolha do tipo de método probabilístico
% Método FOSM
case 1

    % BARRA DE PROGRESSO
    waitbar(5/steps, h, sprintf('%0.f%% ', 5/steps*100))

    %Inicializo o objeto
    fs =
SafetyFactorDerivada(entrada.tamanho_area, load(dado_Kt), load(dado_decl), load(dado_H), entrada.delta_t, entrad
a.matrix_chuva_metros, load(dado_phi), load(dado_Gs), load(dado_n), load(dado_S), load(dado_A), load(dado_lamb), l
oad(dado_alph), load(dado_c), m_inicial, opcao_analise);

    % escolha dos valores de Coeficiente de VAriação utilizados
    %VALORES ÚNICOS OU VALORES EM GRADE
    switch opcao_cv
        % valores únicos
        case 1

            % escolha entre valores DEFAULT ou PERSORNALIZADOS
            % DEFAULT
            if v == 1
                % BARRA DE PROGRESSO
                waitbar(6/steps, h, sprintf('%0.f%% ', 6/steps*100))

                fs = Fosm_default(fs);

            if opcao_analise ~= 3

                fs = ProbRup(fs, FS_crit);

```

```

else
    fs = ProbRupArquivo(fs,dado_quant,FS_crit);
end % end if

PR = fs.PR;

% BARRA DE PROGRESSO
waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
delete(h)

warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
% Controle para saber que análise de perigo foi realizada pelo menos 1 vez
ctrol_per = 1;
else
    % BARRA DE PROGRESSO
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
    fs = Fosm_Customizado(fs,load(cv_phi),load(cv_c),load(cv_chuva),load(cv_kt));

    if opcao_analise ~= 3
        fs = ProbRup(fs,FS_crit);
    else
        fs = ProbRupArquivo(fs,dado_quant,FS_crit);
    end % end if

    PR = fs.PR;

    % BARRA DE PROGRESSO
    waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
    delete(h)

    warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
    % Controle para saber que análise de perigo foi realizada
    % pelo menos 1 vez
    ctrol_per = 1;

end % end if

% valores EM MATRIZES
case 2

% escolha entre valores DEFAULT ou PERSONALIZADOS
% DEFAULT
if v == 1

    % BARRA DE PROGRESSO
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
    fs = Fosm_default_GRADE(fs,load(grade_cv_phi),load(grade_cv_c));

    if opcao_analise ~= 3

        fs = ProbRup(fs,FS_crit);

    else

        fs = ProbRupArquivo(fs,dado_quant,FS_crit);

    end % end if

    PR = fs.PR;

    % BARRA DE PROGRESSO
    waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
    delete(h)

    warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
    % Controle para saber que análise de perigo foi realizada
    % pelo menos 1 vez
    ctrol_per = 1;

else

    % BARRA DE PROGRESSO
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

    fs =
Fosm_Customizado(fs,load(grade_cv_phi),load(grade_cv_c),load(grade_cv_chuva),load(grade_cv_kt));

    if opcao_analise ~= 3

        fs = ProbRup(fs,FS_crit);

```

```

else
    fs = ProbRupArquivo(fs,dado_quant,FS_crit);
end % end if

PR = fs.PR;

% BARRA DE PROGRESSO
waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
delete(h)

warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
% Controle para saber que análise de perigo foi realizada
% pelo menos 1 vez
ctrol_per = 1;

end % end if
end % end switch opcao_cv

% Método de MONTE CARLO
case 2

    % BARRA DE PROGRESSO
    waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))

    %Inicializo o objeto
    fs =
MonteCarloFS(entrada.tamanho_area,load(dado_Kt),load(dado_decl),load(dado_H),entrada.delta_t,entrada.matrix
_chuva_metros,load(dado_phi),load(dado_Gs),load(dado_n),load(dado_S),load(dado_A),load(dado_lamb),load(dado
_alph),load(dado_c),m_inicial,opcao_analise);

    % escolha dos valores de Coeficiente de VARiação utilizados
    %VALORES ÚNICOS OU VALORES EM GRADE
    switch opcao_cv

        % valores únicos
        case 1

            % escolha entre valores DEFAULT ou PERSORNALIZADOS
            % DEFAULT
            if v == 1

                % BARRA DE PROGRESSO
                waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

                % Valores de CV iguais para toda a área de
                % estudo
                % Coesão: CV = 40%
                % Atrito: CV = 10%
                matrix_atrito_cv = 10*ones(fs.tam(1),fs.tam(2));
                matrix_coesao_cv = 40*ones(fs.tam(1),fs.tam(2));

                fs = MonteCarlo_default(fs,matrix_atrito_cv,matrix_coesao_cv,iterac);

                if opcao_analise ~= 3

                    fs = ProbRup(fs,FS_crit);

                else
                    fs = ProbRupArquivo(fs,dado_quant,FS_crit);
                end % end if

                PR = fs.PR;

                % BARRA DE PROGRESSO
                waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
                delete(h)

                warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
                % Controle para saber que análise de perigo foi realizada
                % pelo menos 1 vez
                ctrol_per = 1;

            % Valores PERSONALIZADOS
            else

                % BARRA DE PROGRESSO
                waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

                fs =
MonteCarlo_person(fs,cv_phi*ones(fs.tam(1),fs.tam(2)),cv_c*ones(fs.tam(1),fs.tam(2)),cv_kt*ones(fs.tam(1),f
s.tam(2)),cv_chuva*ones(fs.tam(1),fs.tam(2)),iterac);

                if opcao_analise ~= 3

                    fs = ProbRup(fs,FS_crit);

```

```

else
    % verificar na implementação esta FUNÇÃO
    % NÃO ESTÁ IMPLEMENTADA!!!!
    fs = ProbRupArquivo(fs,dado_quant,FS_crit);
end % end if

PR = fs.PR;

% BARRA DE PROGRESSO
waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
delete(h)

warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
% Controle para saber que análise de perigo foi realizada
% pelo menos 1 vez
ctrol_per = 1;

end % end if

% valores EM MATRIZES
case 2

% escolha entre valores DEFAULT ou PERSONALIZADOS
% DEFAULT
if v == 1
    % BARRA DE PROGRESSO
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

    fs = MonteCarlo_default(fs,load(grade_cv_phi),load(grade_cv_c),iterac);

    if opcao_analise ~= 3

        fs = ProbRup(fs,FS_crit);

    else
        fs = ProbRupArquivo(fs,dado_quant,FS_crit);
    end % end if

    PR = fs.PR;

    % BARRA DE PROGRESSO
    waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
    delete(h)

    warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
    % Controle para saber que análise de perigo foi realizada
    % pelo menos 1 vez
    ctrol_per = 1;

% PERSONALIZADO
else

    % BARRA DE PROGRESSO
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

    fs =
MonteCarlo_person(fs,load(grade_cv_phi),load(grade_cv_c),load(grade_cv_kt),load(grade_cv_chuva),iterac);

    if opcao_analise ~= 3

        fs = ProbRup(fs,FS_crit);

    else
        % VERIFICAR ESTA FUNÇÃO
        fs = ProbRupArquivo(fs,dado_quant,FS_crit);

    end % end if

    PR = fs.PR;

    % BARRA DE PROGRESSO
    waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
    delete(h)

    warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
    % Controle para saber que análise de perigo foi realizada
    % pelo menos 1 vez
    ctrol_per = 1;

end % end if

end % end opcao_cv

% Método de Rosenblueth
% ESTIMATIVAS PUNUAIS
case 3

```

```

% BARRA DE PROGRESSO
waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))

%Inicializo o objeto
fs =
SafetyFactorDerivada(entrada.tamanho_area,load(dado_Kt),load(dado_decl),load(dado_H),entrada.delta_t,entrad
a.matrix_chuva_metros,load(dado_phi),load(dado_Gs),load(dado_n),load(dado_S),load(dado_A),load(dado_lamb),l
oad(dado_alph),load(dado_c),m_inicial,opcao_analise);

% escolha dos valores de Coeficiente de VARiação utilizados
%VALORES ÚNICOS OU VALORES EM GRADE
switch opcao_cv

% valores únicos
case 1

% escolha entre valores DEFAULT ou PERSONALIZADOS
% DEFAULT
if v == 1

% BARRA DE PROGRESSO
waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

fs = MEP_default(fs,10,40);

% PERSONALIZADOS
else
% BARRA DE PROGRESSO
waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
fs = MEP_persona(fs,load(cv_phi),load(cv_c),load(cv_chuva),load(cv_kt));

end % end if v == 1

% opções para cálculo das probabilidades de
% ruptura
if opcao_analise ~= 3

fs = ProbRup(fs,FS_crit);

else

fs = ProbRupArquivo(fs,dado_quant,FS_crit);

end % end if

PR = fs.PR;

% BARRA DE PROGRESSO
waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
delete(h)

warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
% Controle para saber que análise de perigo foi realizada pelo menos 1 vez
ctrol_per = 1;

% valores EM MATRIZES
case 2

% escolha entre valores DEFAULT ou PERSONALIZADOS
% DEFAULT
if v == 1

% BARRA DE PROGRESSO
waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

fs = MEP_default(fs,load(grade_cv_phi),load(grade_cv_c));

else
% BARRA DE PROGRESSO
waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

fs =
MEP_persona(fs,load(grade_cv_phi),load(grade_cv_c),load(grade_cv_chuva),load(grade_cv_kt));

end % end if v == 1

% opções de cálculo de probabilidade de ruptura
if opcao_analise ~= 3

fs = ProbRup(fs,FS_crit);

else

fs = ProbRupArquivo(fs,dado_quant,FS_crit);

end % end if

PR = fs.PR;

```

```

        % BARRA DE PROGRESSO
        waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
        delete(h)

        warndlg('Análise do perigo estabelecido com sucesso!','Operação realizada');
        % Controle para saber que análise de perigo foi realizada pelo menos 1 vez
        ctrol_per = 1;

        end % end switch
    end % end switch opcao_met

% --- Executes on button press in radiobutton8.
function radiobutton8_Callback(hObject, eventdata, handles)

% --- Executes on button press in radiobutton9.
function radiobutton9_Callback(hObject, eventdata, handles)

function edit9_Callback(hObject, eventdata, handles)
global m_ini;

m_ini = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit1_Callback(hObject, eventdata, handles)
global num_passos;

num_passos = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
global dado_chuva;

folder_name = uigetdir;

fullpathname = strcat(folder_name);
set(handles.text43,'String',fullpathname);

dado_chuva = fullpathname;

function edit5_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
global min;
min = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global dado_Kt;

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text26,'String',fullpathname);

dado_Kt = fullfile(pathname,filename);

```

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global dado_decl;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text27, 'String', fullpathname);

dado_decl = fullfile(pathname, filename);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
global dado_H;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text28, 'String', fullpathname);

dado_H = fullfile(pathname, filename);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
global dado_phi;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text29, 'String', fullpathname);

dado_phi = fullfile(pathname, filename);

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
global dado_Gs;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text30, 'String', fullpathname);

dado_Gs = fullfile(pathname, filename);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
global dado_n;
[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text31, 'String', fullpathname);

dado_n = fullfile(pathname, filename);

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
global dado_S;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text32, 'String', fullpathname);

dado_S = fullfile(pathname, filename);

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
global dado_A;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text33, 'String', fullpathname);
dado_A = fullfile(pathname, filename);

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
global dado_lamb;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text34, 'String', fullpathname);

dado_lamb = fullfile(pathname, filename);

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
global dado_alph;

[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text35, 'String', fullpathname);

```

```

dado_alph = fullfile(pathname, filename);

% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
global dado_c;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text36, 'String', fullpathname);

dado_c = fullfile(pathname, filename);

% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
global grade_cv_c;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text55, 'String', fullpathname);

grade_cv_c = fullfile(pathname, filename);

% --- Executes on button press in pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
global grade_cv_phi;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text57, 'String', fullpathname);

grade_cv_phi = fullfile(pathname, filename);

% --- Executes on button press in pushbutton17.
function pushbutton17_Callback(hObject, eventdata, handles)
global grade_cv_kt;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text59, 'String', fullpathname);

grade_cv_kt = fullfile(pathname, filename);

% --- Executes on button press in pushbutton18.
function pushbutton18_Callback(hObject, eventdata, handles)
global grade_cv_chuva;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text61, 'String', fullpathname);

grade_cv_chuva = fullfile(pathname, filename);

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
global opcao_cv;
global v;

v = get(handles.popupmenu1, 'Value');

if opcao_cv == 1
    if v == 1
        set(handles.edit2, 'enable', 'on', 'String', '40');
        set(handles.edit3, 'enable', 'on', 'String', '10');
        set(handles.edit7, 'enable', 'off', 'String', '');
        set(handles.edit8, 'enable', 'off', 'String', '');

        set(handles.pushbutton15, 'enable', 'off');
        set(handles.pushbutton16, 'enable', 'off');
        set(handles.pushbutton17, 'enable', 'off');
        set(handles.pushbutton18, 'enable', 'off');

    elseif v == 2
        set(handles.edit2, 'enable', 'on', 'String', '');
        set(handles.edit3, 'enable', 'on', 'String', '');
        set(handles.edit7, 'enable', 'on', 'String', '');
        set(handles.edit8, 'enable', 'on', 'String', '');

        set(handles.pushbutton15, 'enable', 'off');
        set(handles.pushbutton16, 'enable', 'off');
        set(handles.pushbutton17, 'enable', 'off');
        set(handles.pushbutton18, 'enable', 'off');

    end % end if
elseif opcao_cv == 2

```

```

if v == 1
    set(handles.edit2, 'enable', 'off');
    set(handles.edit3, 'enable', 'off');
    set(handles.edit7, 'enable', 'off');
    set(handles.edit8, 'enable', 'off');

    set(handles.pushbutton15, 'enable', 'on');
    set(handles.pushbutton16, 'enable', 'on');
    set(handles.pushbutton17, 'enable', 'off');
    set(handles.pushbutton18, 'enable', 'off');

elseif v == 2
    set(handles.edit2, 'enable', 'off');
    set(handles.edit3, 'enable', 'off');
    set(handles.edit7, 'enable', 'off');
    set(handles.edit8, 'enable', 'off');

    set(handles.pushbutton15, 'enable', 'on');
    set(handles.pushbutton16, 'enable', 'on');
    set(handles.pushbutton17, 'enable', 'on');
    set(handles.pushbutton18, 'enable', 'on');

end % end if
end % end if

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
global cv_c;
cv_c = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
global cv_phi;
cv_phi = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
global cv_kt;

cv_kt = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
global cv_chuva;

cv_chuva = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes when selected object is changed in uipanel10.
function uipanel10_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_ch;

```

```

% opção das chuvas em mm/h
if hObject == handles.radiobutton11
    opcao_ch = 1;
    set(handles.edit6, 'enable', 'off');
    % opção das chuvas em mm/dia
elseif hObject == handles.radiobutton12
    opcao_ch = 2;
    set(handles.edit6, 'enable', 'off');
% opção das chuvas em mm por X min
elseif hObject == handles.radiobutton13
    opcao_ch = 3;
    set(handles.edit6, 'enable', 'on');

end %end if

% --- Executes during object creation, after setting all properties.
function uipanel10_CreateFcn(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function uipanel15_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel5.
function uipanel15_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_cv;

if hObject == handles.radiobutton3
    opcao_cv = 1;

    set(handles.pushbutton15, 'enable', 'off');
    set(handles.pushbutton16, 'enable', 'off');
    set(handles.pushbutton17, 'enable', 'off');
    set(handles.pushbutton18, 'enable', 'off');

elseif hObject == handles.radiobutton4
    opcao_cv = 2;

    set(handles.edit2, 'enable', 'off');
    set(handles.edit3, 'enable', 'off');
    set(handles.edit7, 'enable', 'off');
    set(handles.edit8, 'enable', 'off');

end % end if

% --- Executes during object creation, after setting all properties.
function uipanel11_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel11.
function uipanel11_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_m;

% opção do valor de m inicial
if hObject == handles.radiobutton9
    opcao_m = 1;
    set(handles.edit9, 'enable', 'off');

elseif hObject == handles.radiobutton8
    opcao_m = 2;
    set(handles.edit9, 'enable', 'on');

end %end if

% --- Executes during object creation, after setting all properties.
function uipanel4_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel4.
function uipanel4_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_met;

% opção do método a ser utilizado
if hObject == handles.radiobutton1
    opcao_met = 1;
    set(handles.edit10, 'enable', 'off');

elseif hObject == handles.radiobutton2
    opcao_met = 2;
    set(handles.edit10, 'enable', 'on');

elseif hObject == handles.radiobutton20
    opcao_met = 3;
    set(handles.edit10, 'enable', 'off');

```

```

end %end if

function edit10_Callback(hObject, eventdata, handles)
global iterac;

iterac = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in pushbutton21.
function pushbutton21_Callback(hObject, eventdata, handles)
global dado_quant;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text68, 'String', fullpathname);

dado_quant = fullfile(pathname, filename);

% --- Executes during object creation, after setting all properties.
function uipanel13_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel13.
function uipanel13_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_analise;

if hObject == handles.radiobutton23
    opcao_analise = 1;
    set(handles.pushbutton21, 'enable', 'off');

elseif hObject == handles.radiobutton22
    opcao_analise = 2;
    set(handles.pushbutton21, 'enable', 'off');

elseif hObject == handles.radiobutton21
    opcao_analise = 3;
    set(handles.pushbutton21, 'enable', 'on');

end %end if

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)

slid = imread('imag_perigo.jpg');
image(slid);
axis off

function edit11_Callback(hObject, eventdata, handles)
global FS_crit;

FS_crit = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

A.3.2 CLASSES ESPECÍFICAS PARA EXECUÇÃO DO MODELO

```

classdef SafetyFactorDerivada

    properties

        % Propriedades da área de estudo
        tam;
        % Propriedades do talude
        %Inclinação do talude(em radianos - SPRING)
        beta;
        H;
        % Propriedades do solo

```

```

atrito;
coesao_ef;
A;
Lambda;
alpha;
Gs;
n;
nw;
Sr;
Kt;
% Funções para o cálculo do FS
Coesao_Total;
Gama;
Omega;
FS_med;
% Propriedades da chuva
% Intensidade da chuva em m (metros)
It;
% Intervalo de tempo entre os passos em s(segundos)
delta_t;
m;
delta_m;
Ot;
m_inicial;
m_acum;
m_maior_1;
m_menor_1;
INF;
NINF;
%Propriedades para o cálculo da PR (Aplicação do FOSM)
%coeficientes de variação
cv_atrito;
cv_coesao_ef;
cv_It;
cv_Kt;
%matriz de variância (matriz multidimensional)
% Ordem das variâncias salvas: Atrito,coesão, chuva, Kt
variancia;
variancia_It;
%matrizes dos incrementos (deltas) das variáveis Xi's
delta_Xi;
delta_It;
%matrizes dos valores médios + incrementos das variáveis Xi's
Incr_Xi;
Incr_It;
%matriz dos fatores de segurança calculados devido ao incrementos
%nos Xi's
FS_incr_coesao_ef;
FS_incr_Kt;
FS_incr_It;
FS_incr_Atrito;
% FS incemento - FS_médio
delta_FS_atrito;
delta_FS_coesao_ef;
delta_FS_It;
delta_FS_Kt;
%Desvio Padrão do FS
% Armazena as variâncias finais das variáveis aleatórias que contribuem na formação do desvio padrão
do FS final
Varianc_Final;
desvio_FS;
Zcrit;
aux;
PR;
Ind_Conf;
cont;
teste;

quant_analis;
arq_analis;
cont_pr;

% Método das Estimativas Pontuais
vet_coesao;
vet_atrito;
vet_Kt;
vet_It;
soma_FS;
variancia_FS;
F;
end

properties(Constant = true)
Peso_esp_w = 9800;
delta_s = 1;
end

```

methods

```

% Inicialização do objeto (Para o FOSM Default)
function obj =
SafetyFactorDerivada(tamanho_area,matrix_Kt,matrix_beta,matrix_H,tempo,matrix_chuva,atrito,Gs,n,Sr,A,lambda
,alpha,coesao_ef,m_inicial,quant_analis)
obj.tam = tamanho_area;
obj.delta_t = tempo;
obj.Kt = matrix_Kt;
obj.beta = pi*(matrix_beta)/180;
obj.H = matrix_H;
obj.It = matrix_chuva;
obj.atrito = pi*(atrito)/180;
obj.Gs = Gs;
obj.n = n;
obj.Sr = Sr/100;
obj.A = A;
obj.Lambda = lambda;
obj.alpha = alpha;
obj.coesao_ef = coesao_ef*1000;
obj.m_inicial = m_inicial;
obj.quant_analis = quant_analis;

end %end function

% FUNÇÃO FOSM PADRÃO
% Implementação do método FOSM utilizando variação apenas da coesão
% efetiva e do atrito
function obj = Fosm_default(obj)

%definição dos coeficientes de variação default
obj.cv_atrito = 10;
obj.cv_coesao_ef = 40;

%Variância das variáveis Xi's

obj.variancia(:, :, 1) = (obj.cv_atrito.*obj.atrito./100).^2;
obj.variancia(:, :, 2) = (obj.cv_coesao_ef.*obj.coesao_ef./100).^2;

% Valor inicial de m pode ser zero ou não
obj.m(:, :) = obj.m_inicial*ones(obj.tam(1),obj.tam(2));

%Matriz com os valores de m acumulado
obj.m_acum = zeros(obj.tam(1),obj.tam(2),obj.tam(3));

%Definição das funções Omega,Gama e nw para a área de estudo

obj.Gama = obj.Gs.*(1-obj.n)+obj.n.*obj.Sr;
obj.nw = obj.n.*(1-obj.Sr);
obj.Omega = 2./(sin(2.*obj.beta).*obj.H.*obj.Peso_esp_w);

% Cálculo do valor de m para os valores médios e incrementos
% das variáveis Xi's
for k = 1:obj.tam(3)

% Cálculo do valor médio de m (índice 1)
obj.Ot= obj.Kt.*sin(obj.beta).*obj.m.*obj.H.*cos(obj.beta).*obj.delta_t;
obj.delta_m = (obj.It(:, :, k)-obj.Ot)./(obj.n.*obj.H.*(1-obj.Sr));
obj.m = obj.m+obj.delta_m;

%RESTRICÃO DO VALOR DE m

obj.m_maior_1 = obj.m >= 1;
obj.m_menor_1 = obj.m < 1;
obj.m = obj.m_maior_1 + obj.m_menor_1.*obj.m;

% Cálculo do FS médio
obj.Coesao_Total(:, :, 1) = (obj.coesao_ef+obj.A.*(1-
obj.Lambda.*obj.m.^obj.alpha)).*obj.delta_s;
obj.FS_med(:, :, k) = (cot(obj.beta).*tan(obj.atrito).*(obj.Gama+obj.m.*(obj.nw-
1))+obj.Coesao_Total(:, :, 1).*obj.Omega)./(obj.Gama+obj.m.*obj.nw);

% Cálculo do desvio padrão para FS referente a
% variação da coesão efetiva e atrito
%Parcela da variância do FS relativa ao atrito (índice 1)
obj.Varianc_Final(:, :, 1, k) = (((cot(obj.beta).*(obj.Gama+obj.m.*(obj.nw-
1)).*(tan(obj.atrito)).^2+1))./(obj.Gama+obj.m.*obj.nw)).^2.*obj.variancia(:, :, 1));

%Parcela da variância do FS relativa à coesão (índice 2)
obj.Varianc_Final(:, :, 2, k) = ((obj.Omega./(obj.Gama+obj.m.*(obj.nw-
1))).^2.*obj.variancia(:, :, 2));

obj.desvio_FS(:, :, k) = sqrt(obj.Varianc_Final(:, :, 1, k)+obj.Varianc_Final(:, :, 2, k));

obj.m_acum(:, :, k) = obj.m(:, :);

end %end for

```

```

end % end function

% FUNÇÃO FOSM PADRÃO PARA ENTRADA DE CV COMO MATRIZES
% Implementação do método FOSM utilizando variação apenas da coesão
% efetiva e do atrito
function obj = Fosm_default_GRADE(obj,cv_atrito,cv_coesao)

    obj.cv_atrito = cv_atrito;
    obj.cv_coesao_ef = cv_coesao;

    %Variância das variáveis Xi's

    obj.variancia(:, :, 1) = (obj.cv_atrito.*obj.atrito./100).^2;
    obj.variancia(:, :, 2) = (obj.cv_coesao_ef.*obj.coesao_ef./100).^2;

    % Valor inicial de m pode ser zero ou não
    obj.m(:, :) = obj.m_inicial*ones(obj.tam(1),obj.tam(2));

    % Matriz com os valores de m acumulado
    obj.m_acum = zeros(obj.tam(1),obj.tam(2),obj.tam(3));

    %Definição das funções Omega,Gama e nw para a área de estudo

    obj.Gama = obj.Gs.*(1-obj.n)+obj.n.*obj.Sr;
    obj.nw = obj.n.*(1-obj.Sr);
    obj.Omega = 2./(sin(2.*obj.beta).*obj.H.*obj.Peso_esp_w);

    % Cálculo do valor de m para os valores médios e incrementos
    % das variáveis Xi's
    for k = 1:obj.tam(3)

        % Cálculo do valor médio de m (índice 1)
        obj.Ot= obj.Kt.*sin(obj.beta).*obj.m.*obj.H.*cos(obj.beta).*obj.delta_t;
        obj.delta_m = (obj.It(:, :, k)-obj.Ot)./(obj.n.*obj.H.*(1-obj.Sr));
        obj.m = obj.m+obj.delta_m;

        %RESTRICÃO DO VALOR DE m

        obj.m_maior_1 = obj.m >= 1;
        obj.m_menor_1 = obj.m < 1;
        obj.m = obj.m_maior_1 + obj.m_menor_1.*obj.m;

        % Cálculo do FS médio
        obj.Coesao_Total(:, :, 1) = (obj.coesao_ef+obj.A.*(1-
obj.Lambda.*obj.m.^obj.alpha)).*obj.delta_s;
        obj.FS_med(:, :, k) = (cot(obj.beta).*tan(obj.atrito).*(obj.Gama+obj.m.*(obj.nw-
1))+obj.Coesao_Total(:, :, 1).*obj.Omega)./(obj.Gama+obj.m.*obj.nw);

        % Cálculo do desvio padrão para FS referente a
        % variação da coesão efetiva e atrito
        %Parcela da variância do FS relativa ao atrito (índice 1)
        obj.Varianc_Final(:, :, 1, k) = ((cot(obj.beta).*(obj.Gama+obj.m.*(obj.nw-
1)).*(tan(obj.atrito)).^2+1))./(obj.Gama+obj.m.*obj.nw).^2.*obj.variancia(:, :, 1));

        %Parcela da variância do FS relativa à coesão (índice 2)
        obj.Varianc_Final(:, :, 2, k) = ((obj.Omega./(obj.Gama+obj.m.*(obj.nw-
1))).^2.*obj.variancia(:, :, 2));

        obj.desvio_FS(:, :, k) = sqrt(obj.Varianc_Final(:, :, 1, k)+obj.Varianc_Final(:, :, 2, k));

        obj.m_acum(:, :, k) = obj.m(:, :);

    end %end for
end % end function

% FUNÇÃO PARA CÁLCULO DA PR E DO ÍNDICE DE CONFIABILIDADE
% Função que calcula a PR a partir dos valores do FS médio e do
% desvio-padrão

function obj = ProbRup(obj,FS_crit)

    obj.Zcrit = zeros(obj.tam(1),obj.tam(2));
    obj.INF = obj.FS_med == Inf;
    obj.NINF = obj.FS_med ~= Inf;

    % Para todos os passos de chuva
    if obj.quant_analis == 1

        % clear obj.PR;
        obj.PR = zeros(obj.tam(1),obj.tam(2),obj.tam(3));

        for k = 1:obj.tam(3)

```

```

obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.desvio_FS(:,:,k);

for i = 1:obj.tam(1)
    for j = 1:obj.tam(2)

        if obj.FS_med(i,j,k) ~= Inf

            x = -10:0.01:obj.Zcrit(i,j);

            Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

            obj.PR(i,j,k) = trapz(x,Y);

        end % end if
    end % end for
end % end for

% para o primeiro e o último passo
elseif obj.quant_analis == 2
    obj.PR = zeros(obj.tam(1),obj.tam(2),2);
    obj.cont = 1;

    for k = 1:obj.tam(3)-1:obj.tam(3)

        obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.desvio_FS(:,:,k);

        for i = 1:obj.tam(1)
            for j = 1:obj.tam(2)

                if obj.FS_med(i,j,k) ~= Inf

                    x = -10:0.01:obj.Zcrit(i,j);

                    Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

                    obj.PR(i,j,obj.cont) = trapz(x,Y);

                end % end if

            end % end for
        end % end for

        obj.cont = obj.cont+1;

    end %end for
end % end if
end %end function

% Função para o CÁLCULO DA PR PARA PASSOS DEFINIDOS POR ARQUIVO
function obj = ProbRupArquivo(obj,arq_analis,FS_crit)

obj.arq_analis = arq_analis;

obj.PR = zeros(obj.tam(1),obj.tam(2),size(obj.arq_analis));
obj.cont = 1;

while obj.cont < size(obj.arq_analis)+1

    k = obj.arq_analis(obj.cont);

    obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.desvio_FS(:,:,k);

    for i = 1:obj.tam(1)
        for j = 1:obj.tam(2)

            if obj.FS_med(i,j,k) ~= Inf

                x = -10:0.01:obj.Zcrit(i,j);

                Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

                obj.PR(i,j,obj.cont) = trapz(x,Y);

            end % end if
        end % end for
    end % end for

    obj.cont = obj.cont+1;
end % end while
end % end function

% FUNÇÃO PARA O MÉTODO FOSM QUE O USUÁRIO FORNECE OS CV'S
function obj = Fosm_Customizado(obj,cv_atrito,cv_coexao_ef,cv_chuva,cv_Kt)

```

```

%definição do coeficientes de variação
obj.cv_atrito = cv_atrito;
obj.cv_coesao_ef = cv_coesao_ef;
obj.cv_It = cv_chuva;
obj.cv_Kt = cv_Kt;

% Ordem das variáveis na disposição da matriz:atrito,coesão,Kt,It
%Aumento de 10% nos valores médios das variáveis Xi
obj.Incr_Xi(:, :, 1) = 1.1*obj.atrito;
obj.Incr_Xi(:, :, 2) = 1.1*obj.coesao_ef;
obj.Incr_Xi(:, :, 3) = 1.1*obj.Kt;
obj.Incr_It = 1.1*obj.It;

% Delta para as variáveis utilizadas
obj.delta_Xi(:, :, 1) = obj.Incr_Xi(:, :, 1) - obj.atrito;
obj.delta_Xi(:, :, 2) = obj.Incr_Xi(:, :, 2) - obj.coesao_ef;
obj.delta_Xi(:, :, 3) = obj.Incr_Xi(:, :, 3) - obj.Kt;
obj.delta_It = obj.Incr_It - obj.It;

%Variância das variáveis Xi's
obj.variancia(:, :, 1) = (obj.cv_atrito.*obj.atrito./100).^2;
obj.variancia(:, :, 2) = (obj.cv_coesao_ef.*obj.coesao_ef./100).^2;
obj.variancia(:, :, 3) = (obj.cv_Kt.*obj.Kt./100).^2;

for k = 1:obj.tam(3)
    obj.variancia_It(:, :, k) = (obj.cv_It.*obj.It(:, :, k)./100).^2;
end % end for

%Matriz com os valores de m iniciais (nulos)
obj.m = zeros(obj.tam(1),obj.tam(2),3);

%Definição das funções Omega,Gama e nw para a área de estudo

obj.Gama = obj.Gs.*(1-obj.n)+obj.n.*obj.Sr;
obj.nw = obj.n.*(1-obj.Sr);
obj.Omega = 2./(sin(2*obj.beta).*obj.H.*obj.Peso_esp_w);

% Cálculo do valor de m para os valores médios e incrementos das variáveis Xi's
for k = 1:obj.tam(3)

    % Cálculo do valor médio de m (índice 1)
    obj.Ot(:, :, 1) =
obj.Kt.*sin(obj.beta).*obj.m(:, :, 1).*obj.H.*cos(obj.beta).*obj.delta_t;
    obj.delta_m(:, :, 1) = (obj.It(:, :, k) - obj.Ot(:, :, 1)) ./ (obj.n.*obj.H.*(1-obj.Sr));
    obj.m(:, :, 1) = obj.m(:, :, 1) + obj.delta_m(:, :, 1);

    %RESTRICÃO DO VALOR DE m

    obj.m_maior_1 = obj.m(:, :, 1) >= 1;
    obj.m_menor_1 = obj.m(:, :, 1) < 1;
    obj.m(:, :, 1) = obj.m_maior_1 + obj.m_menor_1.*obj.m(:, :, 1);

    % Valor de m devido ao incremento de Kt (índice 2)
    obj.Ot(:, :, 2) =
obj.Incr_Xi(:, :, 3).*sin(obj.beta).*obj.m(:, :, 2).*obj.H.*cos(obj.beta).*obj.delta_t;
    obj.delta_m(:, :, 2) = (obj.It(:, :, k) - obj.Ot(:, :, 2)) ./ (obj.n.*obj.H.*(1-obj.Sr));
    obj.m(:, :, 2) = obj.m(:, :, 2) + obj.delta_m(:, :, 2);

    %RESTRICÃO DO VALOR DE m

    obj.m_maior_1 = obj.m(:, :, 2) > 1;
    obj.m_menor_1 = obj.m(:, :, 2) < 1;
    obj.m(:, :, 2) = obj.m_maior_1 + obj.m_menor_1.*obj.m(:, :, 2);

    % Valor de m devido ao incremento das chuvas (índice 3)
    obj.Ot(:, :, 3) =
obj.Kt.*sin(obj.beta).*obj.m(:, :, 3).*obj.H.*cos(obj.beta).*obj.delta_t;
    obj.delta_m(:, :, 3) = (obj.Incr_It(:, :, k) - obj.Ot(:, :, 3)) ./ (obj.n.*obj.H.*(1-obj.Sr));
    obj.m(:, :, 3) = obj.m(:, :, 3) + obj.delta_m(:, :, 3);

    %RESTRICÃO DO VALOR DE m

    obj.m_maior_1 = obj.m(:, :, 3) > 1;
    obj.m_menor_1 = obj.m(:, :, 3) < 1;
    obj.m(:, :, 3) = obj.m_maior_1 + obj.m_menor_1.*obj.m(:, :, 3);

    %Cálculo do valor de coesão total para valores
    %médios (índice 1)
    obj.Coesao_Total(:, :, 1) = (obj.coesao_ef + obj.A*(1-
obj.Lambda.*obj.m(:, :, 1).^obj.alpha))*obj.delta_s;
    obj.FS_med(:, :, k) = (cot(obj.beta).*tan(obj.atrito).*(obj.Gama + obj.m(:, :, 1).*(obj.nw-
1)) + obj.Coesao_Total(:, :, 1).*obj.Omega) ./ (obj.Gama + obj.m(:, :, 1).*(obj.nw));

    %Cálculo do FS para o incremento em Kt

```

```

        obj.Coesao_Total(:, :, 2) = (obj.coesao_ef+obj.A.*(1-
obj.Lambda.*obj.m(:, :, 2).^obj.alpha)).*obj.delta_s;
        obj.FS_incr_Kt(:, :, k) =
(cot(obj.beta).*tan(obj.atrito).*obj.Gama+obj.m(:, :, 2).*(obj.nw-
1))+obj.Coesao_Total(:, :, 2).*obj.Omega./(obj.Gama+obj.m(:, :, 2).*obj.nw);

        %Cálculo do FS para o incremento nas chuvas
        obj.Coesao_Total(:, :, 3) = (obj.coesao_ef+obj.A.*(1-
obj.Lambda.*obj.m(:, :, 3).^obj.alpha)).*obj.delta_s;
        obj.FS_incr_It(:, :, k) =
(cot(obj.beta).*tan(obj.atrito).*obj.Gama+obj.m(:, :, 3).*(obj.nw-
1))+obj.Coesao_Total(:, :, 3).*obj.Omega./(obj.Gama+obj.m(:, :, 3).*obj.nw);

        %Cálculo do FS para o incremento do atrito
        obj.Coesao_Total(:, :, 1) = (obj.coesao_ef+obj.A.*(1-
obj.Lambda.*obj.m(:, :, 1).^obj.alpha)).*obj.delta_s;
        obj.FS_incr_Atrito(:, :, k) =
(cot(obj.beta).*tan(obj.Incr_Xi(:, :, 1)).*(obj.Gama+obj.m(:, :, 1).*(obj.nw-
1))+obj.Coesao_Total(:, :, 3).*obj.Omega./(obj.Gama+obj.m(:, :, 1).*obj.nw);

        %Cálculo do FS para o incremento da coesão efetiva
        obj.Coesao_Total(:, :, 4) = (obj.Incr_Xi(:, :, 2)+obj.A.*(1-
obj.Lambda.*obj.m(:, :, 1).^obj.alpha)).*obj.delta_s;
        obj.FS_incr_coesao_ef(:, :, k) =
(cot(obj.beta).*tan(obj.atrito(:, :, 1)).*(obj.Gama+obj.m(:, :, 1).*(obj.nw-
1))+obj.Coesao_Total(:, :, 4).*obj.Omega./(obj.Gama+obj.m(:, :, 1).*obj.nw);

    end % end for

    %Cálculo da variação dos FS (delta_FS)
    obj.delta_FS_atrito = obj.FS_incr_Atrito - obj.FS_med;
    obj.delta_FS_coesao_ef = obj.FS_incr_coesao_ef - obj.FS_med;
    obj.delta_FS_It = obj.FS_incr_It - obj.FS_med;
    obj.delta_FS_Kt = obj.FS_incr_Kt - obj.FS_med;

    for k = 1:obj.tam(3)

        obj.desvio_FS(:, :, k) =
sqrt(((obj.delta_FS_atrito(:, :, k)./obj.delta_Xi(:, :, 1)).^2.*obj.variancia(:, :, 1))+((obj.delta_FS_coesao_ef(
(:, :, k)./obj.delta_Xi(:, :, 2)).^2.*obj.variancia(:, :, 2))+((obj.delta_FS_It(:, :, k)./obj.delta_It(:, :, k)).^2.*o
bj.variancia_It(:, :, k))+((obj.delta_FS_Kt(:, :, k)./obj.delta_Xi(:, :, 3)).^2.*obj.variancia(:, :, 3))));

    end %end for

end %end function

% Função do MEP para valores únicos (DEFAULT)
function obj = MEP_default(obj,cv_atrito,cv_coesao)

%definição dos coeficientes de variação default
obj.cv_atrito = cv_atrito;
obj.cv_coesao_ef = cv_coesao;

%Variância das variáveis Xi's

obj.variancia(:, :, 1) = (obj.cv_atrito.*obj.atrito./100).^2;
obj.variancia(:, :, 2) = (obj.cv_coesao_ef.*obj.coesao_ef./100).^2;

% Valor inicial de m pode ser zero ou não
obj.m(:, :) = obj.m_inicial*ones(obj.tam(1),obj.tam(2));

%Matriz com os valores de m acumulado
obj.m_acum = zeros(obj.tam(1),obj.tam(2),obj.tam(3));

%Definição das funções Omega,Gama e nw para a área de estudo

obj.Gama = obj.Gs.*(1-obj.n)+obj.n.*obj.Sr;
obj.nw = obj.n.*(1-obj.Sr);

```

```

obj.Omega = 2./(sin(2.*obj.beta).*obj.H.*obj.Peso_esp_w);

% Cálculo do valor de m para os valores médios e incrementos
% das variáveis Xi's
for k = 1:obj.tam(3)

% Cálculo do valor médio de m (índice 1)
obj.Ot= obj.Kt.*sin(obj.beta).*obj.m.*obj.H.*cos(obj.beta).*obj.delta_t;
obj.delta_m = (obj.It(:, :,k)-obj.Ot)./(obj.n.*obj.H.*(1-obj.Sr));
obj.m = obj.m+obj.delta_m;

%RESTRICÇÃO DO VALOR DE m

obj.m_maior_1 = obj.m >= 1;
obj.m_menor_1 = obj.m < 1;
obj.m = obj.m_maior_1 + obj.m_menor_1.*obj.m;

% VARRE A MATRIZ TODA
for i = 1:obj.tam(1)
for j = 1:obj.tam(2)

obj.vet_coesao = [obj.coesao_ef(i,j) -
sqrt(obj.variancia(i,j,2)),obj.coesao_ef(i,j) + sqrt(obj.variancia(i,j,2))];
obj.vet_atrito = [obj.atrito(i,j) -
sqrt(obj.variancia(i,j,1)),obj.atrito(i,j) + sqrt(obj.variancia(i,j,1))];

obj.soma_FS = 0;
obj.variancia_FS = 0;
obj.F = [0 0;0 0];

for t = 1:2
for l = 1:2

% cálculo da coesão total para toda a
% região (valor de m)
obj.Coesao_Total = (obj.vet_coesao(t)+obj.A(i,j)*(1-
obj.Lambda(i,j)*obj.m(i,j)^obj.alpha(i,j)))*obj.delta_s;

obj.F(t,l) =
(cot(obj.beta(i,j))*tan(obj.vet_atrito(l))*(obj.Gama(i,j)+obj.m(i,j)*(obj.nw(i,j)-
1))+obj.Coesao_Total*obj.Omega(i,j))/(obj.Gama(i,j)+obj.m(i,j)*obj.nw(i,j));

obj.soma_FS = obj.F(t,l)+obj.soma_FS;
end % end for
end %end for

obj.FS_med(i,j,k) = obj.soma_FS/4;

for t =1:2
for l = 1:2
obj.variancia_FS = ((obj.F(t,l)-obj.FS_med(i,j,k))^2)/4 +
obj.variancia_FS;
end % end for
end %end for

obj.desvio_FS(i,j,k) = sqrt(obj.variancia_FS);

end % end if
end % end if

obj.m_acum(:, :,k) = obj.m(:, :);

end % end if

end % end function

% Função do MEP para valores únicos (DEFAULT)
function obj = MEP_persona(obj,cv_atrito,cv_coesao_ef,cv_chuva,cv_Kt)

%definição do coeficientes de variação
obj.cv_atrito = cv_atrito;
obj.cv_coesao_ef = cv_coesao_ef;
obj.cv_It = cv_chuva;
obj.cv_Kt = cv_Kt;

```

```

%Variância das variáveis Xi's
obj.variancia(:, :, 1) = (obj.cv_atrito.*obj.atrito./100).^2;
obj.variancia(:, :, 2) = (obj.cv_coesao_ef.*obj.coesao_ef./100).^2;
obj.variancia(:, :, 3) = (obj.cv_Kt.*obj.Kt./100).^2;

for k = 1:obj.tam(3)
    obj.variancia_It(:, :, k) = (obj.cv_It.*obj.It(:, :, k)./100).^2;
end % end for

%Matriz com os valores de m iniciais (nulos)
obj.m = zeros(obj.tam(1), obj.tam(2), 3);

%Definição das funções Omega, Gama e nw para a área de estudo
obj.Gama = obj.Gs.* (1-obj.n)+obj.n.*obj.Sr;
obj.nw = obj.n.*(1-obj.Sr);
obj.Omega = 2./(sin(2*obj.beta).*obj.H.*obj.Peso_esp_w);

% Cálculo do valor de m para os valores médios e incrementos
% das variáveis Xi's
for k = 1:obj.tam(3)

    % VARRE A MATRIZ TODA
    for i = 1:obj.tam(1)
        for j = 1:obj.tam(2)

            obj.vet_coesao = [obj.coesao_ef(i,j) -
sqrt(obj.variancia(i,j,2)), obj.coesao_ef(i,j) + sqrt(obj.variancia(i,j,2))];
            obj.vet_atrito = [obj.atrito(i,j) -
sqrt(obj.variancia(i,j,1)), obj.atrito(i,j) + sqrt(obj.variancia(i,j,1))];
            obj.vet_Kt = [obj.Kt - sqrt(obj.variancia(i,j,3)), obj.Kt +
sqrt(obj.variancia(i,j,3))];
            obj.vet_It = [obj.It(i,j,k) - sqrt(obj.variancia_It(i,j,k)), obj.It(i,j,k) +
sqrt(obj.variancia_It(i,j,k))];

            obj.soma_FS = 0;
            obj.variancia_FS = 0;
            obj.F = [2, 2, 2, 2];

            % laços para a combinações das 4 variáveis
            % a = coesão; b = atrito; c = Kt; d = It
            for a = 1:2
                for b = 1:2
                    for c = 1:2
                        for d = 1:2

                            obj.Ot =
obj.vet_Kt(c)*sin(obj.beta(i,j))*obj.m(i,j)*obj.H(i,j)*cos(obj.beta(i,j))*obj.delta_t;
                            obj.delta_m = (obj.vet_It(d)-obj.Ot)/(obj.n(i,j)*obj.H(i,j)*(1-
obj.Sr(i,j)));

                            obj.m = obj.m+obj.delta_m;

                            %RESTRIÇÃO DO VALOR DE m

                            obj.m_maior_1 = obj.m >= 1;
                            obj.m_menor_1 = obj.m < 1;
                            obj.m = obj.m_maior_1 + obj.m_menor_1.*obj.m;

                            obj.Coesao_Total = (obj.vet_coesao(a)+obj.A(i,j)*(1-
obj.Lambda(i,j)*obj.m(i,j)^obj.alpha(i,j)))*obj.delta_s;

                            obj.F(a,b,c,d) =
(cot(obj.beta(i,j))*tan(obj.vet_atrito(b))* (obj.Gama(i,j)+obj.m(i,j)*(obj.nw(i,j)-
1))+obj.Coesao_Total*obj.Omega(i,j))/(obj.Gama(i,j)+obj.m(i,j)*obj.nw(i,j));

                            obj.soma_FS = obj.F(a,b,c,d)+obj.soma_FS;

                        end % end for
                    end % end for
                end % end for
            end %end for

            obj.FS_med(i,j,k) = obj.soma_FS/4;

            % cálculo da variância
            for a = 1:2
                for b = 1:2
                    for c = 1:2
                        for d = 1:2

```

```

obj.variancia_FS = ((obj.F(a,b,c,d)-obj.FS_med(i,j,k))^2)/4 +

        end % end for
        end % end for
        end % end for
        end %end for

        obj.desvio_FS(i,j,k) = sqrt(obj.variancia_FS);

        end % end if
        end % end if

        obj.m_acum(:, :, k) = obj.m(:, :);

        end % end if

end % end function

end % end methods

end %end class

classdef MonteCarloFS
    %MONTECARLOFS Summary of this class goes here
    % Detailed explanation goes here

    properties

        % Propriedades da área de estudo
        tam;
        % Propriedades do talude
        %Inclinação do talude(em radianos - SPRING)
        beta;
        H;

        % Propriedades do solo
        atrito;
        coesao_ef;
        A;
        Lambda;
        alpha;
        Gs;
        n;
        nw;
        Sr;
        Kt;
        % Funções para o cálculo do FS
        Coesao_Total;
        Gama;
        Omega;
        FS_med;
        FS;
        % Propriedades da chuva
        % Intensidade da chuva em m (metros)
        It;
        % Intervalo de tempo entre os passos em s(segundos)
        delta_t;
        m_inicial;
        m;
        delta_m;
        Ot;
        %Propriedades para o cálculo da PR (Aplicação do MonteCarlo)
        %coeficientes de variação
        cv_atrito;
        cv_coesao_ef;
        cv_It;
        cv_kt;
        % MONTECARLO
        %número de iterações
        num_iterac;
        %matrizes cujos elementos são vetores com as ditribuições
        distr_est_atrito;
        distr_est_coesao_ef;
        distr_est_kt;
        distr_est_It;
        %matriz de desvio-padrão (matriz multidimensional)
        % Ordem das variâncias salvas: Atrito,coesão, Kt, chuva
        desvio_padrao;
        desvio_padrao_It;
        %Devio Padrão do FS

```

```

desvio_FS;
Zcrit;
INF;
NINF;
PR;
% quantidade de análises que serão realizadas
quant_analis;
arq_analis;
cont;

m_maior_1;
m_menor_1
end

properties(Constant = true)
Peso_esp_w = 9.8;
delta_s = 1;
end

methods

% Inicialização do objeto (Para o MonteCarlo Default)
function obj =
MonteCarloFS(tamanho_area,matrix_Kt,matrix_beta,matrix_H,tempo,matrix_chuva,atrito,Gs,n,Sr,A,lambda,alpha,c
oesao_ef,valor_m_inicial,quant_analis)
obj.tam = tamanho_area;
obj.delta_t = tempo;
obj.Kt = matrix_Kt;
obj.beta = pi*(matrix_beta)/180;
obj.H = matrix_H;
obj.It = matrix_chuva;
obj.atrito = pi*(atrito)/180;
obj.Gs = Gs;
obj.n = n;
obj.Sr = Sr;
obj.A = A;
obj.Lambda = lambda;
obj.alpha = alpha;
obj.coesao_ef = coesao_ef;
obj.m_inicial = valor_m_inicial;
obj.quant_analis = quant_analis;
obj.FS_med = zeros(obj.tam(1),obj.tam(2),obj.tam(3));
obj.desvio_FS = zeros(obj.tam(1),obj.tam(2),obj.tam(3));

if obj.quant_analis == 3
obj.arq_analis = load(dado_quant);
end % end if

end %end function

%Default: Distribuições estatísticas consideradas normais para Coesão e Ângulo de Atrito
function obj = MonteCarlo_default(obj,matriz_cv_atrito,matriz_cv_coesao_ef,Iteracao)

%carrego matrizes de coef. variação
obj.cv_atrito = matriz_cv_atrito;
obj.cv_coesao_ef = matriz_cv_coesao_ef;

%Transforma CV em Desvio Padrão para cada célula (Para atrito e coesão efetiva)
obj.desvio_padrao = zeros(obj.tam(1),obj.tam(2),2);

obj.desvio_padrao(:, :,1) = (obj.cv_atrito.*obj.atrito./100);
obj.desvio_padrao(:, :,2) = (obj.cv_coesao_ef.*obj.coesao_ef./100);

%recebe o número de iterações definida pelo usuário
obj.num_iterac = Iteracao;

% Definição inicial das matrizes
obj.distr_est_atrito = zeros(obj.tam(1),obj.tam(2),obj.num_iterac);
obj.distr_est_coesao_ef = zeros(obj.tam(1),obj.tam(2),obj.num_iterac);

%Cria vetor aleatório de distribuição normal para cada célula que
%representa uma certa porção da área de estudo- função normrnd
%Cria uma matriz de vetores
for i = 1:obj.tam(1)
for j = 1:obj.tam(2)
obj.distr_est_atrito(i,j,:) =
normrnd(obj.atrito(i,j),obj.desvio_padrao(i,j,1),1,obj.num_iterac);
obj.distr_est_coesao_ef(i,j,:) =
normrnd(obj.coesao_ef(i,j),obj.desvio_padrao(i,j,2),1,obj.num_iterac);
end %end for

```

```

end % end for

% Valor inicial de m pode ser zero ou não
obj.m(obj.tam(1),obj.tam(2)) = 0;
obj.m(:, :) = obj.m_inicial*ones(obj.tam(1),obj.tam(2));

%Definição das funções Omega,Gama e nw para a área de estudo
obj.Gama = obj.Gs.*(1-obj.n)+obj.n.*obj.Sr;
obj.nw = obj.n.*(1-obj.Sr);
obj.Omega = 2./(sin(2.*obj.beta).*obj.H.*obj.Peso_esp_w);

%obj.Coesao_Total = zeros(obj.tam(1),obj.tam(2),obj.num_iterac);
obj.FS(obj.tam(1),obj.tam(2),obj.num_iterac) = 0;

for k = 1:obj.tam(3)

    obj.m = obj.m + (obj.It(:, :, k)-
(obj.Kt.*sin(obj.beta).*obj.m.*obj.H.*cos(obj.beta).*obj.delta_t))./(obj.n.*obj.H.*(1-obj.Sr));

    %RESTRICÃO DO VALOR DE m

    obj.m_maior_1 = obj.m >= 1;
    obj.m_menor_1 = obj.m < 1;
    obj.m = obj.m_maior_1 + obj.m_menor_1.*obj.m;

    for l = 1:obj.num_iterac

        obj.FS(:, :, l) =
(cot(obj.beta).*tan(obj.distr_est_atrito(:, :, l)).*(obj.Gama+obj.m.*(obj.nw-
1))+((obj.distr_est_coesao_ef(:, :, l)+obj.A.*(1-
obj.Lambda.*obj.m.^obj.alpha)).*obj.delta_s).*obj.Omega)./(obj.Gama+obj.m.*obj.nw);
        obj.FS_med(:, :, k) = obj.FS_med(:, :, k) + obj.FS(:, :, l);

    end %end for

    obj.FS_med(:, :, k) = obj.FS_med(:, :, k)/obj.num_iterac;

    %obj.desvio_FS(:, :, k) = std(obj.FS,0,3);
    xma = zeros(obj.tam(1),obj.tam(2),obj.num_iterac);

    for i = 1:obj.num_iterac

        xma(:, :, i) = obj.FS_med(:, :, k);

    end

    obj.desvio_FS(:, :, k) = sqrt(sum((obj.FS-xma).^2,3)/obj.num_iterac);

end %end for

clear obj.cv_atrito obj.cv_coesao_ef obj.desvio_padrao;
clear obj.distr_est_atrito obj.distr_est_coesao_ef obj.m;
clear obj.FS xma;

end %end function

%Personalizado: Distribuições estatísticas consideradas normais
%para Coesão, Ângulo de Atrito, Chuva e Kt
function obj =
MonteCarlo_person(obj,matriz_cv_atrito,matriz_cv_coesao_ef,matriz_cv_kt,matriz_cv_chuva,Iteracao)

%recebe o número de iterações definida pelo usuário
obj.num_iterac = Iteracao;

%carrego matrizes de coef. variação
obj.cv_atrito = matriz_cv_atrito;
obj.cv_coesao_ef = matriz_cv_coesao_ef;
obj.cv_kt = matriz_cv_kt;
obj.cv_It = matriz_cv_chuva;

% Criação de matriz de desvios-padrão nula inicialmente
obj.desvio_padrao = zeros(obj.tam(1),obj.tam(2),3);
obj.desvio_padrao_It = zeros(obj.tam(1),obj.tam(2),obj.tam(3));

obj.distr_est_atrito = zeros(obj.tam(1),obj.tam(2),obj.num_iterac);
obj.distr_est_coesao_ef = zeros(obj.tam(1),obj.tam(2),obj.num_iterac);
obj.distr_est_kt = zeros(obj.tam(1),obj.tam(2),obj.num_iterac);

```

```

%Transforma CV em Desvio Padrão para cada célula (Para atrito e coesão efetiva)
obj.desvio_padrao(:, :, 1) = (obj.cv_atrito.*obj.atrito./100);
obj.desvio_padrao(:, :, 2) = (obj.cv_coesao_ef.*obj.coesao_ef./100);
obj.desvio_padrao(:, :, 3) = (obj.cv_kt.*obj.Kt./100);

for i = 1:obj.tam(3)
    obj.desvio_padrao_It(:, :, i) = (obj.cv_It.*obj.It(:, :, i)./100);
end % end for

%Cria vetor aleatório de distribuição normal para cada célula que
%representa uma certa porção da área de estudo- função normrnd
%Cria uma matriz de vetores
for i = 1:obj.tam(1)
    for j = 1:obj.tam(2)
        obj.distr_est_atrito(i, j, :) =
normrnd(obj.atrito(i, j), obj.desvio_padrao(i, j, 1), 1, obj.num_iterac);
        obj.distr_est_coesao_ef(i, j, :) =
normrnd(obj.coesao_ef(i, j), obj.desvio_padrao(i, j, 2), 1, obj.num_iterac);
        obj.distr_est_kt(i, j, :) =
normrnd(obj.Kt(i, j), obj.desvio_padrao(i, j, 3), 1, obj.num_iterac);

    end %end for
end % end for

% Valor inicial de m pode ser zero ou não
obj.m = obj.m_inicial*ones(obj.tam(1), obj.tam(2), obj.num_iterac);

%Definição das funções Omega, Gama e nw para a área de estudo
obj.Gama = obj.Gs.* (1-obj.n)+obj.n.*obj.Sr;
obj.nw = obj.n.* (1-obj.Sr);
obj.Omega = 2./(sin(2.*obj.beta).*obj.H.*obj.Peso_esp_w);

for k = 1:obj.tam(3)

    % Matriz com valores de distribuição normal para a chuva
    % Para cada passo de chuva, é criada uma matriz com os
    % valores que seguem a distr. normal
    for i = obj.tam(1)
        for j = obj.tam(2)
            obj.distr_est_It(i, j, :) =
normrnd(obj.It(i, j, k), obj.desvio_padrao_It(i, j, k), 1, obj.num_iterac);
        end % end for
    end % end for

    for l = 1:obj.num_iterac

        % Cálculo do valor médio de m (índice 1)
        obj.Ot(:, :, 1) =
obj.distr_est_kt(:, :, 1).*sin(obj.beta).*obj.m(:, :, 1).*obj.H.*cos(obj.beta).*obj.delta_t;
        obj.delta_m(:, :, 1) = (obj.distr_est_It(:, :, 1)-obj.Ot(:, :, 1))./(obj.n.*obj.H.*(1-
obj.Sr));

        obj.m(:, :, 1) = obj.m(:, :, 1) + obj.delta_m(:, :, 1);

        %RESTRIÇÃO DO VALOR DE m

        obj.m_maior_1 = obj.m >= 1;
        obj.m_menor_1 = obj.m < 1;
        obj.m = obj.m_maior_1 + obj.m_menor_1.*obj.m;

        % Cálculo do FS médio
        obj.Coesao_Total(:, :, 1) = (obj.distr_est_coesao_ef(:, :, 1)+obj.A.*(1-
obj.Lambda.*obj.m(:, :, 1).^obj.alpha)).*obj.delta_s;
        obj.FS(:, :, 1) =
(cot(obj.beta).*tan(obj.distr_est_atrito(:, :, 1)).*(obj.Gama+obj.m(:, :, 1)).*(obj.nw-
1))+obj.Coesao_Total(:, :, 1).*obj.Omega)./(obj.Gama+obj.m(:, :, 1).*obj.nw);

    end % end for

    obj.FS_med(:, :, k) = mean(obj.FS, 3);
    obj.desvio_FS(:, :, k) = std(obj.FS, 0, 3);

end % end for
end % end function

% FUNÇÃO PARA CÁLCULO DA PR E DO ÍNDICE DE CONFIABILIDADE
% Função que calcula a PR a partir dos valores do FS médio e do
% desvio-padrão
function obj = ProbRup(obj, FS_crit)

```

```

obj.Zcrit = zeros(obj.tam(1),obj.tam(2));
obj.INF = obj.FS_med == Inf;
obj.NINF = obj.FS_med ~= Inf;

% Para todos os passos de chuva
if obj.quant_analis == 1

    clear obj.PR;

    obj.PR(obj.tam(1),obj.tam(2),obj.tam(3)) = 0;

    for k = 1:obj.tam(3)

        obj.Zcrit = real((FS_crit-obj.FS_med(:,:,k))./obj.desvio_FS(:,:,k));
        %obj.Zcrit = real(obj.Zcrit);
        %zcrit = obj.Zcrit;
        %save zcrit -ascii

        for i = 1:obj.tam(1)
            for j = 1:obj.tam(2)

                if obj.FS_med(i,j,k) ~= Inf

                    x = -10:0.01:obj.Zcrit(i,j);

                    Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

                    obj.PR(i,j,k) = trapz(x,Y,2);

                end % end if

            end % end for
        end % end for
    end %end for

    clear x Y obj.FS_med obj.desvio_FS obj.Zcrit;

    % para o primeiro e o último passo
elseif obj.quant_analis == 2
obj.PR = zeros(obj.tam(1),obj.tam(2),2);
obj.cont = 1;
for k = 1:obj.tam(3)-1:obj.tam(3)

    obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.desvio_FS(:,:,k);

    for i = 1:obj.tam(1)
        for j = 1:obj.tam(2)
            if obj.FS_med(i,j,k) ~= Inf

                x = -10:0.01:obj.Zcrit(i,j);

                Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

                obj.PR(i,j,obj.cont) = trapz(x,Y);

            end % end if
        end % end for
    end % end for

    obj.cont = obj.cont+1;

end %end for

end % end if
end % end function

% Função para o CÁLCULO DA PR PARA PASSOS DEFINIDOS POR ARQUIVO
function obj = ProbRupArquivo(obj,arq_analis,FS_crit)

obj.arq_analis = arq_analis;

obj.PR = zeros(obj.tam(1),obj.tam(2),size(obj.arq_analis));
obj.cont = 1;

while obj.cont < size(obj.arq_analis)+1

    k = obj.arq_analis(obj.cont);

    obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.desvio_FS(:,:,k);

    for i = 1:obj.tam(1)
        for j = 1:obj.tam(2)

            if obj.FS_med(i,j,k) ~= Inf

```

```

        x = -10:0.001:obj.Zcrit(i,j);
        Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);
        obj.PR(i,j,obj.cont) = trapz(x,Y);
    end % end if
end % end for
end % end for

obj.cont = obj.cont+1;
end % end while
end %end function
end % end methods
end %end class

```

A.4 FUNÇÕES PARA O MODELO DE DESLIZAMENTOS E FLUXO DE DETRITOS

A.4.1 INTERFACE GRÁFICA

```

function varargout = FLUXO_DETRITOS(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @FLUXO_DETRITOS_OpeningFcn, ...
                  'gui_OutputFcn',  @FLUXO_DETRITOS_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function FLUXO_DETRITOS_OpeningFcn(hObject, eventdata, handles, varargin)
global opcao_met_detr;
global num_cv_2;
global num_cv_1;
global num_cv_3;
global num_cv_4;
global num_cv_5;
global opcao_cv;
global opcao_ch;
global opcao_analise;
global FS_crit;

global ctrol_calibrar;

% valores iniciais de coeficientes de variação
num_cv_1 = 40;
num_cv_2 = 10;
num_cv_3 = 0;
num_cv_4 = 0;
num_cv_5 = 0;
opcao_ch = 1;
opcao_cv = 1;
opcao_met_detr = 2;
opcao_analise = 1;
FS_crit = 1;
% controle de calibração
ctrol_calibrar = 0;
% Choose default command line output for FLUXO_DETRITOS
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = FLUXO_DETRITOS_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global distrito;

```

```

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text27, 'String', fullpathname);

distrito = fullfile(pathname, filename);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global dado_decl;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text28, 'String', fullpathname);

dado_decl = fullfile(pathname, filename);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
global dado_A_b;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text29, 'String', fullpathname);

dado_A_b = fullfile(pathname, filename);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
global dado_V_est;

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text30, 'String', fullpathname);

dado_V_est = fullfile(pathname, filename);

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)

function edit12_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit9_Callback(hObject, eventdata, handles)

```

```

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit14_Callback(hObject, eventdata, handles)
global min;

min = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton20.
function pushbutton20_Callback(hObject, eventdata, handles)
global dado_chuva;

folder_name = uigetdir;

fullpathname = strcat(folder_name);
set(handles.text75,'String',fullpathname);

dado_chuva = fullpathname;

function edit13_Callback(hObject, eventdata, handles)
global num_passos;

num_passos = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

end

% --- Executes on button press in pushbutton34.
function pushbutton34_Callback(hObject, eventdata, handles)
global dado_quant;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text124, 'String', fullpathname);

dado_quant = fullfile(pathname, filename);

function edit15_Callback(hObject, eventdata, handles)
global pixel_cal;

pixel_cal = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in pushbutton25.
function pushbutton25_Callback(hObject, eventdata, handles)
global dado_V_Din;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text94, 'String', fullpathname);

dado_V_Din = fullfile(pathname, filename);

function edit16_Callback(hObject, eventdata, handles)
global Iz_prob;

Iz_prob1 = str2double(get(hObject, 'String'));
Iz_prob = Iz_prob1(1);

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in pushbutton35.
function pushbutton35_Callback(hObject, eventdata, handles)

% --- Executes on button press in pushbutton36.
function pushbutton36_Callback(hObject, eventdata, handles)
global grade_cv_gama;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text134, 'String', fullpathname);

grade_cv_gama = fullfile(pathname, filename);

% --- Executes on button press in pushbutton37.
function pushbutton37_Callback(hObject, eventdata, handles)
global grade_cv_phi;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text136, 'String', fullpathname);

grade_cv_phi = fullfile(pathname, filename);

% --- Executes on button press in pushbutton38.
function pushbutton38_Callback(hObject, eventdata, handles)
global grade_cv_c;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text138, 'String', fullpathname);

grade_cv_c = fullfile(pathname, filename);

% --- Executes on button press in pushbutton39.
function pushbutton39_Callback(hObject, eventdata, handles)
global grade_cv_kz;

[filename pathname] = uigetfile({'*..*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text140, 'String', fullpathname);

```

```

grade_cv_kz = fullfile(pathname,filename);

function edit27_Callback(hObject, eventdata, handles)
global num_cv_2;

num_cv_2 = str2double(get(hObject,'String'));

function edit27_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit28_Callback(hObject, eventdata, handles)
global num_cv_1;

num_cv_1 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit28_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit29_Callback(hObject, eventdata, handles)
global num_cv_3;

num_cv_3 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit29_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit30_Callback(hObject, eventdata, handles)
global num_cv_4;

num_cv_4 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit30_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit31_Callback(hObject, eventdata, handles)
global num_cv_5;

num_cv_5 = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit31_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
global opcao_cv;
global v;
global num_cv_2;
global num_cv_1;
global num_cv_3;
global num_cv_4;
global num_cv_5;

v = get(handles.popupmenu4,'Value');

if opcao_cv == 1
    if v == 1

```

```

set(handles.edit27, 'enable', 'on','String','40');
set(handles.edit28, 'enable', 'on','String','10');
num_cv_1 = 40;
num_cv_2 = 10;
num_cv_3 = 0;
num_cv_4 = 0;
num_cv_5 = 0;
set(handles.edit29, 'enable', 'off','String','');
set(handles.edit30, 'enable', 'off','String','');
set(handles.edit31, 'enable', 'off','String','');

set(handles.pushbutton44, 'enable', 'off');
set(handles.pushbutton38, 'enable', 'off');
set(handles.pushbutton37, 'enable', 'off');
set(handles.pushbutton36, 'enable', 'off');

set(handles.pushbutton39, 'enable', 'off');

elseif v == 2
set(handles.edit27, 'enable', 'on','String','');
set(handles.edit28, 'enable', 'on','String','');
set(handles.edit29, 'enable', 'on','String','');
set(handles.edit30, 'enable', 'on','String','');
set(handles.edit31, 'enable', 'on','String','');

set(handles.pushbutton44, 'enable', 'off');
set(handles.pushbutton38, 'enable', 'off');
set(handles.pushbutton37, 'enable', 'off');
set(handles.pushbutton36, 'enable', 'off');

set(handles.pushbutton39, 'enable', 'off');

end % end if

elseif opcao_cv == 2

if v == 1

set(handles.edit27, 'enable', 'off');
set(handles.edit28, 'enable', 'off');
set(handles.edit29, 'enable', 'off');
set(handles.edit30, 'enable', 'off');
set(handles.edit31, 'enable', 'off');

set(handles.pushbutton38, 'enable', 'on');
set(handles.pushbutton37, 'enable', 'on');
set(handles.pushbutton36, 'enable', 'off');
set(handles.pushbutton44, 'enable', 'off');
set(handles.pushbutton39, 'enable', 'off');

elseif v == 2

set(handles.edit27, 'enable', 'off');
set(handles.edit28, 'enable', 'off');
set(handles.edit29, 'enable', 'off');
set(handles.edit30, 'enable', 'off');
set(handles.edit31, 'enable', 'off');

set(handles.pushbutton38, 'enable', 'on');
set(handles.pushbutton37, 'enable', 'on');
set(handles.pushbutton36, 'enable', 'on');
set(handles.pushbutton44, 'enable', 'on');
set(handles.pushbutton39, 'enable', 'on');

end % end if
end % end if

% --- Executes during object creation, after setting all properties.
function popsubmenu4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton40.
function pushbutton40_Callback(hObject, eventdata, handles)

global opcao_met_detrit;
global distrito;
global dado_decl;
global dado_A_b;
global dado_V_est;
global dado_V_Din;
global pixel_cal;

```

```

global entrada;
global FS_crit;

% CALIBRAÇÃO PARA TODA ÁREA
if opcao_met_detrit == 2

    % BARRA DE PROGRESSO
    h = waitbar(0,'1','Name','Calibração...','...
        'CreateCancelBtn',...
        'setappdata(gcf,'canceling',1)');
    setappdata(h,'canceling',0)

    % BARRA DE PROGRESSO
    steps = 5;
    waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))

    % Importa dados de entrada e cria objeto
    entrada =
Input_Debris(load(distrito),load(dado_decl),load(dado_A_b),load(dado_V_est),load(dado_V_Din));

    % BARRA DE PROGRESSO
    waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))

    % Mostra os valores de Iz_steady na tela
    set(handles.text147,'String',entrada.input_dynamic(3,1));
    set(handles.text148,'String',entrada.input_dynamic(3,2));
    set(handles.text149,'String',entrada.input_dynamic(3,3));

    % BARRA DE PROGRESSO
    waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))

    %Gera os vetores das variáveis dinâmicas dentro do intervalo determinado
    entrada = Vector_Dynamic(entrada);

    % BARRA DE PROGRESSO
    waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))

    % Calibration model
    entrada = Model_FS_Total(entrada,FS_crit);

    % libera o botão EXECUTAR
    set(handles.pushbutton42, 'enable', 'on');

    set(handles.radiobutton1, 'enable', 'on');
    set(handles.radiobutton2, 'enable', 'on');
    set(handles.radiobutton3, 'enable', 'on');
    set(handles.edit14, 'enable', 'off');
    set(handles.edit13, 'enable', 'on');
    set(handles.pushbutton20, 'enable', 'on');
    set(handles.text75, 'enable', 'on');

    % Campos de Iz(steady) ativados na opção pixels totais
    set(handles.radiobutton28, 'enable', 'on');
    set(handles.radiobutton29, 'enable', 'on');
    set(handles.radiobutton30, 'enable', 'on');
    set(handles.text147, 'enable', 'on');
    set(handles.text148, 'enable', 'on');
    set(handles.text149, 'enable', 'on');
    set(handles.edit32, 'enable', 'on');

    %bloqueia o botão CALIBRAR
    % TIREI ESTE BLOQUEIO PORQUE O USUÁRIO PODERÁ CALIBRAR NOVAMENTE O
    % MODELO, MAS, UMA VEZ JÁ EXISTINDO UMA CALIBRAÇÃO DISPONÍVEL, O BOTÃO
    % EXECUTAR ESTARÁ TAMBÉM DISPONÍVEL
    %set(handles.pushbutton40, 'enable', 'off');

    % BARRA DE PROGRESSO
    waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))
    delete(h)

% CALIBRAÇÃO PARA CERTOS PIXELS
elseif opcao_met_detrit == 3

    % BARRA DE PROGRESSO
    h = waitbar(0,'1','Name','Calibração...','...
        'CreateCancelBtn',...
        'setappdata(gcf,'canceling',1)');
    setappdata(h,'canceling',0)

    % BARRA DE PROGRESSO
    steps = 5;
    waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))

    % Importa dados de entrada e cria objeto
    entrada =
Input_Debris(load(distrito),load(dado_decl),load(dado_A_b),load(dado_V_est),load(dado_V_Din));

```

```

% BARRA DE PROGRESSO
waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))

% Gera posições aleatórias dentro da área de estudo
entrada = Input_Choise_Pixel(entrada,pixel_cal);

% BARRA DE PROGRESSO
waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))

%Gera os vetores das variáveis dinâmicas dentro do intervalo determinado
entrada = Vector_Dynamic(entrada);

% BARRA DE PROGRESSO
waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))

entrada = Model_FS_Pixel(entrada,FS_crit);

% Mostra os valores de Iz_steady na tela
set(handles.text147,'String',entrada.input_dynamic(3,1));
set(handles.text148,'String',entrada.input_dynamic(3,2));
set(handles.text149,'String',entrada.input_dynamic(3,3));

% libera o botão EXECUTAR
set(handles.pushbutton42, 'enable', 'on');

set(handles.radiobutton1, 'enable', 'on');
set(handles.radiobutton2, 'enable', 'on');
set(handles.radiobutton3, 'enable', 'on');
set(handles.edit14, 'enable', 'off');
set(handles.edit13, 'enable', 'on');
set(handles.pushbutton20, 'enable', 'on');
set(handles.text75, 'enable', 'on');

% Campos de Iz(eady) ativados na opção certos pixels
set(handles.radiobutton28, 'enable', 'on');
set(handles.radiobutton29, 'enable', 'on');
set(handles.radiobutton30, 'enable', 'on');
set(handles.text147, 'enable', 'on');
set(handles.text148, 'enable', 'on');
set(handles.text149, 'enable', 'on');
set(handles.edit32, 'enable', 'on');

%bloqueia o botão CALIBRAR
% TIREI ESTE BLOQUEIO PORQUE O USUÁRIO PODERÁ CALIBRAR NOVAMENTE O
% MODELO, MAS, UMA VEZ JÁ EXISTINDO UMA CALIBRAÇÃO DISPONÍVEL, O BOTÃO
% EXECUTAR ESTARÁ TAMBÉM DISPONÍVEL
%set(handles.pushbutton40, 'enable', 'off');

waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))
delete(h)

end % end if

% --- Executes on button press in pushbutton42.
function pushbutton42_Callback(hObject, eventdata, handles)
global ctrol_per;
global opcao_met_detrin;
global dado_chuva;
global num_passos;
global opcao_ch;
global min;
global Iz_prob;
global Iz_est_PR;
global opcao_analise;
global dado_quant;
global num_cv_2;
global num_cv_1;
global num_cv_3;
global num_cv_4;
global num_cv_5;
global distrito;
global dado_decl;
global dado_A_b;
global dado_V_est;
global dado_V_Din;
global PR;
global entrada;
global FS_crit;
global dummy;

%Executa o método probabilístico
if opcao_met_detrin == 1

```

```

% Importa dados de entrada e cria objeto
entrada = Input_Debris_Probability(load(distrito),load(dado_decl),load(dado_A_b),load(dado_V_est));

% obtém a matriz de Iz que será usado na análise
entrada = Input_Rain(entrada,num_passos,dado_chuva);

if opcao_ch ~= 3

    entrada = InputFS(entrada,opcao_ch);

else

    entrada = InputFS_MIN(entrada,min);

end % end if

entrada = Fosm(entrada,Iz_prob,num_cv_2,num_cv_1,num_cv_3,num_cv_4,num_cv_5);

if opcao_analise ~= 3

    entrada = ProbRup(entrada,opcao_analise,FS_crit);

else

    entrada = ProbRupArquivo(entrada,dado_quant,FS_crit);

end % end if

PR = entrada.PR;
ctrol_per = 1;

% Mensagem indicando fim das análises de PERIGO
warndlg('Análise do perigo estabelecida com sucesso!','Operação realizada');

% Executa para toda a área ou para um número determinado de pixels
else

    % Input data of rain

    entrada = Input_Rain(entrada,num_passos,dado_chuva);

    if opcao_ch ~= 3

        entrada = InputFS(entrada,opcao_ch);

    else

        entrada = InputFS_MIN(entrada,min);

    end % end if

    %Application of Rains
    %DETERMINA QUAIS PASSOS DE PR SERÃO CALCULADOS
    if opcao_analise ~= 3

        entrada = Thresholds(entrada,Iz_est_PR,opcao_analise,dummy);

    else

        entrada = Thresholds_Arq(entrada,Iz_est_PR,load(dado_quant));

    end % end if

    PR = entrada.Prob_Rup;
    ctrol_per = 1;

    % Mensagem indicando fim das análises de PERIGO
    warndlg('Análise do perigo estabelecida com sucesso!','Operação realizada');

end % end if

% --- Executes on button press in pushbutton43.
function pushbutton43_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function uipanel7_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel7.
function uipanel7_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_ch;

% opção das chuvas em mm/h
if hObject == handles.radiobutton1
    opcao_ch = 1;
    set(handles.edit14, 'enable', 'off');
% opção das chuvas em mm/dia
elseif hObject == handles.radiobutton2

```

```

        opcao_ch = 2;
        set(handles.edit14, 'enable', 'off');
% opção das chuvas em mm por X min
elseif hObject == handles.radiobutton3
    opcao_ch = 3;
    set(handles.edit14, 'enable', 'on');

end %end if

% --- Executes during object creation, after setting all properties.
function uipanel9_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel9.
function uipanel9_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_met_detrit;

% opção da análise probabilística
if hObject == handles.radiobutton6
    opcao_met_detrit = 1;
    set(handles.edit15, 'enable', 'off');
    set(handles.radiobutton31, 'enable', 'on');
    set(handles.radiobutton32, 'enable', 'on');
    set(handles.popupmenu4, 'enable', 'on');
    % Libera parte da chuva
    set(handles.radiobutton1, 'enable', 'on');
    set(handles.radiobutton2, 'enable', 'on');
    set(handles.radiobutton3, 'enable', 'on');
    set(handles.edit14, 'enable', 'off');
    set(handles.edit13, 'enable', 'on');
    set(handles.pushbutton20, 'enable', 'on');
    set(handles.text75, 'enable', 'on');

    % Libera o botão EXECUTAR e bloqueia o botão CALIBRAR
    set(handles.pushbutton42, 'enable', 'on');
    set(handles.pushbutton40, 'enable', 'off');
    % Libera o campo do Iz_steady
    set(handles.edit16, 'enable', 'on');

    % Campos de Iz(steady) desativados na opção FOSM
    set(handles.text147, 'enable', 'off');
    set(handles.text148, 'enable', 'off');
    set(handles.text149, 'enable', 'off');
    set(handles.edit32, 'enable', 'off');

    % Botão para variáveis dinâmicas desativado no FOSM
    set(handles.pushbutton25, 'enable', 'off');

    % Campos de Iz(steady) ativados na opção FOSM
    set(handles.radiobutton28, 'enable', 'on');
    set(handles.radiobutton29, 'enable', 'on');
    set(handles.radiobutton30, 'enable', 'on');

% opção determinística com calibração para toda área
elseif hObject == handles.radiobutton4
    opcao_met_detrit = 2;
    set(handles.edit15, 'enable', 'off');
    set(handles.radiobutton31, 'enable', 'off');
    set(handles.radiobutton32, 'enable', 'off');
    set(handles.popupmenu4, 'enable', 'off');
    set(handles.radiobutton1, 'enable', 'off');
    set(handles.radiobutton2, 'enable', 'off');
    set(handles.radiobutton3, 'enable', 'off');
    set(handles.edit14, 'enable', 'off');
    set(handles.edit13, 'enable', 'off');
    set(handles.pushbutton20, 'enable', 'off');

    % Botão para variáveis dinâmicas ativado
    set(handles.pushbutton25, 'enable', 'on');
    set(handles.text75, 'enable', 'off');

    % Bloqueia o botão EXECUTAR e libera o botão CALIBRAR
    set(handles.pushbutton42, 'enable', 'off');
    set(handles.pushbutton40, 'enable', 'on');
    % Trava o campo do Iz_steady
    set(handles.edit16, 'enable', 'off');

% opção determinística com calibração para alguns pixels
elseif hObject == handles.radiobutton5
    opcao_met_detrit = 3;
    set(handles.edit15, 'enable', 'on');
    set(handles.radiobutton31, 'enable', 'off');
    set(handles.radiobutton32, 'enable', 'off');
    set(handles.popupmenu4, 'enable', 'off');
    set(handles.radiobutton1, 'enable', 'off');
    set(handles.radiobutton2, 'enable', 'off');
    set(handles.radiobutton3, 'enable', 'off');
    set(handles.edit14, 'enable', 'off');

```

```

set(handles.edit13, 'enable', 'off');
set(handles.pushbutton20, 'enable', 'off');
set(handles.text75, 'enable', 'off');

% Botão para variáveis dinâmicas ativado
set(handles.pushbutton25, 'enable', 'on');

% Bloqueia o botão EXECUTAR e libera o botão CALIBRAR
set(handles.pushbutton42, 'enable', 'off');
set(handles.pushbutton40, 'enable', 'on');

% Trava o campo do Iz_steady
set(handles.edit16, 'enable', 'off');

end %end if

% --- Executes during object creation, after setting all properties.
function uipanel28_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel28.
function uipanel28_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_cv;

if hObject == handles.radiobutton31
    opcao_cv = 1;

    set(handles.pushbutton38, 'enable', 'off');
    set(handles.pushbutton37, 'enable', 'off');
    set(handles.pushbutton36, 'enable', 'off');
    set(handles.pushbutton39, 'enable', 'off');
    set(handles.pushbutton44, 'enable', 'off');

elseif hObject == handles.radiobutton32
    opcao_cv = 2;

    set(handles.edit27, 'enable', 'off');
    set(handles.edit28, 'enable', 'off');
    set(handles.edit29, 'enable', 'off');
    set(handles.edit30, 'enable', 'off');
    set(handles.edit31, 'enable', 'off');

end % end if

% --- Executes during object creation, after setting all properties.
function uipanel26_CreateFcn(hObject, eventdata, handles)

% --- Executes when selected object is changed in uipanel26.
function uipanel26_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_analise;

% opção para todos os passos
if hObject == handles.radiobutton28
    opcao_analise = 1;
    set(handles.pushbutton34, 'enable', 'off');
% opção para primeiro e último
elseif hObject == handles.radiobutton29
    opcao_analise = 2;
    set(handles.pushbutton34, 'enable', 'off');
% opção para passos definidos
elseif hObject == handles.radiobutton30
    opcao_analise = 3;
    set(handles.pushbutton34, 'enable', 'on');

end %end if

% --- Executes on button press in pushbutton44.
function pushbutton44_Callback(hObject, eventdata, handles)
global grade_cv_H;

[filename pathname] = uigetfile({'*. '*}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text132, 'String', fullpathname);

grade_cv_H = fullfile(pathname, filename);

function edit32_Callback(hObject, eventdata, handles)
global Iz_est_PR;

Iz_est_PR = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit32_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
    end

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)

% Hint: place code in OpeningFcn to populate axes1
flux = imread('imag_perigo.jpg');
image(flux);
axis off

function edit33_Callback(hObject, eventdata, handles)
global FS_crit;

FS_crit = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit33_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit34_Callback(hObject, eventdata, handles)
global dummy;

dummy = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit34_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end % end class

```

A.4.2 CLASSES ESPECÍFICAS PARA EXECUÇÃO DO MODELO

```

classdef Input_Debris

    properties
        % matriz com as divisões homogêneas dos distritos da bacia (resultante do SIG)
        matrix_area;

        % matrix of static input variables
        % Rows-> distritos distintos
        % Columns-> Na sequência: Zt, gama_s, atrito, coesão, Kx, Kz, Do
        matrix_static;

        % matrix of topographic variables
        matrix_decliv;
        matrix_Ab;
        tam;

        %Random array of pixels
        rdm_pix;

        %number of pixel
        number_pixel;

        % dynamic input variables
        %
        % matriz com os dados das variáveis dinâmicas (3X3)
        % Rows: 1->data of T, 2->data of Iz; 3->data of Iz(steady)
        %Columns: 1->minimum value, 2->maximum value,3->number of value
        input_dynamic;
        vector_T;
        vector_Iz;
        vector_Iz_std;
        %dimensões dos vetores das variáveis dinâmicas
        % columns 1: vetor Iz_steady; columns 2: vetor Iz; columns 1: vetor T;
        dimen_vect_dyn;

        %Model properties
        Zt_d;
        Psi_Z_0;
        T_ast;
        R;
        Psi_Z_T;
        FS;
        cont;
        result;
        T;
        Iz;
    end
end

```

```

% Input Rain
%
% number of data rain
num_rain;
unidade_chuva;
delta_t;
%matriz multidimensional referente aos dados de intensidade de chuva(cada dimensão corresponde os
dados de intensidade de chuva da região em estudo para um determinado tempo)
intens_rain;
%vetor que traz os tempos nos quais foram tomados os valores de intensidade de chuva
time_rain;
tam_rain;
string_nome;
string_complem;
numero_arquivo;
nome_arquivo;
Prob_Rup;
arq_analis;

%Input probabilistic method
%
% Vetor para os valores de Iz-steady,Iz e T
vector_dynamic;
% Vetor para os valores de coeficiente de variação
vector_cv;
% Vetor com os incrementos (+10%) dos valores médios da matrix static
vector_incr;
delta;
varianc;
FS_fosm;
FS_desvio;
PR;
quant_analis;
Zcrit;
INF;
NINF;
FS_med;
% Dados para busca binária
lim_sup;
lim_inf;
med;
Buscado;
% valor para usência de DADOS (EM RELAÇÃO A MATRIZ DE CHUVAS)
no_data;

end % end properties

properties(Constant = true)
    % Peso específico da água em N/m³
    Peso_esp_w = 9800;

end % end properties constant

methods

    %MÉTODOS DE INICIALIZAÇÃO DO OBJETO E ENTRADA DE DADOS
    %
    % inicializa objeto
    function obj = Input_Debris(area_estudo,declividade,drenagem_bacia,matrix_static,matrix_dynamic)

        % data of GIS (SPRING)
        obj.matrix_area = area_estudo;
        obj.matrix_decliv = declividade;
        obj.matrix_Ab = drenagem_bacia;

        obj.matrix_static = matrix_static;

        % transforma gama_solo da unidade de kN/m³ para N/m³
        obj.matrix_static(:,2) = 1000*obj.matrix_static(:,2);

        % transforma a coesão da unidade de kPa para Pa
        obj.matrix_static(:,4) = 1000*obj.matrix_static(:,4);

        % transforma Kx da unidade de mm/s para m/s
        obj.matrix_static(:,5) = obj.matrix_static(:,5)/1000;

        % transforma Kz da unidade de mm/s para m/s
        obj.matrix_static(:,6) = obj.matrix_static(:,6)/1000;

        % transforma Do da unidade de mm²/s para m²/s
        obj.matrix_static(:,7) = obj.matrix_static(:,7);

        %load array of dynamic variables

```

```

    obj.input_dynamic = matrix_dynamic;
    obj.tam = size(obj.matrix_area);

end % end function

% Lê os dados de chuva e salva-os em uma matriz multidimensional
%Deve-se passar o número de passos de chuva como argumento
function obj = Input_Rain(obj,num_rain,dado_chuva)

    obj.num_rain = num_rain;
    obj.string_nome = 'input_chuva';
    obj.string_complem = '.txt';

    for i =1:obj.num_rain
        obj.numero_arquivo = mat2str(i);
        obj.nome_arquivo =
strcat(dado_chuva,'\',obj.string_nome,obj.numero_arquivo,obj.string_complem);
        obj.intens_rain(:, :, i) = load(obj.nome_arquivo);
    end %end for

    obj.tam_rain = zeros(1,3);
    obj.tam_rain = size(obj.intens_rain);

    if obj.num_rain == 1
        obj.tam_rain(1,3) = 1;
    end %end if

end %end function

%Dados para o método que utiliza o cálculo do Fator de Segurança Chuva
%em horas ou dias
function obj = InputFS(obj,unidade_chuva)

    obj.unidade_chuva = unidade_chuva;

    switch unidade_chuva

        %Intensidade em horas
        case 1

            % transforma a intensidade de chuva da unidade de mm/h para m/s
            obj.intens_rain = 2.78e-7*obj.intens_rain;
            obj.time_rain = 3600*ones(1,obj.tam_rain(1,3));

            %Intensidade em dias
            case 2

            % transforma a intensidade de chuva da unidade de mm/dia para m/s
            obj.intens_rain = 1.157e-8*obj.intens_rain;
            obj.time_rain = 86400*ones(1,obj.tam_rain(1,3));
        end %end switch

    end %end function

% Função para o caso que se está utilizando uma precipitação
% genérica (por exemplo, W mm de chuva por X min)
function obj = InputFS_MIN(obj,tempo)

    obj.intens_rain = (1/(1000*60*tempo))*obj.intens_rain;
    obj.time_rain = (tempo*60)*ones(1,obj.tam_rain(1,3));

end %end function

%gera as posições da matriz aleatoriamente mediante o número de
%pixels estabelecido pelo usuário
%matriz nX2, onde n é n° de células que o usuário entrou
function obj = Input_Choise_Pixel(obj,number)

    obj.rdm_pix(:,1) = randi(obj.tam(1),number,1);
    obj.rdm_pix(:,2) = randi(obj.tam(2),number,1);
    obj.number_pixel = number;

end % end function

%load array of dynamic variables
function obj = Vector_Dynamic(obj)

    obj.vector_T = obj.input_dynamic(1,1):(obj.input_dynamic(1,2)-
obj.input_dynamic(1,1))/(obj.input_dynamic(1,3)-1):obj.input_dynamic(1,2);

    % transforma os valores de T de horas para segundos
    obj.vector_T = 3600*obj.vector_T;

    obj.vector_Iz = obj.input_dynamic(2,1):(obj.input_dynamic(2,2)-
obj.input_dynamic(2,1))/(obj.input_dynamic(2,3)-1):obj.input_dynamic(2,2);

```

```

% transforma os valores de Iz de mm/h para m/s
obj.vector_Iz = 2.78e-7*obj.vector_Iz;

obj.vector_Iz_std = obj.input_dynamic(3,1):(obj.input_dynamic(3,2)-
obj.input_dynamic(3,1))/(obj.input_dynamic(3,3)-1):obj.input_dynamic(3,2);

% transforma os valores de Iz (steady) de mm/mes para m/s
obj.vector_Iz_std = 3.858e-10*obj.vector_Iz_std;

obj.dimen_vect_dyn(1) = size(obj.vector_Iz_std,2);
obj.dimen_vect_dyn(2) = size(obj.vector_Iz,2);
obj.dimen_vect_dyn(3) = size(obj.vector_T,2);

end %end function

%MÉTODOS DE CÁLCULO DO MODELO
%
% Cálculo do modelo para a opção de número de pixels fornecido pelo usuário
function obj = Model_FS_Pixel(obj,FS_crit)
%Iz_steady
for i = 1:obj.dimen_vect_dyn(1)

    for l = 1:obj.number_pixel
        obj.Zt_d =
(obj.vector_Iz_std(i)/obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),5))*obj.matrix_A
b(obj.rdm_pix(1,1),obj.rdm_pix(1,2))/(sin(pi*obj.matrix_decliv(obj.rdm_pix(1,1),obj.rdm_pix(1,2))/180)*cos(
pi*obj.matrix_decliv(obj.rdm_pix(1,1),obj.rdm_pix(1,2))/180));
        obj.Psi_Z_0(1) =
obj.Zt_d*(cos(pi*obj.matrix_decliv(obj.rdm_pix(1,1),obj.rdm_pix(1,2))/180))^2;
        end % end for

%Iz
for j = 1:obj.dimen_vect_dyn(2)

    % T
    for k = 1:obj.dimen_vect_dyn(3)
        obj.cont = 0;

        for l = 1:obj.number_pixel

            obj.T_ast =
obj.vector_T(k)/(((obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),1))^2)/(4*obj.matri
x_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),7)*cos(pi*obj.matrix_decliv(obj.rdm_pix(1,1),ob
j.rdm_pix(1,2))/180)^2));

            obj.R = sqrt(obj.T_ast/pi)*exp(-1/obj.T_ast)-erfc(1/sqrt(obj.T_ast));

            obj.Psi_Z_T =
obj.Psi_Z_0(1)+obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),1)*obj.vector_Iz(j)*obj
.R/obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),6);

            obj.FS =
tan(obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),3)*pi/180)/tan(obj.matrix_decliv(o
bj.rdm_pix(1,1),obj.rdm_pix(1,2))*pi/180)+(obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1
,2)),4)-
obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),3)*pi/1
80))/(obj.matrix_static(obj.matrix_area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),2)*obj.matrix_static(obj.matrix_
area(obj.rdm_pix(1,1),obj.rdm_pix(1,2)),1)*sin(obj.matrix_decliv(obj.rdm_pix(1,1),obj.rdm_pix(1,2))*pi/180)
*cos(obj.matrix_decliv(obj.rdm_pix(1,1),obj.rdm_pix(1,2))*pi/180));

            if obj.FS <= FS_crit

                obj.cont = obj.cont+1;

            end % end if
        end % end for
    end % end for

    % MODIFICAÇÃO EM 16/02/2013: PR DADA EM DECIMAL
    %obj.cont
    obj.result(j,k,i) = obj.cont/obj.number_pixel;

end % end for
end % end for
end % end for

end %end function

% Cálculo do modelo para a opção de NÚMERO TOTAL de pixels da área de estudo
function obj = Model_FS_Total(obj,FS_crit)

% número total de pixels
obj.number_pixel = obj.tam(1)*obj.tam(2);

% Laço referente a Iz (steady)

```

```

for i = 1:obj.dimen_vect_dyn(1)

    % Cálculo do Psi(Z,0) para toda área de estudo em relação a um valor de Iz (steady)
    for l = 1:obj.tam(1)
        for m = 1:obj.tam(2)
            obj.Zt_d =
(objj.vector_Iz_std(i)/obj.matrix_static(obj.matrix_area(1,m),5))*obj.matrix_Ab(1,m)/(sin(pi*obj.matrix_decliv(1,m)/180)*cos(pi*obj.matrix_decliv(1,m)/180));
            obj.Psi_Z_0(1,m) = obj.Zt_d*(cos(pi*obj.matrix_decliv(1,m)/180))^2;
        end % end for
    end % end for

    %Iz linha
    for j = 1:obj.dimen_vect_dyn(2)
        % T coluna
        for k = 1:obj.dimen_vect_dyn(3)

            obj.cont = 0;

            % Varre a área de estudo toda
            for l = 1:obj.tam(1)
                for m = 1:obj.tam(2)

                    obj.T_ast =
objj.vector_T(k)/(((obj.matrix_static(obj.matrix_area(1,m),1))^2)/(4*obj.matrix_static(obj.matrix_area(1,m),7)*cos(pi*obj.matrix_decliv(1,m)/180)^2));

                    obj.R = sqrt(obj.T_ast/pi)*exp(-1/obj.T_ast)-erfc(1/sqrt(obj.T_ast));

                    obj.Psi_Z_T =
obj.Psi_Z_0(1,m)+obj.matrix_static(obj.matrix_area(1,m),1)*objj.vector_Iz(j)*obj.R/obj.matrix_static(obj.matrix_area(1,m),6);

                    obj.FS =
tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.matrix_static(obj.matrix_area(1,m),4)-obj.Psi_Z_T*objj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180))/(obj.matrix_static(obj.matrix_area(1,m),2)*obj.matrix_static(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matrix_decliv(1,m)*pi/180));

                    if obj.FS <= FS_crit

                        obj.cont = obj.cont+1;

                    end % end if

                end % end for

            end % end for
            % MODIFICAÇÃO EM 16/02/2013: PR DADA EM DECIMAL
            obj.result(j,k,i) = obj.cont/obj.number_pixel;
        end %end for
    end %end for

end % end for
end %end function

function obj = Fosm(obj,num_Iz_st,cv_atrito,cv_c,cv_gama_s,cv_H,cv_K)

if obj.tam_rain(3) == 1

    obj.FS_med = zeros(obj.tam(1),obj.tam(2));
    obj.FS_desvio = zeros(obj.tam(1),obj.tam(2));

else

    obj.FS_med = zeros(obj.tam(1),obj.tam(2),obj.num_rain);
    obj.FS_desvio = zeros(obj.tam(1),obj.tam(2),obj.num_rain);

end % end if

% transforma os valores de Iz (steady) de mm/mes para m/s
objj.vector_dynamic = [3.858e-10*num_Iz_st];

%definição dos coeficientes de variação
%Columns-> Na sequência: Zt, gama_s,atrito, coesão, Kz
objj.vector_cv = [cv_H cv_gama_s cv_atrito cv_c cv_K];

% Aumento de 10% nos valores médios das variáveis Xi
objj.vector_incr = 1.1*obj.matrix_static;

obj.delta = objj.vector_incr - obj.matrix_static;

```

```

%Cálculo das variâncias
for i = 1:size(obj.matrix_static,1)
    for j = 1:4
        obj.varianc(i,j) = (obj.vector_cv(j)*obj.matrix_static(i,j)/100)^2;
    end % end for
end %end for

for i = 1:size(obj.matrix_static,1)
    obj.varianc(i,5) = (obj.vector_cv(5)*obj.matrix_static(i,6)/100)^2;
end % end for

% varre a área de estudo
for k = 1:obj.num_rain
    for l = 1:obj.tam(1)
        for m = 1:obj.tam(2)

            % Factor of Safety mean
            obj.Zt_d =
            (obj.vector_dynamic(1)/obj.matrix_static(obj.matrix_area(l,m),5))*obj.matrix_Ab(l,m)/(sin(pi*obj.matrix_decliv(l,m)/180)*cos(pi*obj.matrix_decliv(l,m)/180));
            obj.Psi_Z_0 = obj.Zt_d*(cos(pi*obj.matrix_decliv(l,m)/180))^2;

            obj.T_ast =
            obj.time_rain(k)/(((obj.matrix_static(obj.matrix_area(l,m),1))^2)/(4*obj.matrix_static(obj.matrix_area(l,m),7)*cos(pi*obj.matrix_decliv(l,m)/180)^2));

            obj.R = sqrt(obj.T_ast/pi)*exp(-1/obj.T_ast)-erfc(1/sqrt(obj.T_ast));

            obj.Psi_Z_T = obj.Psi_Z_0 +
            obj.matrix_static(obj.matrix_area(l,m),1)*obj.intens_rain(l,m,k)*obj.R/obj.matrix_static(obj.matrix_area(l,m),6);

            % FS_fosm MEAN-> 3ª dimension with index equal 1
            obj.FS_fosm(l,m,1) =
            tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180)/tan(obj.matrix_decliv(l,m)*pi/180)+(obj.matrix_static(obj.matrix_area(l,m),4)-
            obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180))/(obj.matrix_static(obj.matrix_area(l,m),2)*obj.matrix_static(obj.matrix_area(l,m),1)*sin(obj.matrix_decliv(l,m)*pi/180)*cos(obj.matrix_decliv(l,m)*pi/180));

            %Incrementos
            % FS_fosm com incremento do atrito -> 3ª dimension with index equal 2
            obj.FS_fosm(l,m,2) =
            tan(obj.vector_incr(obj.matrix_area(l,m),3)*pi/180)/tan(obj.matrix_decliv(l,m)*pi/180)+(obj.matrix_static(obj.matrix_area(l,m),4)-
            obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.vector_incr(obj.matrix_area(l,m),3)*pi/180))/(obj.matrix_static(obj.matrix_area(l,m),2)*obj.matrix_static(obj.matrix_area(l,m),1)*sin(obj.matrix_decliv(l,m)*pi/180)*cos(obj.matrix_decliv(l,m)*pi/180));

            % FS_fosm com incremento da coesão -> 3ª dimension with index equal 3
            obj.FS_fosm(l,m,3) =
            tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180)/tan(obj.matrix_decliv(l,m)*pi/180)+(obj.vector_incr(obj.matrix_area(l,m),4)-
            obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180))/(obj.matrix_static(obj.matrix_area(l,m),2)*obj.matrix_static(obj.matrix_area(l,m),1)*sin(obj.matrix_decliv(l,m)*pi/180)*cos(obj.matrix_decliv(l,m)*pi/180));

            % FS_fosm com incremento do peso específico -> 3ª dimension with index equal 4
            obj.FS_fosm(l,m,4) =
            tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180)/tan(obj.matrix_decliv(l,m)*pi/180)+(obj.matrix_static(obj.matrix_area(l,m),4)-
            obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180))/(obj.vector_incr(obj.matrix_area(l,m),2)*obj.matrix_static(obj.matrix_area(l,m),1)*sin(obj.matrix_decliv(l,m)*pi/180)*cos(obj.matrix_decliv(l,m)*pi/180));

            % FS_fosm com incremento da permeabilidade Kz -> 3ª dimension with index equal 5
            obj.Psi_Z_T = obj.Psi_Z_0 +
            obj.matrix_static(obj.matrix_area(l,m),1)*obj.intens_rain(l,m,k)*obj.R/obj.vector_incr(obj.matrix_area(l,m),6);

            obj.FS_fosm(l,m,5) =
            tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180)/tan(obj.matrix_decliv(l,m)*pi/180)+(obj.matrix_static(obj.matrix_area(l,m),4)-
            obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(l,m),3)*pi/180))/(obj.matrix_static(obj.matrix_area(l,m),2)*obj.matrix_static(obj.matrix_area(l,m),1)*sin(obj.matrix_decliv(l,m)*pi/180)*cos(obj.matrix_decliv(l,m)*pi/180));

            % FS_fosm com incremento da espessura do solo -> 3ª dimension with index equal 6
            obj.T_ast =
            obj.time_rain(k)/(((obj.vector_incr(obj.matrix_area(l,m),1))^2)/(4*obj.matrix_static(obj.matrix_area(l,m),7)*cos(pi*obj.matrix_decliv(l,m)/180)^2));

            obj.R = sqrt(obj.T_ast/pi)*exp(-1/obj.T_ast)-erfc(1/sqrt(obj.T_ast));

            obj.Psi_Z_T = obj.Psi_Z_0 +
            obj.vector_incr(obj.matrix_area(l,m),1)*obj.intens_rain(l,m,k)*obj.R/obj.matrix_static(obj.matrix_area(l,m),6);
        end
    end
end

```

```

        obj.FS_fosm(1,m,6) =
tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.matrix_static
(obj.matrix_area(1,m),4)-
obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180))/(obj.matrix_static(obj.ma
trix_area(1,m),2)*obj.vector_incr(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matrix
_decliv(1,m)*pi/180));

        %Cálculo do desvio padrão

        obj.FS_desvio(1,m,k)= sqrt(((obj.FS_fosm(1,m,2)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),3))^2*obj.varianc(obj.matrix_area(1,m),3) +
((obj.FS_fosm(1,m,3)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),4))^2*obj.varianc(obj.matrix_area(1,m),4) +
((obj.FS_fosm(1,m,4)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),2))^2*obj.varianc(obj.matrix_area(1,m),2) +
((obj.FS_fosm(1,m,5)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),6))^2*obj.varianc(obj.matrix_area(1,m),5) +
((obj.FS_fosm(1,m,6)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),1))^2*obj.varianc(obj.matrix_area(1,m),1));
        obj.FS_med(1,m,k) = obj.FS_fosm(1,m,1);

        end % end for
        end % end for
    end % end for

end % end function

% FUNÇÃO PARA CÁLCULO DA PR E DO ÍNDICE DE CONFIABILIDADE
% Função que calcula a PR a partir dos valores do FS médio e do
% desvio-padrão
function obj = ProbRup(obj,quant_anal,FS_crit)

    obj.quant_analis = quant_anal;
    obj.Zcrit = zeros(obj.tam(1),obj.tam(2));
    obj.INF = obj.FS_med == Inf;
    obj.NINF = obj.FS_med ~= Inf;

    clear obj.PR;

    % Para todos os passos de chuva
    if obj.quant_analis == 1

        % Laço para apenas 1 matriz de chuva (QUANDO SE TEM APENAS 1 CHUVA)
        if obj.num_rain == 1

            obj.PR = zeros(obj.tam(1),obj.tam(2));

            obj.Zcrit = real((FS_crit-obj.FS_med(:,:))./obj.FS_desvio(:,:));

            for i = 1:obj.tam(1)
                for j = 1:obj.tam(2)

                    if obj.FS_med(i,j) ~= Inf

                        x = -10:0.01:obj.Zcrit(i,j);
                        Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);
                        obj.PR(i,j) = trapz(x,Y,2);

                    end % end if

                end % end for
            end % end for

        % PARA TODOS OS PASSOS DE CHUVA (QUANDO SE TEM MAIS DE 1)
        else

            obj.PR = zeros(obj.tam(1),obj.tam(2),obj.num_rain);
            obj.Zcrit = zeros(obj.tam(1),obj.tam(2));

            for k = 1:obj.num_rain

                obj.Zcrit = real((FS_crit-obj.FS_med(:,:,k))./obj.FS_desvio(:,:,k));

                for i = 1:obj.tam(1)
                    for j = 1:obj.tam(2)

                        if obj.FS_med(i,j,k) ~= Inf

                            x = -10:0.01:obj.Zcrit(i,j);
                            Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

```

```

        obj.PR(i,j,k) = trapz(x,Y,2);
    end % end if

    end % end for
end % end for

end %end for

end %end if

% para o primeiro e o último passo
elseif obj.quant_analis == 2

obj.PR = zeros(obj.tam(1),obj.tam(2),2);
obj.cont = 1;

for k = 1:obj.num_rain-1:obj.num_rain

obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.FS_desvio(:,:,k);

for i = 1:obj.tam(1)
for j = 1:obj.tam(2)

if obj.FS_med(i,j,k) ~= Inf

x = -10:0.01:obj.Zcrit(i,j);

Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

obj.PR(i,j,obj.cont) = trapz(x,Y,2);

end % end if

end % end for
end % end for

obj.cont = obj.cont+1;

end %end for
end % end if

end %end function

% Função para o CÁLCULO DA PR PARA PASSOS DEFINIDOS POR ARQUIVO
function obj = ProbRupArquivo(obj,arq_analis,FS_crit)

obj.arq_analis = arq_analis;

obj.PR = zeros(obj.tam(1),obj.tam(2),size(obj.arq_analis,2));
obj.cont = 1;

while obj.cont < size(obj.arq_analis)+1

k = obj.arq_analis(obj.cont);

obj.Zcrit = real((FS_crit-obj.FS_med(:,:,k))./obj.FS_desvio(:,:,k));

for i = 1:obj.tam(1)
for j = 1:obj.tam(2)

if obj.FS_med(i,j,k) ~= Inf

x = -10:0.01:obj.Zcrit(i,j);

Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

obj.PR(i,j,obj.cont) = trapz(x,Y,2);

end % end if

end % end for
end % end for

obj.cont = obj.cont+1;

end % end while

end % end fuction

% MÉTODOS DE GERAÇÃO DE RESULTADOS
%
% função para determinar os valores de probabilidade de ruptura a
% partir da matriz de resultados

```

```

function obj = Thresholds(obj,Iz_est_PR,opc_analis,dummy)

Iz_est_PR = 3.858e-10*Iz_est_PR;

[obj.T,obj.Iz] = meshgrid(obj.vector_T,obj.vector_Iz);

% TODOS OS PASSOS PARA ENCONTRAR OS VALORES PR
if opc_analis == 1

    obj.Prob_Rup = zeros(obj.tam_rain(1),obj.tam_rain(2),obj.num_rain);

    % Busca binária para encontrar posição de Iz (steady) escolhido
    % pelo usuário no vetor correspondente
    obj.lim_inf = 0;
    obj.lim_sup = size(obj.vector_Iz_std,2);

    while obj.lim_inf <= obj.lim_sup

        obj.med = (obj.lim_inf + obj.lim_sup)/2;

        if obj.vector_Iz_std(obj.med) == Iz_est_PR
            obj.Buscado = obj.med;
            break

        else if obj.vector_Iz_std(obj.med) < Iz_est_PR
            obj.lim_inf = obj.med+1;
        else
            obj.lim_sup = obj.med-1;
        end %end if

    end % end if

end % end while

%laço referente aos diversos planos de intensidade de chuva
for j = 1:obj.num_rain
    %varre as linhas da área de estudo
    for k = 1:obj.tam_rain(1)
        %varre as colunas da área de estudo
        for l = 1:obj.tam_rain(2)

            if obj.intens_rain(k,l,j) ~= dummy
                % REALIZEI MODIFICAÇÃO NO TERMO obj.intens_rain(k,l) PARA obj.intens_rain(k,l,j)
                % EM 06/02/13
                obj.Prob_Rup(k,l,j) =
interp2(obj.T,obj.Iz,obj.result(:, :, obj.Buscado),obj.time_rain(j),obj.intens_rain(k,l,j));

            else
                obj.Prob_Rup(k,l,j) = dummy;
            end % end if

        end % end for
    end %end for
end %end for

% apenas o 1° e o último PASSO DE CHUVA
else

    obj.Prob_Rup = zeros(obj.tam_rain(1),obj.tam_rain(2),2);

    % Busca binária para encontrar posição de Iz (steady) escolhido
    % pelo usuário no vetor correspondente
    obj.lim_inf = 0;
    obj.lim_sup = size(obj.vector_Iz_std,2);

    while obj.lim_inf <= obj.lim_sup

        obj.med = (obj.lim_inf + obj.lim_sup)/2;

        if obj.vector_Iz_std(obj.med) == Iz_est_PR
            obj.Buscado = obj.med;
            break

        else if obj.vector_Iz_std(obj.med) < Iz_est_PR
            obj.lim_inf = obj.med+1;
        else
            obj.lim_sup = obj.med-1;
        end %end if

    end % end if

end % end while

%varre as linhas da área de estudo
for k = 1:obj.tam_rain(1)

```

```

%varre as colunas da área de estudo
for l = 1:obj.tam_rain(2)
% REALIZEI MODIFICAÇÃO NO TERMO obj.intens_rain(k,l) PARA obj.intens_rain(k,l,j)
% EM 06/02/13
% PARA O PRIMEIRA MATRIZ DE CHUVA
if obj.intens_rain(k,l,1) ~= dummy
obj.Prob_Rup(k,l,1) =
interp2(obj.T,obj.Iz,obj.result(:, :,obj.Buscado),obj.time_rain(1),obj.intens_rain(k,l,1));
else
obj.Prob_Rup(k,l,1) = dummy;
end % endi if

% PARA A ÚLTIMA MATRIZ DE CHUVA
if obj.intens_rain(k,l,1) ~= dummy
obj.Prob_Rup(k,l,2) =
interp2(obj.T,obj.Iz,obj.result(:, :,obj.Buscado),obj.time_rain(obj.num_rain),obj.intens_rain(k,l,obj.num_ra
in));
else
obj.Prob_Rup(k,l,2) = dummy;
end % end if

end % end for
end %end for

end % end if

%
end % end function

function obj = Thresholds_Arq(obj, Iz_est_PR, Arq_Dados, dummy)

obj.Prob_Rup = zeros(obj.tam_rain(1),obj.tam_rain(2),obj.size(Arq_Dados,2));

% Busca binária para encontrar posição de Iz (steady) escolhido
% pelo usuário no vetor correspondente
obj.lim_inf = 0;
obj.lim_sup = size(obj.vector_Iz_std,2);

while obj.lim_inf <= obj.lim_sup

obj.med = (obj.lim_inf + obj.lim_sup)/2;

if obj.vector_Iz_std(obj.med) == Iz_est_PR
obj.Buscado = obj.med;
break

else if obj.vector_Iz_std(obj.med) < Iz_est_PR
obj.lim_inf = obj.med+1;
else
obj.lim_sup = obj.med-1;
end %end if

end % end if
end % end while

for i = 1:size(Arq_Dados)

for k = 1:obj.tam_rain(1)
%varre as colunas da área de estudo
for l = 1:obj.tam_rain(2)
% REALIZEI MODIFICAÇÃO NO TERMO obj.intens_rain(k,l) PARA obj.intens_rain(k,l,j)
% EM 06/02/13

if obj.intens_rain(k,l,i) ~= dummy
obj.Prob_Rup(k,l,i) =
interp2(obj.T,obj.Iz,obj.result(:, :,obj.Buscado),obj.time_rain(i),obj.intens_rain(k,l,i));
else
obj.Prob_Rup(k,l,i) = dummy;
end % end if

end % end for
end %end for
end % end for

end % end function

end %end methods
end %end class

classdef Input_Debris_Probability
%UNTITLED Summary of this class goes here
% Detailed explanation goes here
properties
% matriz com as divisões homogêneas dos distritos da bacia (resultante do SIG)
matrix_area;

```

```

% matrix of static input variables
% Rows-> distritos distintos
% Columns-> Na sequência: Zt, gama_s, atrito, coesão, Kx, Kz, Do

matrix_static;

% matrix of topographic variables
matrix_decliv;
matrix_Ab;
tam;

%Random array of pixels
rdm_pix;

%number of pixel
number_pixel;

% dynamic input variables
%
% matriz com os dados das variáveis dinâmicas (3X3)
% Rows: 1->data of T, 2->data of Iz; 3->data of Iz(steady)
%Columns: 1->minimum value, 2->maximum value,3->number of value
input_dynamic;
vector_T;
vector_Iz;
vector_Iz_std;
%dimensões dos vetores das variáveis dinâmicas
% columns 1: vetor Iz_steady; columns 2: vetor Iz; columns 1: vetor T;
dimen_vect_dyn;

%Model properties
Zt_d;
Psi_Z_0;
T_ast;
R;
Psi_Z_T;
FS;
cont;
result;
T;
Iz;

% Input Rain
%
% number of data rain
num_rain;
unidade_chuva;
delta_t;
%matriz multidimensional referente aos dados de intensidade de chuva(cada dimensão corresponde os
dados de intensidade de chuva da região em estudo para um determinado tempo)
intens_rain;
%vetor que traz os tempos nos quais foram tomados os valores de intensidade de chuva
time_rain;
tam_rain;
string_nome;
string_complem;
numero_arquivo;
nome_arquivo;
Prob_Rup;
arq_analis;

%Input probabilistic method
%
% Vetor para os valores de Iz-steady, Iz e T
vector_dynamic;
% Vetor para os valores de coeficiente de variação
vector_cv;
% Vetor com os incrementos (+10%) dos valores médios da matrix static
vector_incr;
delta;
varianc;
FS_fosm;
FS_desvio;
PR;

quant_analis;
Zcrit;
INF;
NINF;
FS_med;

end % end properties

properties(Constant = true)
% Peso específico da água em N/m³

```

```

Peso_esp_w = 9800;

end % end properties constant

methods

%MÉTODOS DE INICIALIZAÇÃO DO OBJETO E ENTRADA DE DADOS
%
% inicializa objeto
function obj = Input_Debris_Probability(area_estudo,declividade,drenagem_bacia,matrix_static)

    % data of GIS (SPRING)
    obj.matrix_area = area_estudo;
    obj.matrix_decliv = declividade;
    obj.matrix_Ab = drenagem_bacia;

    obj.matrix_static = matrix_static;

    % transforma gama_solo da unidade de kN/m³ para N/m³
    obj.matrix_static(:,2) = 1000*obj.matrix_static(:,2);

    % transforma a coesão da unidade de kPa para Pa
    obj.matrix_static(:,4) = 1000*obj.matrix_static(:,4);

    % transforma Kx da unidade de mm/s para m/s
    obj.matrix_static(:,5) = obj.matrix_static(:,5)/1000;

    % transforma Kz da unidade de mm/s para m/s
    obj.matrix_static(:,6) = obj.matrix_static(:,6)/1000;

    % transforma Do da unidade de mm²/s para m²/s
    obj.matrix_static(:,7) = obj.matrix_static(:,7);

    obj.tam = size(obj.matrix_area);

end % end function

% Lê os dados de chuva e salva-os em uma matriz multidimensional
%Deve-se passar o número de passos de chuva como argumento
function obj = Input_Rain(obj,num_rain,dado_chuva)

    obj.num_rain = num_rain;

    obj.string_nome = 'input_chuva';
    obj.string_complem = '.txt';

    for i =1:obj.num_rain
        obj.numero_arquivo = mat2str(i);
        obj.nome_arquivo =
strcat(dado_chuva,'\',obj.string_nome,obj.numero_arquivo,obj.string_complem);
        obj.intens_rain(:,i) = load(obj.nome_arquivo);
    end %end for

    obj.tam_rain = zeros(1,3);

    obj.tam_rain = size(obj.intens_rain);

    if obj.num_rain == 1
        obj.tam_rain(1,3) = 1;
    end %end if

end %end function

%Dados para o método que utiliza o cálculo do Fator de Segurança Chuva
%em horas ou dias
function obj = InputFS(obj,unidade_chuva)

    obj.unidade_chuva = unidade_chuva;

    switch unidade_chuva

        %Intensidade em horas
        case 1

            % transforma a intensidade de chuva da unidade de mm/h para m/s
            obj.intens_rain = 2.78e-7*obj.intens_rain;
            obj.time_rain = 3600*ones(1,obj.tam_rain(1,3));

            %Intensidade em dias
            case 2
            % transforma a intensidade de chuva da unidade de mm/dia para m/s
            obj.intens_rain = 1.157e-8*obj.intens_rain;
            obj.time_rain = 86400*ones(1,obj.tam_rain(1,3));
        end %end switch

```

```

end %end function

% Função para o caso que se está utilizando uma precipitação
% genérica (por exemplo, W mm de chuva por X min)
function obj = InputFS_MIN(obj,tempo)

    obj.intens_rain = (1/(1000*60*tempo))*obj.intens_rain;
    obj.time_rain = (tempo*60)*ones(1,obj.tam_rain(1,3));

end %end function

%MÉTODOS DE CÁLCULO DO MODELO
%
function obj = Fosm(obj,num_Iz_st,cv_atrito,cv_c,cv_gama_s,cv_H,cv_K)

if obj.tam_rain(3) == 1

    obj.FS_med = zeros(obj.tam(1),obj.tam(2));
    obj.FS_desvio = zeros(obj.tam(1),obj.tam(2));

else

    obj.FS_med = zeros(obj.tam(1),obj.tam(2),obj.num_rain);
    obj.FS_desvio = zeros(obj.tam(1),obj.tam(2),obj.num_rain);

end % end if

% transforma os valores de Iz (steady) de mm/mes para m/s
obj.vector_dynamic = [3.858e-10*num_Iz_st];

%definição dos coeficientes de variação
%Columns-> Na sequência: Zt, gama_s,atrito, coesão, Kz
obj.vector_cv = [cv_H cv_gama_s cv_atrito cv_c cv_K];

% Aumento de 10% nos valores médios das variáveis Xi
obj.vector_incr = 1.1*obj.matrix_static;

obj.delta = obj.vector_incr - obj.matrix_static;

%Cálculo das variâncias
for i = 1:size(obj.matrix_static,1)
    for j = 1:4
        obj.varianc(i,j) = (obj.vector_cv(j)*obj.matrix_static(i,j)/100)^2;
    end % end for
end %end for

for i = 1:size(obj.matrix_static,1)
    obj.varianc(i,5) = (obj.vector_cv(5)*obj.matrix_static(i,6)/100)^2;
end % end for
% varre a área de estudo
for k =1:obj.num_rain
    for l = 1:obj.tam(1)
        for m = 1:obj.tam(2)

            % Factor of Safety mean
            obj.Zt_d =
(obj.vector_dynamic(1)/obj.matrix_static(obj.matrix_area(1,m),5))*obj.matrix_Ab(1,m)/(sin(pi*obj.matrix_decliv(1,m)/180)*cos(pi*obj.matrix_decliv(1,m)/180));
            obj.Psi_Z_0 = obj.Zt_d*(cos(pi*obj.matrix_decliv(1,m)/180))^2;

            obj.T_ast =
obj.time_rain(k)/(((obj.matrix_static(obj.matrix_area(1,m),1))^2)/(4*obj.matrix_static(obj.matrix_area(1,m),7)*cos(pi*obj.matrix_decliv(1,m)/180)^2));

            obj.R = sqrt(obj.T_ast/pi)*exp(-1/obj.T_ast)-erfc(1/sqrt(obj.T_ast));

            obj.Psi_Z_T = obj.Psi_Z_0 +
obj.matrix_static(obj.matrix_area(1,m),1)*obj.intens_rain(1,m,k)*obj.R/obj.matrix_static(obj.matrix_area(1,m),6);

            % FS_fosm MEAN-> 3ª dimension with index equal 1
            obj.FS_fosm(1,m,1) =
tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.matrix_static(obj.matrix_area(1,m),4)-obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180))/(obj.matrix_static(obj.matrix_area(1,m),2)*obj.matrix_static(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matrix_decliv(1,m)*pi/180));

            %Incrementos
            % FS_fosm com incremento do atrito -> 3ª dimension with index equal 2
            obj.FS_fosm(1,m,2) =
tan(obj.vector_incr(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.matrix_static(obj.matrix_area(1,m),4)-obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.vector_incr(obj.matrix_area(1,m),3)*pi/180))/(obj.matrix_static(obj.matrix_area(1,m),2)*obj.matrix_static(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matrix_decliv(1,m)*pi/180));

        end
    end
end

```

```

ix_area(1,m),2)*obj.matrix_static(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matrix
_decliv(1,m)*pi/180));

    % FS_fosm com incremento da coesão -> 3ª dimension with index equal 3
    obj.FS_fosm(1,m,3) =
tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.vector_incr(o
bj.matrix_area(1,m),4)-
obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180))/(obj.matrix_static(obj.ma
trix_area(1,m),2)*obj.matrix_static(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matr
ix_decliv(1,m)*pi/180));

    % FS_fosm com incremento do peso específico -> 3ª dimension with index equal 4
    obj.FS_fosm(1,m,4) =
tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.matrix_static
(obj.matrix_area(1,m),4)-
obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180))/(obj.vector_incr(obj.matr
ix_area(1,m),2)*obj.matrix_static(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matrix
_decliv(1,m)*pi/180));

    % FS_fosm com incremento da permeabilidade Kz -> 3ª dimension with index equal 5
    obj.Psi_Z_T = obj.Psi_Z_0 +
obj.matrix_static(obj.matrix_area(1,m),1)*obj.intens_rain(1,m,k)*obj.R/obj.vector_incr(obj.matrix_area(1,m)
,6);
    obj.FS_fosm(1,m,5) =
tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.matrix_static
(obj.matrix_area(1,m),4)-
obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180))/(obj.matrix_static(obj.ma
trix_area(1,m),2)*obj.matrix_static(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matr
ix_decliv(1,m)*pi/180));
    % FS_fosm com incremento da espessura do solo -> 3ª dimension with index equal 6
    obj.T_ast =
obj.time_rain(k)/((obj.vector_incr(obj.matrix_area(1,m),1))^2)/(4*obj.matrix_static(obj.matrix_area(1,m),7)
*cos(pi*obj.matrix_decliv(1,m)/180)^2));
    obj.R = sqrt(obj.T_ast/pi)*exp(-1/obj.T_ast)-erfc(1/sqrt(obj.T_ast));
    obj.Psi_Z_T = obj.Psi_Z_0 +
obj.vector_incr(obj.matrix_area(1,m),1)*obj.intens_rain(1,m,k)*obj.R/obj.matrix_static(obj.matrix_area(1,m)
,6);
    obj.FS_fosm(1,m,6) =
tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180)/tan(obj.matrix_decliv(1,m)*pi/180)+(obj.matrix_static
(obj.matrix_area(1,m),4)-
obj.Psi_Z_T*obj.Peso_esp_w*tan(obj.matrix_static(obj.matrix_area(1,m),3)*pi/180))/(obj.matrix_static(obj.ma
trix_area(1,m),2)*obj.vector_incr(obj.matrix_area(1,m),1)*sin(obj.matrix_decliv(1,m)*pi/180)*cos(obj.matrix
_decliv(1,m)*pi/180));
    %Cálculo do desvio padrão
    obj.FS_desvio(1,m,k) = sqrt(((obj.FS_fosm(1,m,2)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),3))^2*obj.varianc(obj.matrix_area(1,m),3) +
((obj.FS_fosm(1,m,3)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),4))^2*obj.varianc(obj.matrix_area(1,m),4) +
((obj.FS_fosm(1,m,4)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),2))^2*obj.varianc(obj.matrix_area(1,m),2) +
((obj.FS_fosm(1,m,5)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),6))^2*obj.varianc(obj.matrix_area(1,m),5) +
((obj.FS_fosm(1,m,6)-
obj.FS_fosm(1,m,1))/obj.delta(obj.matrix_area(1,m),1))^2*obj.varianc(obj.matrix_area(1,m),1));
    obj.FS_med(1,m,k) = obj.FS_fosm(1,m,1);
end % end for
end % end for
end % end for

end % end function

% FUNÇÃO PARA CÁLCULO DA PR E DO ÍNDICE DE CONFIABILIDADE
% Função que calcula a PR a partir dos valores do FS médio e do
% desvio-padrão
function obj = ProbRup(obj,quant_anal,FS_crit)

obj.quant_analis = quant_anal;
obj.Zcrit = zeros(obj.tam(1),obj.tam(2));
obj.INF = obj.FS_med == Inf;
obj.NINF = obj.FS_med ~= Inf;

clear obj.PR;

% Para todos os passos de chuva
if obj.quant_analis == 1

    % Laço para apenas 1 matriz de chuva (QUANDO SE TEM APENAS 1 CHUVA)
    if obj.num_rain == 1

        obj.PR = zeros(obj.tam(1),obj.tam(2));

        obj.Zcrit = real((FS_crit-obj.FS_med(:,:))./obj.FS_desvio(:,:));

```

```

for i = 1:obj.tam(1)
    for j = 1:obj.tam(2)

        if obj.FS_med(i,j) ~= Inf

            x = -10:0.01:obj.Zcrit(i,j);

            Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

            obj.PR(i,j) = trapz(x,Y,2);

        end % end if
    end % end for

end % end for

% PARA TODOS OS PASSOS DE CHUVA (QUANDO SE TEM MAIS DE 1)
else

    obj.PR = zeros(obj.tam(1),obj.tam(2),obj.num_rain);
    obj.Zcrit = zeros(obj.tam(1),obj.tam(2));

    for k = 1:obj.num_rain

        obj.Zcrit = real((FS_crit-obj.FS_med(:,:,k))./obj.FS_desvio(:,:,k));

        for i = 1:obj.tam(1)
            for j = 1:obj.tam(2)

                if obj.FS_med(i,j,k) ~= Inf

                    x = -10:0.01:obj.Zcrit(i,j);

                    Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

                    obj.PR(i,j,k) = trapz(x,Y,2);

                end % end if
            end % end for
        end % end for
    end %end for
end %end if

% para o primeiro e o último passo
elseif obj.quant_analis == 2

    obj.PR = zeros(obj.tam(1),obj.tam(2),2);
    obj.cont = 1;

    for k = 1:obj.num_rain-1:obj.num_rain

        obj.Zcrit = (FS_crit-obj.FS_med(:,:,k))./obj.FS_desvio(:,:,k);

        for i = 1:obj.tam(1)
            for j = 1:obj.tam(2)

                if obj.FS_med(i,j,k) ~= Inf

                    x = -10:0.01:obj.Zcrit(i,j);

                    Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

                    obj.PR(i,j,obj.cont) = trapz(x,Y,2);

                end % end if

            end % end for
        end % end for

        obj.cont = obj.cont+1;

    end %end for
end % end if

end %end function

% Função para o CÁLCULO DA PR PARA PASSOS DEFINIDOS POR ARQUIVO
function obj = ProbRupArquivo(obj,arq_analis,FS_crit)

    obj.arq_analis = arq_analis;

    obj.PR = zeros(obj.tam(1),obj.tam(2),size(obj.arq_analis,2));
    obj.cont = 1;

```

```

while obj.cont < size(obj.arq_analis)+1

    k = obj.arq_analis(obj.cont);

    obj.Zcrit = real((FS_crit-obj.FS_med(:, :, k))./obj.FS_desvio(:, :, k));

    for i = 1:obj.tam(1)
        for j = 1:obj.tam(2)

            if obj.FS_med(i, j, k) ~= Inf

                x = -10:0.01:obj.Zcrit(i, j);

                Y = 1/sqrt(2*pi)*exp(-0.5*x.^2);

                obj.PR(i, j, obj.cont) = trapz(x, Y, 2);

            end % end if

        end % end for
    end % end for

    obj.cont = obj.cont+1;

end % end

end % end function
end %end methods

end %end class

```

A.5 FUNÇÕES PARA CÁLCULO DA VULNERABILIDADE

A.5.1 INTERFACE GRÁFICA

```

function varargout = vulnerabilidade(varargin)
% VULNERABILIDADE M-file for vulnerabilidade.fig
%   VULNERABILIDADE, by itself, creates a new VULNERABILIDADE or raises the existing
%   singleton*.
%
%   H = VULNERABILIDADE returns the handle to a new VULNERABILIDADE or the handle to
%   the existing singleton*.
%
%   VULNERABILIDADE('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in VULNERABILIDADE.M with the given input arguments.
%
%   VULNERABILIDADE('Property','Value',...) creates a new VULNERABILIDADE or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before vulnerabilidade_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to vulnerabilidade_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help vulnerabilidade

% Last Modified by GUIDE v2.5 25-Sep-2014 21:28:57

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @vulnerabilidade_OpeningFcn, ...
                  'gui_OutputFcn',  @vulnerabilidade_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before vulnerabilidade is made visible.
function vulnerabilidade_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to vulnerabilidade (see VARARGIN)

% Choose default command line output for vulnerabilidade
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes vulnerabilidade wait for user response (see UIRESUME)
% uiwait(handles.figure1);

handles.axes1 = imread('imag_vulnerab.jpg');
image(handles.axes1);
axis off

% --- Outputs from this function are returned to the command line.
function varargout = vulnerabilidade_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton29.
function pushbutton29_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton29 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton30.
function pushbutton30_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton30 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton31.
function pushbutton31_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton31 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
global detrito_vetor;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text95, 'String', fullpathname);

detrito_vetor = fullfile(pathname, filename);

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
global veloc_vetor;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text97, 'String', fullpathname);

veloc_vetor = fullfile(pathname, filename);

% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
global detrito_area_estudo;
% hObject handle to pushbutton4 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector'});
fullpathname = strcat(pathname,filename);
set(handles.text98,'String',fullpathname);

detrito_area_estudo = fullfile(pathname,filename);

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
global veloc_area_estudo;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector'});
fullpathname = strcat(pathname,filename);
set(handles.text96,'String',fullpathname);

veloc_area_estudo = fullfile(pathname,filename);

% --- Executes on button press in pushbutton36.
function pushbutton36_Callback(hObject, eventdata, handles)
global person_lim_desloc;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector'});
fullpathname = strcat(pathname,filename);
set(handles.text74,'String',fullpathname);

person_lim_desloc = fullfile(pathname,filename);

% --- Executes on button press in pushbutton37.
function pushbutton37_Callback(hObject, eventdata, handles)
global person_altur;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector'});
fullpathname = strcat(pathname,filename);
set(handles.text99,'String',fullpathname);

person_altur = fullfile(pathname,filename);

% --- Executes on button press in pushbutton38.
function pushbutton38_Callback(hObject, eventdata, handles)
global person_manut;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector'});
fullpathname = strcat(pathname,filename);
set(handles.text77,'String',fullpathname);

person_manut = fullfile(pathname,filename);

% --- Executes on button press in pushbutton39.
function pushbutton39_Callback(hObject, eventdata, handles)
global person_tipo;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector'});
fullpathname = strcat(pathname,filename);
set(handles.text79,'String',fullpathname);

person_tipo = fullfile(pathname,filename);

% --- Executes on button press in pushbutton40.
function pushbutton40_Callback(hObject, eventdata, handles)
global matriz_lim_desloc_fund;
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

[filename pathname] = uigetfile({'*.*.','File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text82,'String',fullpathname);

matriz_lim_desloc_fund = fullfile(pathname,filename);

% --- Executes on button press in pushbutton41.
function pushbutton41_Callback(hObject, eventdata, handles)
global matriz_desloc_fund;
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text83,'String',fullpathname);

matriz_desloc_fund = fullfile(pathname,filename);

% --- Executes on button press in pushbutton42.
function pushbutton42_Callback(hObject, eventdata, handles)
global matriz_prof_fund;
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text85,'String',fullpathname);

matriz_prof_fund = fullfile(pathname,filename);

% --- Executes on button press in pushbutton43.
function pushbutton43_Callback(hObject, eventdata, handles)
global matriz_tipo_estrut;
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text90,'String',fullpathname);

matriz_tipo_estrut = fullfile(pathname,filename);

% --- Executes on button press in pushbutton44.
function pushbutton44_Callback(hObject, eventdata, handles)
global matriz_manutencao;
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text91,'String',fullpathname);

matriz_manutencao = fullfile(pathname,filename);

% --- Executes on button press in pushbutton45.
function pushbutton45_Callback(hObject, eventdata, handles)
global matriz_altura;
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*.','File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text92,'String',fullpathname);

matriz_altura = fullfile(pathname,filename);

% --- Executes on button press in pushbutton48.
function pushbutton48_Callback(hObject, eventdata, handles)
% BOTÃO SAIR

% hObject    handle to pushbutton48 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(get(hObject, 'parent'));

```

```

% --- Executes on button press in pushbutton49.
function pushbutton49_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton49 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global opcao;

% variáveis advindas do autômato
global veloc_automato
global detrito_automato;

global veloc_area_estudo;
global detrito_area_estudo;
global matriz_desloc_fund;
global matriz_lim_desloc_fund;
global VULNER;
global ctr_vulnerability;
global veloc_vetor;
global detrito_vetor;
global matriz_prof_fund;
global matriz_tipo_estrut;
global matriz_manutencao;
global matriz_altura;
global person_tipo;
global person_manut;
global person_lim_desloc;

switch opcao

    % ARQUIVOS ORGANIZADOS
    case 1

        % barra de status
        h = waitbar(0,'1','Name','Análise de Vulnerabilidade...',...
            'CreateCancelBtn',...
            'setappdata(gcf,'canceling',1)');
            setappdata(h,'canceling',0)

        steps = 4;

        entrada_vul =
        InputVulnerability(load(veloc_area_estudo),load(detrito_area_estudo),load(matriz_desloc_fund),load(matriz_1
im_desloc_fund));
        waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))
        entrada_vul =
        Input_Fator_Resist(entrada_vul,load(matriz_prof_fund),load(matriz_tipo_estrut),load(matriz_manutencao),load
(matriz_altura));
        waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))
        vulnerabilidade =
        Vulnerability(entrada_vul.posicao_velocidade,entrada_vul.posicao_zonas_impact,entrada_vul.posicao_Caract_Es
trutura,entrada_vul.matrix_Ds,entrada_vul.posicao_Dst,entrada_vul.tam);
        waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))
        vulnerabilidade = Calculo_Vulnerab(vulnerabilidade);
        waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))
        VULNER = vulnerabilidade.matrix_Vulner;

        delete(h)

        % Controle para saber que análise de vulnerabilidade foi realizada pelo menos 1 vez
        ctr_vulnerability = 1;

        warndlg('Análise da vulnerabilidade estabelecida com sucesso!','Operação realizada');

    case 2

        % barra de status
        h = waitbar(0,'1','Name','Análise de Vulnerabilidade...',...
            'CreateCancelBtn',...
            'setappdata(gcf,'canceling',1)');
            setappdata(h,'canceling',0)

        steps = 8;

        entrada_vul =
        InputVulnerability(load(veloc_area_estudo),load(detrito_area_estudo),load(matriz_desloc_fund),load(matriz_1
im_desloc_fund));
        waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))
        entrada_vul = Input_Velocidade_Deslizamento(entrada_vul,load(veloc_vetor));
        waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))
        entrada_vul = Input_Espessura_Detritos(entrada_vul,load(detrito_vetor));
        waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))

```

```

    entrada_vul =
Input_Fator_Resist(entrada_vul,load(matriz_prof_fund),load(matriz_tipo_estrut),load(matriz_manutencao),load
(matriz_altura));
    waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))
    entrada_vul = Input_Resistencia_default(entrada_vul);
    waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))
    entrada_vul = Input_Dst_default(entrada_vul);
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
    vulnerabilidade =
Vulnerability(entrada_vul.velocidade_desl,entrada_vul.zonas_impacto,entrada_vul.factor_resist,entrada_vul.
matrix_Ds,entrada_vul.matrix_Dst,entrada_vul.tam);
    waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
    vulnerabilidade = Calculo_Vulnerab(vulnerabilidade);
    waitbar(8/steps,h,sprintf('%0.f%% ',8/steps*100))
    VULNER = vulnerabilidade.matrix_Vulner;

delete(h)

% Controle para saber que análise de vulnerabilidade foi realizada pelo menos 1 vez
ctr_vulnerability = 1;

warndlg('Análise da vulnerabilidade estabelecida com sucesso!','Operação realizada');

case 3

% barra de status
h = waitbar(0,'1','Name','Análise de Vulnerabilidade...','...
'CreateCancelBtn',...
'setappdata(gcf,'canceling',1)');
setappdata(h,'canceling',0)

steps = 7;

    entrada_vul =
InputVulnerability(veloc_area_estudo,load(detrito_area_estudo),load(matriz_desloc_fund),load(matriz_1
im_desloc_fund));
    waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))
    entrada_vul = Input_Velocidade_Deslizamento(entrada_vul,veloc_vetor);
    waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))
    entrada_vul =
Input_Fator_Resist(entrada_vul,load(matriz_prof_fund),load(matriz_tipo_estrut),load(matriz_manutencao),load
(matriz_altura));
    waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))
    entrada_vul = Input_Espessura_Detritos(entrada_vul,load(detrito_vetor));
    waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))
    entrada_vul =
Input_Resistencia_Usuario(entrada_vul,load(person_tipo),load(person_manut),load(person_altur));
    waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))
    entrada_vul = Input_Dst_Usuario(entrada_vul,load(person_lim_desloc));
    waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
    vulnerabilidade =
Vulnerability(entrada_vul.velocidade_desl,entrada_vul.zonas_impacto,entrada_vul.factor_resist,entrada_vul.
matrix_Ds,entrada_vul.matrix_Dst,entrada_vul.tam);
    waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))
    vulnerabilidade = Calculo_Vulnerab(vulnerabilidade);
    waitbar(8/steps,h,sprintf('%0.f%% ',8/steps*100))
    VULNER = vulnerabilidade.matrix_Vulner;

delete(h)

% Controle para saber que análise de vulnerabilidade foi realizada pelo menos 1 vez
ctr_vulnerability = 1;

warndlg('Análise da vulnerabilidade estabelecida com sucesso!','Operação realizada');

% AUTÔMATOS CELULARES - ESTRUTURA PADRÃO
case 4

% barra de status
h = waitbar(0,'1','Name','Análise de Vulnerabilidade...','...
'CreateCancelBtn',...
'setappdata(gcf,'canceling',1)');
setappdata(h,'canceling',0)

steps = 6;

    entrada_vul =
InputVulnerability(veloc_automato,detrito_automato,load(matriz_desloc_fund),load(matriz_lim_desloc_fund));
    waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))
    entrada_vul =
Input_Fator_Resist(entrada_vul,load(matriz_prof_fund),load(matriz_tipo_estrut),load(matriz_manutencao),load
(matriz_altura));
    waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))
    entrada_vul = Input_Resistencia_default(entrada_vul);
    waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))
    entrada_vul = Input_Dst_default(entrada_vul);
    waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))

```

```

        vulnerabilidade =
Vulnerability(entrada_vul.velocidade_deslzl,entrada_vul.zonas_impacto,entrada_vul.factor_resist,entrada_vul.
matrix_Ds,entrada_vul.matrix_Dst,entrada_vul.tam);
        waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))
        vulnerabilidade = Calculo_Vulnerab(vulnerabilidade);
        waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
        VULNER = vulnerabilidade.matrix_Vulner;

delete(h)

        % Controle para saber que análise de vulnerabilidade foi realizada pelo menos 1 vez
        ctr_vulnerability = 1;

        warndlg('Análise da vulnerabilidade estabelecida com sucesso!','Operação realizada');

% AUTÔMATOS CELULARES - ESTRUTURA PERSONALIZADO
case 5

        % barra de status
        h = waitbar(0,'1','Name','Análise de Vulnerabilidade...',...
        'CreateCancelBtn',...
        'setappdata(gcf,'canceling',1)');
        setappdata(h,'canceling',0)

        steps = 6;

        entrada_vul =
InputVulnerability(veloc_automato,detrito_automato,load(matrix_desloc_fund),load(matrix_lim_desloc_fund));
        waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))
        entrada_vul =
Input_Fator_Resist(entrada_vul,load(matrix_prof_fund),load(matrix_tipo_estrut),load(matrix_manutencao),load
(matrix_altura));
        waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))
        entrada_vul =
Input_Resistencia_Usuario(entrada_vul,load(person_tipo),load(person_manut),load(person_altur));
        waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))
        entrada_vul = Input_Dst_Usuario(entrada_vul,load(person_lim_desloc));
        waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))
        vulnerabilidade =
Vulnerability(entrada_vul.velocidade_deslzl,entrada_vul.zonas_impacto,entrada_vul.factor_resist,entrada_vul.
matrix_Ds,entrada_vul.matrix_Dst,entrada_vul.tam);
        waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))
        vulnerabilidade = Calculo_Vulnerab(vulnerabilidade);
        waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))
        VULNER = vulnerabilidade.matrix_Vulner;

delete(h)

        % Controle para saber que análise de vulnerabilidade foi realizada pelo menos 1 vez
        ctr_vulnerability = 1;

        warndlg('Análise da vulnerabilidade estabelecida com sucesso!','Operação realizada');

end % end switch

% --- Executes on button press in pushbutton50.
function pushbutton50_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton50 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

clear strings

f = figure('menu','none','toolbar','none');
set(f,'name','Ajuda','numbertitle','off')

fid = fopen('vulner_help.txt');
ph = uipanel(f,'Units','normalized','position',[0.05 0.05 0.9 0.9],'title',...
'Modelos para Análise de Perigo');
lbh = uicontrol(ph,'style','listbox','Units','normalized','position',...
[0 0 1 1],'FontSize',9);

indic = 1;
while 1
    tline = fgetl(fid);
    if ~ischar(tline),
        break
    end
    strings{indic}=tline;
    indic = indic + 1;
end
fclose(fid);
set(lbh,'string',strings);
set(lbh,'Value',1);
set(lbh,'Selected','on');

```

```

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes on button press in pushbutton54.
function pushbutton54_Callback(hObject, eventdata, handles)
global ctrol_automato;
% hObject    handle to pushbutton54 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

if ctrol_automato == 0

    GUI_AUTOMATO_handle = GUI_AUTOMATO;

    set(handles.pushbutton56,'enable','on');

else
    resp = questdlg('Quer realizar outra análise usando os autômatos celulares?','ANÁLISE JÁ REALIZADA');

    if strcmp(resp,'Yes')

        GUI_AUTOMATO_handle = GUI_AUTOMATO;

        set(handles.pushbutton56,'enable','on');

    end % end if
end % end if

set(handles.text100,'String','VERIFIQUE A EXECUÇÃO DA ANÁLISE');

% --- Executes on button press in radiobutton10.
function radiobutton10_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton10

% --- Executes on button press in radiobutton9.
function radiobutton9_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function uipanel18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uipanel18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes when selected object is changed in uipanel18.
function uipanel18_SelectionChangeFcn(hObject, eventdata, handles)
global opcao;
% hObject    handle to the selected object in uipanel16
% eventdata  structure with the following fields (see UIBUTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

% INPUT ARQUIVOS ORGANIZADOS
if hObject == handles.radiobutton17
    % salva a escolha do usuário
    opcao = 1;

    % botões ligados
    set(handles.pushbutton14,'enable','on');
    set(handles.pushbutton13,'enable','on');
    set(handles.pushbutton49,'enable','on');
    % botões desligados

    set(handles.pushbutton12,'enable','off');
    set(handles.pushbutton11,'enable','off');
    set(handles.text100,'String','');
    set(handles.pushbutton54,'enable','off');

```

```

set(handles.pushbutton56,'enable','off');

% FATORES DE RESISTÊNCIA PERSONALIZADOS
set(handles.pushbutton39,'enable','off');
set(handles.pushbutton38,'enable','off');
set(handles.pushbutton37,'enable','off');
set(handles.pushbutton36,'enable','off');

% INPUT PADRÃO
elseif hObject == handles radiobutton12
% salva a escolha do usuário
opcao = 2;

% botões ligados
set(handles.pushbutton14,'enable','on');
set(handles.pushbutton13,'enable','on');
set(handles.pushbutton12,'enable','on');
set(handles.pushbutton11,'enable','on');
set(handles.pushbutton49,'enable','on');
% botão desligado
set(handles.text100,'String','');
set(handles.pushbutton54,'enable','off');
set(handles.pushbutton56,'enable','off');
% FATORES DE RESISTÊNCIA PERSONALIZADOS
set(handles.pushbutton39,'enable','off');
set(handles.pushbutton38,'enable','off');
set(handles.pushbutton37,'enable','off');
set(handles.pushbutton36,'enable','off');

% INPUT PERSONALIZADO
elseif hObject == handles radiobutton18
% salva a escolha do usuário
opcao = 3;

% botões ligados
set(handles.pushbutton14,'enable','on');
set(handles.pushbutton13,'enable','on');
set(handles.pushbutton12,'enable','on');
set(handles.pushbutton11,'enable','on');
set(handles.pushbutton39,'enable','on');
set(handles.pushbutton38,'enable','on');
set(handles.pushbutton37,'enable','on');
set(handles.pushbutton36,'enable','on');
set(handles.pushbutton49,'enable','on');
% botão desligado
set(handles.text100,'String','');
set(handles.pushbutton54,'enable','off');
set(handles.pushbutton56,'enable','off');

% INPUT AUTÔMATOS CELULARES PADRÃO
elseif hObject == handles radiobutton14
% salva a escolha do usuário
opcao = 4;

% botões desligados
set(handles.pushbutton14,'enable','off');
set(handles.pushbutton13,'enable','off');
set(handles.pushbutton12,'enable','off');
set(handles.pushbutton11,'enable','off');
set(handles.pushbutton56,'enable','off');
% botão ligado
set(handles.text100,'String','Execute as análises por meio dos autômatos celulares e verifique a
realização da simulação');
set(handles.pushbutton54,'enable','on');

% botões desligados
% FATORES DE RESISTÊNCIA PERSONALIZADOS
set(handles.pushbutton39,'enable','off');
set(handles.pushbutton38,'enable','off');
set(handles.pushbutton37,'enable','off');
set(handles.pushbutton36,'enable','off');
set(handles.pushbutton49,'enable','off');
elseif hObject == handles radiobutton19
% salva a escolha do usuário
opcao = 5;

% botões desligados
set(handles.pushbutton14,'enable','off');
set(handles.pushbutton13,'enable','off');
set(handles.pushbutton12,'enable','off');
set(handles.pushbutton11,'enable','off');
set(handles.pushbutton56,'enable','off');
set(handles.pushbutton49,'enable','off');
% botão ligado

```

```

        set(handles.text100,'String','Execute as análises por meio dos autômatos celulares e verifique a
realização da simulação');
        set(handles.pushbutton54,'enable','on');
        set(handles.pushbutton39,'enable','on');
        set(handles.pushbutton38,'enable','on');
        set(handles.pushbutton37,'enable','on');
        set(handles.pushbutton36,'enable','on');

end % end if

% --- Executes on button press in pushbutton55.
function pushbutton55_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton55 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function text100_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text100 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in pushbutton56.
function pushbutton56_Callback(hObject, eventdata, handles)
global ctrol_automato;
% hObject    handle to pushbutton56 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if ctrol_automato == 0

    set(handles.text100,'String','ANÁLISE POR AUTÔMATOS CELULARES NÃO EXECUTADA! PRESSIONE SIMULAÇÃO');

else

    set(handles.text100,'String','ANÁLISE POR AUTÔMATOS CELULARES REALIZADA! ANÁLISE DE VULNERABILIDADE
LIBERADA!');
    set(handles.pushbutton49,'enable','on');

end % end if

```

A.5.2 CLASSES ESPECÍFICAS PARA EXECUÇÃO DO MODELO

```

classdef InputVulnerability
%INPUTVULNERABILITY Summary of this class goes here
% Detailed explanation goes here

properties
    velocidade_deslz;
    posicao_velocidade;
    %Vetor com os valores de velocidade dos deslizamentos
    class_veloc;

    %Matriz com os valores de espessura de detritos já substituídos
    %prontos para serem utilizados no cálculo da vulnerabilidade
    zonas_impacto;

    posicao_zonas_impact;

    %Vetor de valores com a espessura das zonas impactadas para substituição
    %mediante a matriz de posição de zonas(cada elemento do vetor representa uma profundidade de
detritos)
    %1ª coluna - valor 1; 2ª coluna - valor 2,...
    espessura_impact;

    %Matriz com os valores dos fatores de resistência já substituídos
    %prontos para serem utilizados no cálculo da vulnerabilidade
    %Matriz multidimensional (4 dimensões)
    factor_resist;

    % matriz multidimensional (4 dimensões) da área de estudo com as diferentes características das
estruturas
    %Ordem das características: 1-Profundidade da fundação;2-Tipo de estrutura;
    %3-Estado de manutenção;4-Altura das construções
    posicao_Caract_Estrutura;

    %Vetores com os valores fornecidos pelo usuário dos fatores de resistência
    factor_Tipo_estrutura;
    factor_Estado_manutecao;
    factor_Altura;

    tam;

    % Matriz espacial dos valores dos índices de deslocamento da fundação

```

```

% que será fornecida pelo usuário
matrix_Ds;
% Matriz de posição dos valores limites de deslocamento da fundação (Dst)
%esta matriz poderá ser fornecida pelo usuário ou será criada
%a partir dos valores limites já estabelecidos
matrix_Dst;
posicao_Dst;

%Matriz coluna com os valores limites para deslocamento da fundação fornecido
%pelo usuário
Matrix_Limite_Fundacao;

end

%Valores default para os fatores de resistência de estruturas e para os valores
%limites de deslocamento de fundação (retirados do Artigo de Uzielli)
% Ressalta-se que a numeração dos tipos de estruturas presentes na matriz de posição deve ser
%realizada obedecendo a ordem aqui estabelecida e que leva em consideração a disposição do artigo:

%
%-----
%FATORES SALVOS EM VETORES (CADA COLUNA REPRESENTA UM TIPO DE ESTRUTURA)

%FATORES DE RESISTÊNCIA PARA TIPOS DA ESTRUTURA:
%1-ESTRUTURAS MAIS LEVES (CONSTRUÇÕES DE MADEIRA SIMPLES) LT-> 0.1
%2-ESTRUTURAS LEVES LS -> 0.2
%3-ESTRUTURAS MISTAS (CONCRETO E MADEIRA) MS -> 0.4
%4-PAREDES DE TIJOLOS, CONCRETO BC -> 0.8
%5-CONCRETO REFORÇADO RC -> 1.3
%6-REFORÇADO RF -> 1.5

%FATORES DE RESISTÊNCIA PARA ESTADO DE MANUTENÇÃO:
%1-MUITO POBRE VP -> 0.1
%2-POBRE P -> 0.4
%3-MÉDIO M -> 0.8
%4-BOM G -> 1.2
%5-MUITO BOM VG -> 1.5

%FATORES DE RESISTÊNCIA PARA ALTURA:
%1-ANDAR ÚNICO (1 ANDAR) S -> 0.1
%2-EDIFÍCIO BAIXO (2 ANDARES) L-> 0.4
%3-EDIFÍCIO MÉDIO (3,4,5 ANDARES) M-> 0.9
%4-EDIFÍCIO ALTO (6 OU + ANDARES) H-> 1.5

%
%-----
%MATRIZ DE VALORES LIMITES PARA O DESLOCAMENTO DE FUNDAÇÕES (CADA LINHA REPRESENTA OS TIPOS
% DE ESTRUTURA E AS CARACTERÍSTICAS DAS DEFORMAÇÕES E AS COLUNAS OS TIPOS DE SOLO)

%Características de deformação                Solo de fundação
%                Solo compressível de médio a baixo        Solo muito compressível

%
%-----
%1-Estrutura de alvenaria (inclinação parcial da fundação) em solo compressível de médio a baixo ->
0.002
%2-Estrutura de alvenaria (inclinação parcial da fundação) em solo muito compressível -> 0.003
%3-Estruturas em quadros com único andar (recalque - mm) -> 120 (apenas para solo compressível médio)
%4-Estruturas em quadros com único andar solo muito compressível(recalque - mm) -> 200
%5-Edifício arranha-céu com forma simples (recalque médio da fundação - mm) em solo compressível de
médio a baixo -> 200
%Edifícios industriais e civis comuns
%6-Estruturas em quadros (diferença de recalque entre estacas adjacentes - mm) em solo compressível de
médio a baixo-> 0.002
%7-Estruturas em quadros (diferença de recalque entre estacas adjacentes - mm) em solo muito
compressível-> 0.003
%8-Colunas externas com parede de alvenaria (diferença de recalque entre estacas adjacentes - mm) em
solo compressível de médio a baixo-> 0.0007
%9-Colunas externas com parede de alvenaria (diferença de recalque entre estacas adjacentes - mm) em
solo muito compressível-> 0.001
%Edifício Arranha-céu - Inclinação integral em solo compressível de médio a baixo (Hg - altura da
construção acima do solo)
%10-Hg<=24 ->0.004
%11-24<Hg<=60 ->0.003
%12-60<Hg<=100 ->0.0025
%13-Hg>100 ->0.002
%Estrutura de torre - inclinação da fundação em solo compressível de médio a baixo
%14-Hg<=20 ->0.008
%15-20<Hg<=50 -> 0.006
%16-50<Hg<=100 -> 0.005
%17-100<Hg<=150 -> 0.004
%18-150<Hg<=200 -> 0.003
%19-200<Hg<=250 -> 0.002
%Estrutura de torre - recalque da fundação (mm) em solo compressível de médio a baixo
%20-Hg<=100 -> 400
%21-100<Hg<=200 -> 300
%22-200<Hg<=250 -> 200

```

```

properties(Constant = true)
default_Tipo_str = [0.1 0.2 0.4 0.8 1.3 1.5];
default_State_Maint = [0.1 0.4 0.8 1.2 1.5];
default_Altura = [0.1 0.2 0.9 1.5];
% default_Dst = [0.002 0.003;120 200;200 0;0.002 0.003;0.0007 0.001;0.004 0;0.003 0;0.0025 0;0.002
0;0.008 0;0.006 0;0.005 0;0.004 0;0.003 0;0.002 0;400 0;300 0;200 0];
default_Dst = [0.002; 0.003; 120; 200; 200; 0.002; 0.003; 0.0007; 0.001; 0.004; 0.003; 0.0025; 0.002;
0.008; 0.006; 0.005; 0.004; 0.003; 0.002; 400; 300; 200]
end

methods

%Método para a inicialização e criação do objeto
%Este método será usado tanto no caso em que os dados já estejam
%organizados nos arquivos txt quanto na necessidade de serem organizados antes da sua utilização no
%cálculo da vulnerabilidade
function obj = InputVulnerability(posicao_velocidade,posicao_zonas_impact,matrix_Ds,posicao_Dst)

obj.posicao_velocidade = posicao_velocidade;
obj.posicao_zonas_impact = posicao_zonas_impact;
obj.matrix_Ds = matrix_Ds;
obj.posicao_Dst = posicao_Dst;
obj.tam = size(obj.posicao_velocidade);

end %end function

%Geração da matriz com as velocidades dos deslizamentos
function obj = Input_Velocidade_Deslizamento(obj,arquivo_velocidade_deslizamentos_substituicao)

obj.class_veloc = arquivo_velocidade_deslizamentos_substituicao;
%inicializo matriz de velocidade de deslizamentos com zeros e
%com as mesmas dimensões da área de estudo
obj.velocidade_desl = zeros(obj.tam(1),obj.tam(2));

for i = 1:size(obj.class_veloc,2)
    obj.velocidade_desl = obj.velocidade_desl + obj.class_veloc(i)*(obj.posicao_velocidade ==
i);
end % end for

end %end function

%Método que substitui os valores das classes de espessuras de
%detritos pelo real tamanho de profundidade de detritos
function obj = Input_Espessura_Detritos(obj,arquivo_profundidade_detritos_substituicao)

obj.espessura_impact = arquivo_profundidade_detritos_substituicao;

%inicializo matriz das zonas de impacto com zeros e
%com as mesmas dimensões da área de estudo
obj.zonas_impacto = zeros(obj.tam(1),obj.tam(2));

for i = 1:size(obj.espessura_impact,2)
    obj.zonas_impacto = obj.zonas_impacto + obj.espessura_impact(i)*(obj.posicao_zonas_impact ==
i);
end % end for

end %end function

%Método para importar e unir em uma única matriz multidimensional os dados dos fatores de
resistência das
%estruturas que fornecidos pelo usuário
function obj =
Input_Fator_Resist(obj,arquivo_profundidade_fundacao,arquivo_Matriz_Posicao_resistencia_tipo_estrutura,arq
ivo_Matriz_Posicao_resistencia_manutencao,arquivo_Matriz_Posicao_resistencia_altura)

obj.posicao_Caract_Estrutura = zeros(obj.tam(1),obj.tam(2),4);

obj.posicao_Caract_Estrutura(:, :, 1) = arquivo_profundidade_fundacao;

obj.posicao_Caract_Estrutura(:, :, 1) = (obj.posicao_Caract_Estrutura(:, :, 1) ./ 10) . ^ (1/3) + 0.05;

% tipo de estrutura
obj.posicao_Caract_Estrutura(:, :, 2) = arquivo_Matriz_Posicao_resistencia_tipo_estrutura;
% manutenção
obj.posicao_Caract_Estrutura(:, :, 3) = arquivo_Matriz_Posicao_resistencia_manutencao;
% altura
obj.posicao_Caract_Estrutura(:, :, 4) = arquivo_Matriz_Posicao_resistencia_altura;

```

```

end % end function

%Método para a substituição dos valores das classes pelos dados
%default para Fatores de Resistência
function obj = Input_Resistencia_default(obj)

    obj.factor_resist = zeros(obj.tam(1),obj.tam(2),4);

    obj.factor_resist(:,:,1) = obj.posicao_Caract_Estrutura(:,:,1);

    for i = 1:4
        % tipo de estrutura - fator de resistência
        obj.factor_resist(:,:,2) = obj.factor_resist(:,:,2) +
obj.default_Tipo_str(i)*(obj.posicao_Caract_Estrutura(:,:,2) == i);

        % manutenção da estrutura - fator de resistência
        obj.factor_resist(:,:,3) = obj.factor_resist(:,:,3) +
obj.default_State_Maint(i)*(obj.posicao_Caract_Estrutura(:,:,3) == i);

        % manutenção da estrutura - fator de resistência
        obj.factor_resist(:,:,4) = obj.factor_resist(:,:,4) +
obj.default_Altura(i)*(obj.posicao_Caract_Estrutura(:,:,4) == i);

    end % end for

    % manutenção da estrutura - fator de resistência
    obj.factor_resist(:,:,3) = obj.factor_resist(:,:,3) +
obj.default_State_Maint(5)*(obj.posicao_Caract_Estrutura(:,:,3) == 5);

    for i = 5:6
        % tipo de estrutura - fator de resistência
        obj.factor_resist(:,:,2) = obj.factor_resist(:,:,2) +
obj.default_Tipo_str(i)*(obj.posicao_Caract_Estrutura(:,:,2) == i);

    end %end for

end %end function

%Método para a substituição dos valores das classes pelos dados
%default para Limites de deslocamento da fundação
function obj = Input_Dst_default(obj)

    %inicializo matriz de matrix_Dst com zeros e
    %com as mesmas dimensões da área de estudo
    obj.matrix_Dst = zeros(obj.tam(1),obj.tam(2));

    for i = 1:22
        obj.matrix_Dst = obj.matrix_Dst + obj.default_Dst(i,1)*(obj.posicao_Dst == i);
    end % end for

end %end function

% Função que organiza os dados dos fatores de resistência fornecidos pelo usuário
function obj =
Input_Resistencia_Usuario(obj,arquivo_Fatores_tipo_estrutura,arquivo_Fatores_estado_manutencao,arquivo_Fatores_Altura)

    %inicializa a função organizada dos fatores de resistência
    obj.factor_resist = zero(obj.tam(1),obj.tam(2),4);

    obj.factor_Tipo_estrutura = arquivo_Fatores_tipo_estrutura;
    obj.factor_Estado_manutencao = arquivo_Fatores_estado_manutencao;
    obj.factor_Altura = arquivo_Fatores_Altura;

    obj.factor_resist(:,:,1) = obj.posicao_Caract_Estrutura(:,:,1);

    for i = 1:size(obj.factor_Tipo_estrutura,2)

        obj.factor_resist(:,:,2) = obj.factor_resist(:,:,2) +
obj.factor_Tipo_estrutura(i)*(obj.posicao_Caract_Estrutura(:,:,2) == i);
        obj.factor_resist(:,:,3) = obj.factor_resist(:,:,3) +
obj.factor_Estado_manutencao(i)*(obj.posicao_Caract_Estrutura(:,:,3) == i);
        obj.factor_resist(:,:,4) = obj.factor_resist(:,:,4) +
obj.factor_Altura(i)*(obj.posicao_Caract_Estrutura(:,:,4) == i);
    end

```

```

        end % end for
    end %end function

    % Função que organiza os dados dos valores limites dos deslocamentos das fundações fornecidos pelo
usuário
    function obj = Input_Dst_Usuario(obj,arquivo_limite_Deslocamento_fundacao)

        obj.Matrix_Limite_Fundacao = arquivo_limite_Deslocamento_fundacao;

        %inicializo matriz de matrix_Dst com zeros e
        %com as mesmas dimensões da área de estudo
        obj.matrix_Dst = zeros(obj.tam(1),obj.tam(2));

        for i = 1:size(obj.Matrix_Limite_Fundacao,1)

            obj.matrix_Dst = obj.matrix_Dst + obj.Matrix_Limite_Fundacao(i)*(obj.posicao_Dst == i);

        end % end for

    end % end function

end %end methods

end %end class

classdef Vulnerability
    %VULNERABILITY Summary of this class goes here
    % OS DADOS UTILIZADOS NESTA CLASSE JÁ DEVEM ESTAR TOTALMENTE
    % ESPACIALIZADOS (DEVEM SER TRATADOS ANTERIORMENTE)

    properties
        %dimensões da área de estudo
        tam;
        %matriz de vulnerabilidade para as estruturas na área de estudo
        matrix_Vulner;
        %matriz de intensidade dos deslizamentos para estruturas
        matrix_Istr;
        %matriz de velocidade associada aos deslizamentos (em mm/s)
        veloc_desl;
        %matriz do fator de intensidade dinâmica
        matrix_Idyn;
        %matriz das zonas impactadas com as suas respectivas profundidades
        %de detritos
        matrix_zon_impact;
        %Fator profundidade de detritos
        matrix_Idpt;
        %Fator de deformação para estruturas
        matrix_Idfm;
        %matriz referente aos índices de deslocamento da fundação ou
        %estrutura
        matrix_Ds;
        %matriz com valores limites de Ds
        matrix_Dst;

        %DADOS PARA O CÁLCULO DA RESISTÊNCIA

        %matriz do fator de resistência para estruturas
        matrix_Rstr;
        Rstr_aux;
        %matriz multidimensional dos fatores de resistência relativos às características das estruturas
        %Ordem dos índices em relação às dimensões:1-profundidade da fundação;
        %2-tipo da estrutura; 3-estado de manutenção; 4-altura da construção
        matrix_fator_resis;

    end

    methods

        function obj =
Vulnerability(veloc_desl,matrix_zon_impact,matrix_fator_resis,matrix_Ds,matrix_Dst,tam)
            obj.veloc_desl = veloc_desl;
            obj.matrix_zon_impact = matrix_zon_impact;
            obj.matrix_fator_resis = matrix_fator_resis;
            obj.matrix_Ds = matrix_Ds;
            obj.matrix_Dst = matrix_Dst;
            obj.tam = tam;
            %Inicializa-se a matriz de vulnerabilidade com zeros

```

```

obj.matrix_Vulner = zeros(obj.tam(1),obj.tam(2));

obj.matrix_Idyn = zeros(obj.tam(1),obj.tam(2));
obj.matrix_Idpt = zeros(obj.tam(1),obj.tam(2));
obj.matrix_Idfm = zeros(obj.tam(1),obj.tam(2));
obj.matrix_Istr = zeros(obj.tam(1),obj.tam(2));
obj.matrix_Rstr = zeros(obj.tam(1),obj.tam(2));

end % end function

function obj = Calculo_Vulnerab(obj)

%varre toda matriz da área de estudo
for i = 1:obj.tam(1)
    for j = 1:obj.tam(2)

        % Verifica onde há estruturas(quando há valores dos fatores não nulos)
        if obj.matrix_fator_resis(i,j,1) ~= 0

            % Verifica se estrutura está em uma zona de impacto
            if obj.matrix_zon_impact(i,j) ~= 0

                % Cálculo do fator de intensidade dinâmica
                % Velocidade em mm/s
                if obj.veloc_desl(i,j) <= 5e-7
                    %0 valor da vulnerabilidade quando vel<5e-7
                    %é zero, pois Idyn = 0 gerando Istr = 0
                    obj.matrix_Vulner(i,j) = 0;

                elseif obj.veloc_desl(i,j) > 5e-7

                    obj.matrix_Idyn(i,j) = 1/36*(log10(obj.veloc_desl(i,j))+6.3)^2;

                    %Cálculo do fator de profundidade de detritos
                    %(FATOR DE INTENSIDADE GEOMÉTRICA)
                    obj.matrix_Idpt(i,j) = 0.1*obj.matrix_zon_impact(i,j);

                    %Cálculo do fator de intensidade
                    obj.matrix_Istr(i,j) = obj.matrix_Idyn(i,j)*obj.matrix_Idpt(i,j);

                    %Cálculo do fator de resistência
                    obj.Rstr_aux = 1;
                    for k = 1:4
                        obj.Rstr_aux = obj.Rstr_aux*obj.matrix_fator_resis(i,j,k);
                    end % end for
                    obj.matrix_Rstr(i,j) = obj.Rstr_aux^(1/4);

                    %Cálculo da Vulnerabilidade (Eq. 1 - Artigo Uzielli)

                    if obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) <= 0.5
                        obj.matrix_Vulner(i,j) =
2*(obj.matrix_Istr(i,j))^2/(obj.matrix_Rstr(i,j))^2;

                    elseif obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) > 0.5 &&
obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) <= 1
                        obj.matrix_Vulner(i,j) = 1- 2*(obj.matrix_Rstr(i,j)-
obj.matrix_Istr(i,j))^2/(obj.matrix_Rstr(i,j))^2;

                    elseif obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) > 1
                        obj.matrix_Vulner(i,j) = 1;

                    end % end if

                end % end if

            %Caso a estrutura não esteja sobre uma zona de impacto, verifica se ela está na
            área INTERIOR ao deslizamento
            %Neste caso, a velocidade será diferente de zero
            elseif obj.veloc_desl(i,j) ~= 0

                % Cálculo do fator de intensidade dinâmica
                % Velocidade em mm/s
                if obj.veloc_desl(i,j) > 5e-7 && obj.veloc_desl(i,j) <= 0.5
                    obj.matrix_Idyn(i,j) = 1/36*(log10(obj.veloc_desl(i,j))+6.3)^2;

                    %Cálculo do fator de deformação para
                    %estruturas (FATOR DE INTENSIDADE GEOMÉTRICA)
                    obj.matrix_Idfm(i,j) = obj.matrix_Ds(i,j)/obj.matrix_Dst(i,j);

```

```

%Cálculo do fator de intensidade
obj.matrix_Istr(i,j) = obj.matrix_Idyn(i,j)*obj.matrix_Idfm(i,j);

%Cálculo do fator de resistência
obj.Rstr_aux = 1;
for k = 1:4
    obj.Rstr_aux = obj.Rstr_aux*obj.matrix_fator_resis(i,j,k);
end % end for
obj.matrix_Rstr(i,j) = obj.Rstr_aux^(1/4);

%Cálculo da Vulnerabilidade
    if obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) <= 0.5
        obj.matrix_Vulner(i,j) =
2*(obj.matrix_Istr(i,j))^2/(obj.matrix_Rstr(i,j))^2;
    elseif obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) > 0.5 &&
obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) <= 1
        obj.matrix_Vulner(i,j) = 1- 2*(obj.matrix_Rstr(i,j)-
obj.matrix_Istr(i,j))^2/(obj.matrix_Rstr(i,j))^2;
    elseif obj.matrix_Istr(i,j)/obj.matrix_Rstr(i,j) > 1
        obj.matrix_Vulner(i,j) = 1;
    end % end if

    % Valor limite de velocidade acima do qual a vulnerabilidade é considerada como
1
    elseif obj.veloc_desl(i,j) > 0.5

        obj.matrix_Vulner(i,j) = 1;

    % Caso o valor da velocidade esteja abaixo do limite inferior, Idyn é nulo
    elseif obj.veloc_desl(i,j) <= 5e-7

        obj.matrix_Vulner(i,j) = 0;

    end % end if

end %end if

    end % end if
end %end for
end %end for

end % end function

end %end methods

end %end class

```

A.6 FUNÇÕES PARA USO DOS AUTÔMATOS CELULARES E CARACTERÍSTICAS DOS DESLIZAMENTOS

A.6.1 INTERFACES GRÁFICAS

```

function varargout = GUI_AUTOMATO(varargin)
% GUI_AUTOMATO M-file for GUI_AUTOMATO.fig
% GUI_AUTOMATO, by itself, creates a new GUI_AUTOMATO or raises the existing
% singleton*.
%
% H = GUI_AUTOMATO returns the handle to a new GUI_AUTOMATO or the handle to
% the existing singleton*.
%
% GUI_AUTOMATO('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI_AUTOMATO.M with the given input arguments.
%
% GUI_AUTOMATO('Property','Value',...) creates a new GUI_AUTOMATO or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before GUI_AUTOMATO_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to GUI_AUTOMATO_OpeningFcn via varargin.
%
%

```

```

%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_AUTOMATO

% Last Modified by GUIDE v2.5 07-Sep-2014 11:45:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_AUTOMATO_OpeningFcn, ...
                  'gui_OutputFcn',  @GUI_AUTOMATO_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_AUTOMATO is made visible.
function GUI_AUTOMATO_OpeningFcn(hObject, eventdata, handles, varargin)
global decl_lim;
global vel_fonte;
global tf;
global termo_serie;
global opc_curvat;
global opc_freatico;
global visual_result;
global ro;
global espes_min;
global iter_newton;
global opcao_cv;
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_AUTOMATO (see VARARGIN)

% Choose default command line output for GUI_AUTOMATO
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

decl_lim = 5;
vel_fonte = 0;
tf = 50;
termo_serie = 1;
opc_curvat = 1;
opc_freatico = 1;
set(handles.pushbutton20, 'enable', 'off');
visual_result = 0;
ro = 0.5;
espes_min = 0;
iter_newton = 100;
opcao_cv = 1;
% UIWAIT makes GUI_AUTOMATO wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_AUTOMATO_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton17.
function pushbutton17_Callback(hObject, eventdata, handles)
%BOTÃO SAIR
% hObject    handle to pushbutton17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

delete(get(hObject, 'parent'));

% --- Executes on button press in pushbutton18.
function pushbutton18_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global num_passos;
global dado_chuva;
global min;
global ro;
global pixel;
global cv_unico;
global cv_grade;
global tf;
global Kt_AUT;
global decl_AUT;
global H_AUT;
global phi_AUT;
global gama_nat_AUT;
global gama_sat_AUT;
global S_AUT;
global n_AUT;
global cell_i;
global mnt;
global curvat;
global opcao_ch;
global opcao_m;
global m_ini;
global espes_min;
global opcao_cv;
global decl_lim;
global vel_fonte;
global termo_serie;
global curvat_vet;
global opc_crvat;
global opc_freatico;
global freat_unico;
global iter_newton;
global visual_result;
global automato_visualiza;
global iter_visualiza;

global ctrol_automato;

%Variáveis que vão para o cálculo da vulnerabilidade
global veloc_automato;
global detrito_automato;

h = waitbar(0,'1','Name','Simulação dos Autômatos Celulares...',...
           'CreateCancelBtn',...
           'setappdata(gcf,'canceling',1)');
setappdata(h,'canceling',0)

    steps = 7;
    waitbar(1/steps,h,sprintf('%0.f%% ',1/steps*100))

% declaração do objeto
autom_obj = automatos_class(load(cell_i));

    waitbar(2/steps,h,sprintf('%0.f%% ',2/steps*100))

% dependendo da opção se o freático é global ou se deve usar o modelo
    if opc_freatico == 1

        autom_obj = InputChuva(autom_obj,num_passos,dado_chuva);

        % Opção para unidade da chuva
        if opcao_ch ~= 3

            autom_obj = InputFS(autom_obj,opcao_ch);

        else

            autom_obj = InputFS_MIN(autom_obj,opcao_ch,min);

        end % end if

        % opção do coeficiente de adensamento
        if opcao_cv == 1

            cv = cv_unico;

```

```

else
    cv = load(cv_grade);
end % end if

% Valor inicial para m
if opcao_m == 2
    m_inicial = m_ini;
elseif opcao_m ==1
    m_inicial = 0;
end %end if

autom_obj =
InputDADOS(autom_obj,load(Kt_AUT),load(gama_nat_AUT),load(gama_sat_AUT),load(decl_AUT),load(H_AUT),load(phi
_AUT),load(n_AUT),load(S_AUT),m_inicial,load(mnt),tf,load(curvat),pixel,ro,cv,espes_min);

autom_obj = Calculo_s(autom_obj);

else
    autom_obj = Tamanho_freatico(autom_obj);

    autom_obj = Importa_s(autom_obj,freat_unico);

    % opção do coeficiente de adensamento
if opcao_cv == 1
    cv = cv_unico;
else
    cv = load(cv_grade);
end % end if

%
%   % Valor inicial para m
%   if opcao_m == 2
%   %
%   %       m_inicial = m_ini;
%   %
%   %   elseif opcao_m ==1
%   %   %
%   %       m_inicial = 0;
%   %
%   %   end %end if
%

    autom_obj =
InputDADOS_UNICO(autom_obj,load(gama_nat_AUT),load(gama_sat_AUT),load(decl_AUT),load(H_AUT),load(phi_AUT),l
oad(mnt),tf,load(curvat),pixel,ro,cv,espes_min);

end % end if

waitbar(3/steps,h,sprintf('%0.f%% ',3/steps*100))

if visual_result == 1
    autom_obj = Visualiza(autom_obj);
end % end if

waitbar(4/steps,h,sprintf('%0.f%% ',4/steps*100))

```

```

% escolha da opção de curvatura que será usada
% opção padrão
if opc_curvat == 1

    % valores default
    % Valores de curvatura CÔNCAVA
    cond_curvat = [1 4 7];

else

    cond_curvat = load(curvat_vet);

end % end if

waitbar(5/steps,h,sprintf('%0.f%% ',5/steps*100))

autom_obj =
Automato(autom_obj,opcao_cv,decl_lim,vel_fonte,termo_serie,cond_curvat,iter_newton,visual_result);

waitbar(6/steps,h,sprintf('%0.f%% ',6/steps*100))

% libera os botões para visualização do resultado
if visual_result == 1

    automato_visualiza = autom_obj.visualiza;
    iter_visualiza = autom_obj.iter_visualiza;

    set(handles.pushbutton22, 'enable', 'on');

end %end if

waitbar(7/steps,h,sprintf('%0.f%% ',7/steps*100))

delete(h)

% última configuração que vai para o cálculo da vulnerabilidade
veloc_automato = autom_obj.vel;
detrito_automato = autom_obj.detrito;

ctrol_automato = 1;

function edit10_Callback(hObject, eventdata, handles)
global num_passos;
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double
num_passos = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
global dado_chuva;
% hObject    handle to pushbutton20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

folder_name = uigetdir;

% CÓDIGO PARA ABRIR SOMENTE O ARQUIVO
% [filename pathname] = uigetfile({'*.*'},'File Selector');
% fullpathname = strcat(pathname,filename);
% set(handles.text36,'String',fullpathname);
%
% dado_chuva = fullfile(pathname,filename);

fullpathname = strcat(folder_name);

```

```

set(handles.text71,'String',fullpathname);

dado_chuva = fullpathname;

function edit11_Callback(hObject, eventdata, handles)
global min;
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double
min = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
global ro;
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double
ro = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
global pixel;
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text

```

```

%         str2double(get(hObject,'String')) returns contents of edit6 as a double
pixel = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
global cv_unico;
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6 as a double
cv_unico = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton19.
function pushbutton19_Callback(hObject, eventdata, handles)
global cv_grade;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text26, 'String', fullpathname);

cv_grade = fullfile(pathname, filename);

function edit12_Callback(hObject, eventdata, handles)
global tf;
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6 as a double
tf = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global Kt_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

%set(handles.text100,'String','AUTOMATO REALIZADO');

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text24,'String',fullpathname);

Kt_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global decl_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text25,'String',fullpathname);

decl_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
global H_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text26,'String',fullpathname);

H_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
global phi_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text27,'String',fullpathname);

phi_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
global gama_nat_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text28,'String',fullpathname);

gama_nat_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
global gama_sat_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text29,'String',fullpathname);

gama_sat_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
global S_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text30,'String',fullpathname);

```

```

S_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
global n_AUT;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text31, 'String', fullpathname);

n_AUT = fullfile(pathname,filename);

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
global cell_i;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text32, 'String', fullpathname);

cell_i = fullfile(pathname,filename);

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
global mnt;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text33, 'String', fullpathname);

mnt = fullfile(pathname,filename);

% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
global curvat;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'}, 'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text34, 'String', fullpathname);

curvat = fullfile(pathname,filename);

% --- Executes during object creation, after setting all properties.
function uipanel9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uipanel9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes when selected object is changed in uipanel9.
function uipanel9_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_ch;
% hObject    handle to the selected object in uipanel10
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

% opção das chuvas em mm/h
if hObject == handles.radiobutton10
    opcao_ch = 1;
    set(handles.edit11, 'enable', 'off');
% opção das chuvas em mm/dia
elseif hObject == handles.radiobutton11
    opcao_ch = 2;
    set(handles.edit11, 'enable', 'off');

```

```

% opção das chuvas em mm por X min
elseif hObject == handles.radiobutton12
    opcao_ch = 3;
    set(handles.edit11, 'enable', 'on');

end %end if

% --- Executes during object creation, after setting all properties.
function uipanel7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uipanel7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes when selected object is changed in uipanel11.
function uipanel11_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_m;
% hObject    handle to the selected object in uipanel11
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

% opção do valor de m inicial
if hObject == handles.radiobutton15
    opcao_m = 1;
    set(handles.edit13, 'enable', 'off');

elseif hObject == handles.radiobutton16
    opcao_m = 2;
    set(handles.edit13, 'enable', 'on');

end %end if

function edit13_Callback(hObject, eventdata, handles)
global m_ini;
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a double

m_ini = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function uipanel10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uipanel10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes when selected object is changed in uipanel10.
function uipanel10_SelectionChangeFcn(hObject, eventdata, handles)
global opcao_cv;
% hObject    handle to the selected object in uipanel10
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

% opção
if hObject == handles.radiobutton14

```

```

        opcao_cv = 1;
        set(handles.edit8, 'enable', 'on');
        set(handles.pushbutton19, 'enable', 'off');
    % opção
elseif hObject == handles.radiobutton13
    opcao_cv = 2;
    set(handles.edit8, 'enable', 'off');
    set(handles.pushbutton19, 'enable', 'on');

end %end if

function edit14_Callback(hObject, eventdata, handles)
global espes_min;
% hObject    handle to edit9 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of edit9 as a double

    espes_min = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)
global decl_lim;
% hObject    handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%         str2double(get(hObject,'String')) returns contents of edit15 as a double
decl_lim = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)
global vel_fonte;
% hObject    handle to edit16 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%         str2double(get(hObject,'String')) returns contents of edit16 as a double
vel_fonte = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit17_Callback(hObject, eventdata, handles)
global termo_serie;
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%         str2double(get(hObject,'String')) returns contents of edit17 as a double
termo_serie = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton20.
function pushbutton20_Callback(hObject, eventdata, handles)
global curvat_vet;
% hObject    handle to pushbutton20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[filename pathname] = uigetfile({'*.*'},'File Selector');
fullpathname = strcat(pathname,filename);
set(handles.text85,'String',fullpathname);

curvat_vet = fullfile(pathname,filename);

% --- Executes when selected object is changed in uipanel12.
function uipanel12_SelectionChangeFcn(hObject, eventdata, handles)
global opc_curvat;
% hObject    handle to the selected object in uipanel10
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

% opção
if hObject == handles.radiobutton17
    opc_curvat = 1;
    set(handles.pushbutton20, 'enable', 'off');
% opção
elseif hObject == handles.radiobutton18
    opc_curvat = 2;
    set(handles.pushbutton20, 'enable', 'on');

end %end if

% --- Executes when selected object is changed in uipanel14.
function uipanel14_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uipanel14
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

function edit19_Callback(hObject, eventdata, handles)
global freat_unico;
% hObject    handle to edit17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit17 as text
%         str2double(get(hObject,'String')) returns contents of edit17 as a double
freat_unico = str2double(get(hObject,'String'));

```

```

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes when selected object is changed in uipanel15.
function uipanel15_SelectionChangeFcn(hObject, eventdata, handles)
global opc_freatico;
% hObject    handle to the selected object in uipanel10
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

% opção
if hObject == handles.radiobutton23
    opc_freatico = 1;

    set(handles.edit19, 'enable', 'off');
    set(handles.pushbutton1, 'enable', 'on');

    set(handles.pushbutton5, 'enable', 'on');
    set(handles.pushbutton6, 'enable', 'on');
    set(handles.pushbutton7, 'enable', 'on');
    set(handles.pushbutton8, 'enable', 'on');
    set(handles.edit10, 'enable', 'on');
    set(handles.edit11, 'enable', 'off');
    set(handles.radiobutton10, 'enable', 'on');
    set(handles.radiobutton11, 'enable', 'on');
    set(handles.radiobutton12, 'enable', 'on');
    set(handles.pushbutton16, 'enable', 'on');

    set(handles.radiobutton15, 'enable', 'on');
    set(handles.radiobutton16, 'enable', 'on');

% opção FREÁTICO ÚNICO
elseif hObject == handles.radiobutton24
    opc_freatico = 2;

    set(handles.edit19, 'enable', 'on');
    set(handles.pushbutton1, 'enable', 'off');

    set(handles.pushbutton5, 'enable', 'on');
    set(handles.pushbutton6, 'enable', 'on');
    set(handles.pushbutton7, 'enable', 'off');
    set(handles.pushbutton8, 'enable', 'off');

    set(handles.edit10, 'enable', 'off');
    set(handles.edit11, 'enable', 'off');
    set(handles.radiobutton10, 'enable', 'off');
    set(handles.radiobutton11, 'enable', 'off');
    set(handles.radiobutton12, 'enable', 'off');
    set(handles.pushbutton16, 'enable', 'off');
    set(handles.text71, 'String', '');

    set(handles.radiobutton15, 'enable', 'off');
    set(handles.radiobutton16, 'enable', 'off');
    set(handles.edit13, 'enable', 'off');

    set(handles.text24, 'string', '');
    set(handles.text25, 'string', '');
    set(handles.text26, 'string', '');
    set(handles.text27, 'string', '');
    set(handles.text28, 'string', '');
    set(handles.text29, 'string', '');
    set(handles.text30, 'string', '');
    set(handles.text31, 'string', '');

end %end if

```

```

function edit20_Callback(hObject, eventdata, handles)
global iter_newton;
% hObject handle to edit20 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as text
% str2double(get(hObject,'String')) returns contents of edit20 as a double
iter_newton = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit20 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes when selected object is changed in uipanel18.
function uipanel18_SelectionChangeFcn(hObject, eventdata, handles)
global visual_result;
% hObject handle to the selected object in uipanel18
% eventdata structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles structure with handles and user data (see GUIDATA)

if hObject == handles.radiobutton28

    visual_result = 1;

elseif hObject == handles.radiobutton27

    visual_result = 0;

end % endo if

% --- Executes on button press in pushbutton22.
function pushbutton22_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton22 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
RESULT_AUTOMATO;

function varargout = RESULT_AUTOMATO(varargin)
% RESULT_AUTOMATO M-file for RESULT_AUTOMATO.fig
% RESULT_AUTOMATO, by itself, creates a new RESULT_AUTOMATO or raises the existing
% singleton*.
%
% H = RESULT_AUTOMATO returns the handle to a new RESULT_AUTOMATO or the handle to
% the existing singleton*.
%
% RESULT_AUTOMATO('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in RESULT_AUTOMATO.M with the given input arguments.
%
% RESULT_AUTOMATO('Property','Value',...) creates a new RESULT_AUTOMATO or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before RESULT_AUTOMATO_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to RESULT_AUTOMATO_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help RESULT_AUTOMATO

% Last Modified by GUIDE v2.5 25-Sep-2014 21:33:48

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @RESULT_AUTOMATO_OpeningFcn, ...
    'gui_OutputFcn', @RESULT_AUTOMATO_OutputFcn, ...

```

```

                'gui_LayoutFcn', [], ...
                'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before RESULT_AUTOMATO is made visible.
function RESULT_AUTOMATO_OpeningFcn(hObject, eventdata, handles, varargin)
global iter_visualiza;
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to RESULT_AUTOMATO (see VARARGIN)

% Choose default command line output for RESULT_AUTOMATO
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

set(handles.text6, 'String', strcat('De 1 a ', num2str(size(iter_visualiza,2))));

% UIWAIT makes RESULT_AUTOMATO wait for user response (see UIRESUME)
% uiwait(handles.figure1);
axis off

% --- Outputs from this function are returned to the command line.
function varargout = RESULT_AUTOMATO_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
global ver;
global iter_visualiza;
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit1 as text
%        str2double(get(hObject, 'String')) returns contents of edit1 as a double
ver = str2double(get(hObject, 'String'));

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global automato_visualiza;
global iter_visualiza;
global ver;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    if ver > size(iter_visualiza,2) || ver < 1

```

```

        warndlg('Entre com um número pertencente ao intervalo','!! Valor Inválido !!');
        ver = str2double(get(hObject,'String'));

    else

        axes(handles.axes1)
        set(handles.text8,'String',num2str(iter_visualiza(ver)));
        res = automato_visualiza(:, :, ver);
        imagesc(res);
        colormap('Summer')
        axis off

    end % end if

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global automato_visualiza;
global iter_visualiza;
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

p = 1;

while p < 5 %loop indefinitely. Use ctrl-c to break
    for i=1:size(automato_visualiza,3)
        imagesc(automato_visualiza(:, :, i))
        colormap('Summer')
        axis off
        drawnow
        pause(0.2) %display at about 20 FPS

        set(handles.text8,'String',num2str(iter_visualiza(i)));
    end
    p = p + 1;
end

% --- Executes during object creation, after setting all properties.
function text6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% BOTÃO SAIR
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(get(hObject, 'parent'));

```

A.6.2 CLASSE ESPECÍFICA PARA EXECUÇÃO DO MODELO

```

classdef automatoss_class
    %UNTITLED2 Summary of this class goes here
    % Detailed explanation goes here

    properties

        num_chuva;
        string_nome;
        string_complem;
        numero_arquivo;
        nome_arquivo;
        matriz_chuva;
        tamanho_area;
        unidade_chuva;
        delta_t;
        tam;
        Kt;
        decl;
        espessura;
        It;
        phi;
        n;
        Sr;
        m_inicial;
        s;
        Ot;
        delta_s;
    end
end

```

```

s_maior_1;
s_menor_1;
cell_i;
mnt;
curvat;
pixel;
ro;
cv;
valor_cv;
tf;
detrito;
sz;
cell;
vel;
vel_med;
vel_cont;
viz;
toSlide;
controle;
contr_vel;
d;
ub0;
ut0;
gama;
gama_sat;
n_termo;
x;
v;
residuo;
x_ant;
cellsN;
espress_minima;
a1;
a2;
ta;
tb;
x_a;
x_b;
soma_a;
soma_b;
a3;
a4;
t;
soma_x;
soma_d;
x_d;
t1;
n_int;
visualiza;
cont_visualiza;
iter_visualiza;

end % end properties

properties(Constant = true)

    % aceleração da gravidade
    g = 9.8;
    %célula com escorregamento
    escor = 1;
    %escorregamento propagado
    esc_f = 2;
    %deposição do escorregamento
    depositic = 3;

end % end properties

methods

    % inicializa o objeto
    function obj = automatoss_class(cell_i)

        obj.cell_i = cell_i;

    end % end function

    % função para tamanho da área de estudo para opção de nível
    % freático único
    function obj = Tamanho_freatico(obj)

        obj.tamanho_area = size(obj.cell_i);

    end % end function

```

```

%Método para importação dos dados de chuva
function obj = InputChuva(obj,num_chuva,dado_chuva)

    obj.num_chuva = num_chuva;

    obj.string_nome = 'input_chuva';
    obj.string_complem = '.txt';

    for i =1:obj.num_chuva
        obj.numero_arquivo = mat2str(i);
        obj.nome_arquivo =
strcat(dado_chuva,'\ ',obj.string_nome,obj.numero_arquivo,obj.string_complem);
        obj.matrix_chuva(:, :,i) = load(obj.nome_arquivo);
    end %end for

    obj.tamanho_area = size(obj.matrix_chuva);

    if obj.num_chuva == 1
        obj.tamanho_area(1,3) = 1;
    end %end if

end % end function

%Dados para o método que utiliza o cálculo do Fator de Segurança Chuva
%em horas ou dias
function obj = InputFS(obj,unidade_chuva)

    for i = 1:obj.num_chuva
        obj.It(:, :,i) = obj.matrix_chuva(:, :,i)/1000;
    end %end for

    obj.unidade_chuva = unidade_chuva;

    switch unidade_chuva

        %Intensidade em horas
        case 1
            obj.delta_t = 3600;

        %Intensidade em dias
        case 2
            obj.delta_t = 86400;

    end %end switch

end %end function

% Função para o caso que se está utilizando uma precipitação
% genérica (por exemplo, W mm de chuva por X min)
function obj = InputFS_MIN(obj,unidade_chuva,tempo)

    for i = 1:obj.num_chuva
        obj.It(:, :,i) = obj.matrix_chuva(:, :,i)/1000;
    end %end for

    obj.unidade_chuva = unidade_chuva;

    %Intensidade em min
    obj.delta_t = (tempo/60)*3600;

end %end function

function obj =
InputDADOS(obj,matrix_Kt,gama,gama_sat,matrix_decl,matrix_H,atrimento,n,Sr,m_inicial,mnt,tf,curvat,pixel,ro,cv
,espess_minima)

    obj.Kt = matrix_Kt;
    obj.decl = pi*(matrix_decl)/180;
    obj.espessura = matrix_H;
    obj.phi = pi*(atrimento)/180;
    obj.n = n;
    obj.Sr = Sr/100;
    obj.m_inicial = m_inicial;

```

```

obj.gama = gama;
obj.gama_sat = gama_sat;
obj.mnt = mnt;
obj.curvat = curvat;
obj.pixel = pixel;
obj.ro = ro;
obj.valor_cv = cv;
obj.tf = tf;
obj.espress_minima = espress_minima;

end %end function

% FUNÇÃO PARA O CASO EM QUE SE UTILIZA A OPÇÃO DO NÍVEL
% FREÁTICO ÚNICO
function obj =
InputDADOS_UNICO(obj,gama,gama_sat,matrix_decl,matrix_H,atrito,mnt,tf,curvat,pixel,ro,cv,espress_minima)

obj.decl = pi*(matrix_decl)/180;
obj.espressura = matrix_H;
obj.phi = pi*(atrito)/180;
obj.mnt = mnt;
obj.curvat = curvat;
obj.pixel = pixel;
obj.ro = ro;
obj.valor_cv = cv;
obj.tf = tf;
obj.espress_minima = espress_minima;
obj.gama = gama;
obj.gama_sat = gama_sat;

end %end function

function obj = Calculo_s(obj)

% Valor inicial de m pode ser zero ou não
obj.s(:, :) = obj.m_inicial*ones(obj.tamanho_area(1),obj.tamanho_area(2));

for k = 1:obj.tamanho_area(3)

% Cálculo do valor médio de m (índice 1)
obj.Ot = obj.Kt.*sin(obj.decl).*obj.s.*obj.espressura.*cos(obj.decl).*obj.delta_t;
obj.delta_s = (obj.It(:, :, k)-obj.Ot)./(obj.n.*obj.espressura.*(1-obj.Sr));
obj.s = obj.s+obj.delta_s;

%RESTRICÃO DO VALOR DE m

obj.s_maior_1 = obj.s >= 1;
obj.s_menor_1 = obj.s < 1;
obj.s_ = obj.s_maior_1 + obj.s_menor_1.*obj.s;

end % end for

end % end function

% Importação direta da grade do nível freático
function obj = Importa_s(obj,freatico)

obj.s = freatico*ones(obj.tamanho_area(1),obj.tamanho_area(2));

end % end function

% Matriz que irá armazenar os resultados da evolução do
% autômato celular para visualização
function obj = Visualiza(obj)

%matriz com dimensão da área de estudo e 3dim igual a
%1/3 das iterações totais previstas
obj.visualiza = zeros(obj.tamanho_area(1),obj.tamanho_area(2),fix(obj.tf/3)+1);

% quantidade de iterações salvas
obj.cont_visualiza = 0;

%número da iteração salva
obj.iter_visualiza = zeros(1,size(obj.visualiza,3));

```

```

end % end function

function obj =
Automato(obj,opcao_cv,decl_lim,vel_fonte,termo_serie,vet_curv,iter_newton,contr_visual)

% Tamanho da área
obj.sz = size(obj.cell_i);

%inicialmente, o plano de detritos terá a mesma
%espessura do solo. Posteriormente, com a transferência
%de detritos, sua espessura irá variar
obj.detrito = obj.espessura;

% o plano da fonte dos escorregamentos é passado para obj.cell
obj.cell = obj.cell_i;

%VISUALIZAÇÃO DE RESULTADOS
if contr_visual == 1
    % recebe
    obj.visualiza(:, :, 1) = obj.cell;

    obj.cont_visualiza = 2;

    obj.iter_visualiza(1) = 0;
end % end if

%
% Matriz que vai armazenar as velocidades
obj.vel = vel_fonte*obj.cell;
%

% repete o procedimento dos autômatos tf iterações
for i = 1:obj.tf

    % Variáveis para cálculo da velocidade média

    % acumula a soma da velocidade das células
    % vizinhas com escorregamento
    obj.vel_med = 0;
    % número de escorregamentos vizinhos da célula avaliada
    obj.vel_cont = 0;

    obj.viz = zeros(obj.sz);
    % fiz ALTERAÇÃO AQUI double(obj.cell ~= obj.esc_f)
    obj.viz = double(obj.cell ~= obj.esc_f).*double(obj.cell ~=
obj.escor).*conv2(double(obj.cell==obj.escor),[1 1 1; 1 0 1; 1 1 1], 'same');
    obj.toSlide = zeros(obj.sz);

    % variável booleana que verifica se em uma dada iteração ainda são gerados
    vizinhos de um dado escorregamento
    % se não, não realiza as demais iterações do autômato e
    % finaliza o processo
    obj.controle = 0;

    % varre toda área de estudo a procura de vizinhos de escorregamento
    for k = 2:obj.sz(1)-1
        for j = 2:obj.sz(2)-1

            % variável de controle para o cálculo da velocidade
            % se o pixel avaliado é um escorregamento, calcula-se a sua
            velocidade (contr_vel == 1)

            obj.contr_vel = 0;

            if obj.viz(k,j) > 0

                obj.controle = 1;

                for a = -1:1
                    for b = -1:1

                        if a ~=0 || b ~= 0

                            if (obj.mnt(k,j)- obj.mnt(k+a,j+b)) < 0
&& obj.cell(k+a,j+b) == 1

```

```

> 0
obj.curvat(k,j) == 4 || obj.curvat(k,j) == 7

(decl_lim*pi/180)

obj.escor;

média decorrente das células adjacentes com escorregamentos
velocidade inicial para a célula atual
+ obj.vel(k+a,j+b);
obj.vel_cont + 1;

%
-----
DETRITOS
(em altura de detritos)
transferência ( )
(tan(obj.decl(k,j)))/( tan(obj.decl(k+a-1,j+b-1)) + tan(obj.decl(k+a-1,j+b)) + tan(obj.decl(k+a-1,j+b+1)) +
tan(obj.decl(k+a,j+b-1)) + tan(obj.decl(k+a,j+b+1)) + tan(obj.decl(k+a+1,j+b-1)) + tan(obj.decl(k+a+1,j+b))
+ tan(obj.decl(k+a+1,j+b+1)) );

obj.d =
(tan(obj.decl(k,j)))/( (obj.mnt(k+a,j+b)-obj.mnt(k+a-1,j+b-1) > 0)*tan(obj.decl(k+a-1,j+b-1)) +
(obj.mnt(k+a,j+b)-obj.mnt(k+a-1,j+b) > 0)*tan(obj.decl(k+a-1,j+b)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a-1,j+b+1)
> 0)*tan(obj.decl(k+a-1,j+b+1)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a,j+b-1) > 0)*tan(obj.decl(k+a,j+b-1)) +
(obj.mnt(k+a,j+b)-obj.mnt(k+a,j+b+1) > 0)*tan(obj.decl(k+a,j+b+1)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a+1,j+b-1)
> 0)*tan(obj.decl(k+a+1,j+b-1)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a+1,j+b) > 0)*tan(obj.decl(k+a+1,j+b)) +
(obj.mnt(k+a,j+b)-obj.mnt(k+a+1,j+b+1) > 0)*tan(obj.decl(k+a+1,j+b+1)) );

altura da
para a célula
obj.detruto(k,j) + obj.d*obj.detruto(k+a,j+b);
obj.d*obj.detruto(k+a,j+b);
%
-----
menor, há a deposição do escorregamento
obj.deposic;

%
-----
DOS DETRITOS
massa (em altura de detritos)
transferência ( )
(tan(obj.decl(k,j)))/( tan(obj.decl(k+a-1,j+b-1)) + tan(obj.decl(k+a-1,j+b)) + tan(obj.decl(k+a-1,j+b+1)) +
tan(obj.decl(k+a,j+b-1)) + tan(obj.decl(k+a,j+b+1)) + tan(obj.decl(k+a+1,j+b-1)) + tan(obj.decl(k+a+1,j+b))
+ tan(obj.decl(k+a+1,j+b+1)) );

de altura da

```

```

% CRITÉRIO DE CURVATURA (CÔNCAVA)
if sum(obj.curvat(k,j) == vet_curv)

```

```

%if obj.curvat(k,j) == 1 ||

```

```

% CRITÉRIO DE DECLIVIDADE
% declividade em GRAUS
if obj.decl(k,j) >

```

```

obj.toSlide(k,j) =

```

```

% Cálculo da velocidade

```

```

% que servem como

```

```

obj.vel_med = obj.vel_med

```

```

obj.vel_cont =

```

```

obj.contr_vel = 1;

```

```

% CÁLCULO DA ESPESSORA DOS

```

```

% Transferência de massa

```

```

% coeficiente de

```

```

% obj.d =

```

```

(tan(obj.decl(k,j)))/( tan(obj.decl(k+a-1,j+b-1)) + tan(obj.decl(k+a-1,j+b)) + tan(obj.decl(k+a-1,j+b+1)) +
tan(obj.decl(k+a,j+b-1)) + tan(obj.decl(k+a,j+b+1)) + tan(obj.decl(k+a+1,j+b-1)) + tan(obj.decl(k+a+1,j+b))
+ tan(obj.decl(k+a+1,j+b+1)) );

```

```

obj.d =

```

```

(tan(obj.decl(k,j)))/( (obj.mnt(k+a,j+b)-obj.mnt(k+a-1,j+b-1) > 0)*tan(obj.decl(k+a-1,j+b-1)) +
(obj.mnt(k+a,j+b)-obj.mnt(k+a-1,j+b) > 0)*tan(obj.decl(k+a-1,j+b)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a-1,j+b+1)
> 0)*tan(obj.decl(k+a-1,j+b+1)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a,j+b-1) > 0)*tan(obj.decl(k+a,j+b-1)) +
(obj.mnt(k+a,j+b)-obj.mnt(k+a,j+b+1) > 0)*tan(obj.decl(k+a,j+b+1)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a+1,j+b-1)
> 0)*tan(obj.decl(k+a+1,j+b-1)) + (obj.mnt(k+a,j+b)-obj.mnt(k+a+1,j+b) > 0)*tan(obj.decl(k+a+1,j+b)) +
(obj.mnt(k+a,j+b)-obj.mnt(k+a+1,j+b+1) > 0)*tan(obj.decl(k+a+1,j+b+1)) );

```

```

% tranfere a fração 'd' de

```

```

% célula de deslizamento

```

```

% que está sendo avaliada

```

```

obj.detruto(k,j) =

```

```

obj.detruto(k+a,j+b) = (1-

```

```

%

```

```

% se a inclinação é

```

```

else

```

```

obj.toSlide(k,j) =

```

```

% CÁLCULO DA ESPESSURA

```

```

% Transferência de

```

```

% coeficiente de

```

```

obj.d =

```

```

(tan(obj.decl(k,j)))/( tan(obj.decl(k+a-1,j+b-1)) + tan(obj.decl(k+a-1,j+b)) + tan(obj.decl(k+a-1,j+b+1)) +
tan(obj.decl(k+a,j+b-1)) + tan(obj.decl(k+a,j+b+1)) + tan(obj.decl(k+a+1,j+b-1)) + tan(obj.decl(k+a+1,j+b))
+ tan(obj.decl(k+a+1,j+b+1)) );

```

```

% tranfere a fração 'd'

```



```

(objj.a3* (-objj.a4*objj.tb + exp(objj.a4*objj.tb) -1) )/(objj.a4^2);

exp(objj.a4*objj.ta) - 1)/objj.a4;

a de [a,b]
(1/2)*objj.ta*(objj.a1*objj.ta+2*objj.vel_med)-objj.pixel;
b de [a,b]
(1/2)*objj.tb*(objj.a1*objj.tb+2*objj.vel_med)-objj.pixel;
intervalo [a,b]
objj.a1*objj.ta + objj.vel_med;

positivo, a derivada pode ser positiva (função
(decrecente)

0.95*objj.ta;

1:objj.n_int

= ( 4*sin(pi/2*(2*nn-1)) * ( pi*(2*nn-1)*objj.ut0 - 2*cos(pi*nn)*(objj.ub0-objj.ut0) ) )/( pi^2 * ((2*nn-
1)^2));

= - (pi^2 * objj.cv * (2*nn-1)^2)/(4*(objj.s(k,j)*objj.espeçsura(k,j))^2);

objj.soma_a = objj.soma_a + (objj.a3* (-objj.a4*objj.ta + exp(objj.a4*objj.ta) -1) )/(objj.a4^2);

for
objj.a2*objj.soma_a + (1/2)*objj.ta*(objj.a1*objj.ta+2*objj.vel_med)-objj.pixel;

0

1.05*objj.tb;

= 0;

1:objj.n_int

= ( 4*sin(pi/2*(2*nn-1)) * ( pi*(2*nn-1)*objj.ut0 - 2*cos(pi*nn)*(objj.ub0-objj.ut0) ) )/( pi^2 * ((2*nn-
1)^2));

objj.soma_b = objj.soma_b +

%calcula a derivada
objj.soma_d = objj.a3*(

end % end for
% valor da função x(t) no ponto
objj.x_a = objj.a2*objj.soma_a +
% valor da função x(t) no ponto
objj.x_b = objj.a2*objj.soma_b +
% valor da derivada no ponto a do
objj.x_d = objj.a2*objj.soma_d +

if objj.x_a*objj.x_b > 0

% se o valor inferior é
% crescente) ou negtiva

if objj.x_a > 0
% crescente
if objj.x_d > 0
while objj.x_a > 0
objj.ta =

objj.soma_a = 0;

for nn =

objj.a3
objj.a4

end % end

objj.x_a =

end % end while
objj.t = objj.ta;
% decrecente
elseif objj.x_d < 0
while objj.x_b >

objj.tb =

objj.soma_b

for nn =

objj.a3
1) ^2));

```

```

= - (pi^2 * obj.cv * (2*nn-1)^2)/(4*(obj.s(k,j)*obj.espesura(k,j))^2);
obj.soma_b = obj.soma_b + (obj.a3* (-obj.a4*obj.tb + exp(obj.a4*obj.tb) -1) )/(obj.a4^2);
for
obj.a2*obj.soma_b + (1/2)*obj.tb*(obj.a1*obj.tb+2*obj.vel_med)-obj.pixel;
end % end
obj.x_b =
end % end while
obj.t = obj.tb;
end % end if

função
% valores negativos da
elseif obj.x_a < 0
% crescente
if obj.x_d > 0
while obj.x_b < 0
obj.tb =
obj.soma_b = 0;
for nn =
obj.a3 = (
4*sin(pi/2*(2*nn-1)) * ( pi*(2*nn-1)*obj.ut0 - 2*cos(pi*nn)*(obj.ub0-obj.ut0) ) )/( pi^2 * ((2*nn-1)^2));
obj.a4 = -
obj.soma_b
= obj.soma_b + (obj.a3* (-obj.a4*obj.tb + exp(obj.a4*obj.tb) -1) )/(obj.a4^2);
end % end for
obj.x_b =
end % end while
obj.t = obj.tb;
% decrescente
elseif obj.x_d < 0
while obj.x_a > 0
obj.ta =
obj.soma_a = 0;
for nn =
obj.a3 = (
4*sin(pi/2*(2*nn-1)) * ( pi*(2*nn-1)*obj.ut0 - 2*cos(pi*nn)*(obj.ub0-obj.ut0) ) )/( pi^2 * ((2*nn-1)^2));
obj.a4 = -
obj.soma_a
= obj.soma_a + (obj.a3* (-obj.a4*obj.ta + exp(obj.a4*obj.ta) -1) )/(obj.a4^2);
end % end for
obj.x_a =
end % end while
obj.t = obj.ta;

```

```

end % end if
end % end if obj.x_a > 0
end % end if obj.x_a*obj.x_b > 0

aproximação da raiz da equação

(igual a velocidade)

4*sin(pi/2*(2*nn-1)) * ( pi*(2*nn-1)*obj.ut0 - 2*cos(pi*nn)*(obj.ub0-obj.ut0) ) / ( pi^2 * ((2*nn-1)^2));
* obj.cv * (2*nn-1)^2 / (4*(obj.s(k,j)*obj.espessura(k,j))^2);

obj.soma_x + (obj.a3*( -obj.a4*obj.t + exp(obj.a4*obj.t) - 1) ) / (obj.a4^2);

obj.a3*( exp(obj.a4*obj.t) - 1) / obj.a4;

%

obj.a2*obj.soma_x + (1/2)*obj.t*(obj.a1*obj.t+2*obj.vel_med)-obj.pixel;

obj.a2*obj.soma_d +obj.a1*obj.t + obj.vel_med;

obj.x/obj.x_d;

obj.t) < 0.01

n_int termos da sequência que forma a velocidade
1:obj.n_int
obj.a3*( exp(obj.a4*obj.t1) - 1) / obj.a4;

TRANSFORMADA PARA MM/S NESSE PONTO
ADEQUAR À METODOLOGIA DE VULNERABILIDADE
1000*(obj.a2*obj.soma_d +obj.a1*obj.t1 + obj.vel_med);

%
velocidade nula se o resultado calculado for negativo
obj.vel(k,j) < 0

obj.vel(k,j) = 0.1;

% iterações para
for r = 1:iter_newton
obj.soma_x = 0;
%somatório da derivada
obj.soma_d = 0;
for nn = 1:obj.n_int
obj.a3 = (
obj.a4 = - (pi^2
obj.soma_x =
obj.soma_d =

end % end for

obj.x =
obj.x_d =
obj.t1 = obj.t -

if abs(obj.t1-
% soma dos
for nn =
obj.soma_d =
end % end for
% VELOCIDADE É
% PARA SE
obj.vel(k,j) =

%considera a
if

end % end if
break
end % end if

obj.t = obj.t1;

end % end for r = iterações

```

```

end % end if obj.contr_vel == 1

end % end if obj.viz(k,j) > 0

end % end for
end % end for

obj.cellsN = obj.toSlide;

%marca os escorregamentos nesta iteração como escorregamentos propagados
obj.cell((obj.cell == 1)) = obj.esc_f;
% atualiza a grade com os novos escorregamentos
obj.cell((obj.cellsN == 1)) = obj.escor;
%cell = double(cellsN) + double(cell);
obj.cell((obj.cellsN == obj.depositic)) = obj.depositic;

% VISUALIZAÇÃO DE RESULTADOS
if contr_visual == 1
if obj.iter_visualiza(obj.cont_visualiza-1)+3 == i

obj.visualiza(:, :,obj.cont_visualiza) = obj.cell;

obj.iter_visualiza(obj.cont_visualiza) = i;

obj.cont_visualiza = obj.cont_visualiza + 1;

end % end if

end %end if

if obj.contróle == 0
break
end % end if

end % end for

obj.detríto = obj.detríto - obj.espessura;

% restringe a espessura de detritos ao valor estabelecido pelo usuário
obj.detríto = obj.detríto.*(obj.detríto > obj.espess_minima);

end % end function

end % end methods

end % end class

```

A.7 FUNÇÕES PARA INSERÇÃO DOS CUSTOS E CÁLCULO DO RISCO

A.7.1 INTERFACE GRÁFICA

```

function varargout = CUSTO_RISCO2(varargin)
% CUSTO_RISCO2 M-file for CUSTO_RISCO2.fig
% CUSTO_RISCO2, by itself, creates a new CUSTO_RISCO2 or raises the existing
% singleton*.
%
% H = CUSTO_RISCO2 returns the handle to a new CUSTO_RISCO2 or the handle to
% the existing singleton*.
%
% CUSTO_RISCO2('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in CUSTO_RISCO2.M with the given input arguments.
%
% CUSTO_RISCO2('Property','Value',...) creates a new CUSTO_RISCO2 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before CUSTO_RISCO2_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to CUSTO_RISCO2_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

```

```

% Edit the above text to modify the response to help CUSTO_RISCO2

% Last Modified by GUIDE v2.5 18-Nov-2014 09:48:58

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @CUSTO_RISCO2_OpeningFcn, ...
                  'gui_OutputFcn',  @CUSTO_RISCO2_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before CUSTO_RISCO2 is made visible.
function CUSTO_RISCO2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to CUSTO_RISCO2 (see VARARGIN)

% Choose default command line output for CUSTO_RISCO2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CUSTO_RISCO2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = CUSTO_RISCO2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global grad_custo;
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text2, 'String', fullpathname);

grad_custo = fullfile(pathname, filename);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
global class_custo;
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename pathname] = uigetfile({'*. *'}, 'File Selector');
fullpathname = strcat(pathname, filename);
set(handles.text4, 'String', fullpathname);

class_custo = fullfile(pathname, filename);

% BOTÃO EXECUTAR
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
global grad_custo;
global class_custo;
global opc_custo;
global CUSTO;

```

```

global ctr_custos;
global tam_pixel_custo;

% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% opção para as classes ou para a grade organizada
if opc_custo == 1
    set(handles.text5,'String','Carregando dados...');
    entrada_custo = InputCusto(load(grad_custo),load(class_custo));
    entrada_custo = Input_Custo_Consequencias(entrada_custo);

    % MULTIPLICA A GRADE DE CUSTO PELA ÁREA DE CADA PIXEL PARA FORNECER O
    % VALOR TOTAL POR CÉLULA
    CUSTO = (tam_pixel_custo^2)*entrada_custo.matrix_Custo;

    warndlg('Dados das consequências carregados com sucesso!','Operação realizada');
    % Controle para saber que os custos das consequências foram carregados pelo menos 1 vez
    ctr_custos = 1;
    set(handles.text5,'String','Grade de custos carregada com sucesso!!');

elseif opc_custo == 2
    set(handles.text5,'String','Carregando dados...');

    % MULTIPLICA A GRADE DE CUSTO PELA ÁREA DE CADA PIXEL PARA FORNECER O
    % VALOR TOTAL POR CÉLULA
    CUSTO = (tam_pixel_custo^2)*load(grad_custo);

    warndlg('Dados das consequências carregados com sucesso!','Operação realizada');
    % Controle para saber que os custos das consequências foram carregados pelo menos 1 vez
    ctr_custos = 1;
    set(handles.text5,'String','Grade de custos carregada com sucesso!!');

end % end if

% --- Executes during object creation, after setting all properties.
function uipanel2_CreateFcn(hObject, eventdata, handles)
% hObject handle to uipanel2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% --- Executes when selected object is changed in uipanel2.
function uipanel2_SelectionChangeFcn(hObject, eventdata, handles)
global opc_custo;
% hObject handle to the selected object in uipanel2
% eventdata structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if none was selected
%   NewValue: handle of the currently selected object
% handles structure with handles and user data (see GUIDATA)
if hObject == handles.radiobutton1

    opc_custo = 1;
    % ligados
    set(handles.pushbutton2,'enable','on');

elseif hObject == handles.radiobutton2

    opc_custo = 2;
    % ligados
    set(handles.pushbutton2,'enable','off');

end % end if

% BOTÃO VERIFICAR
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
global ctr_custos;
global ctr_vulnerability;
global ctrl_per;
global PR;

% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Controle do perigo
if ctrl_per == 1

```

```

set(handles.text10,'String','Análise de Perigo realizada');
set(handles.text10,'BackgroundColor','green');

else
set(handles.text10,'String','Etapa não realizada!');
set(handles.text10,'BackgroundColor','red','ForegroundColor','white');

% Campos relativos às opções de perigo disponíveis que podem ser usadas
% na análise de perigo
set(handles.text19,'String','Não há intervalo válido','BackgroundColor','red');
set(handles.edit1,'enable','off');

end % end if

% Controle dos custos
if ctr_custos == 1
set(handles.text14,'String','Custos adicionados');
set(handles.text14,'BackgroundColor','green');
else
set(handles.text14,'String','Etapa não realizada!');
set(handles.text14,'BackgroundColor','red','ForegroundColor','white');
end % end if

% Controle da vulnerabilidade
if ctr_vulnerability == 1
set(handles.text12,'String','Análise de Vulnerabilidade realizada');
set(handles.text12,'BackgroundColor','green');
else
set(handles.text12,'String','Etapa não realizada!');
set(handles.text12,'BackgroundColor','red','ForegroundColor','white');
end % end if

crt_result = ctrl_per*ctr_vulnerability*ctr_custos;

% Se crt_result é um, então todas as análises foram realizadas e o risco
% pode ser efetivado
if crt_result == 1

% Campos relativos às opções de perigo disponíveis que podem ser usadas
% na análise de perigo
set(handles.text19,'String',strcat('De 1 a ',num2str(size(PR,3))),'BackgroundColor','green');
set(handles.edit1,'enable','on');

set(handles.text16,'String','Cálculo do Risco pode ser efetivado!');
set(handles.text16,'BackgroundColor','green');
set(handles.pushbutton5,'enable','on');

else
set(handles.text16,'String','Risco ainda não pode ser calculado!');
set(handles.text16,'BackgroundColor','red','ForegroundColor','white');
set(handles.pushbutton5,'enable','off');

end % end if

% EXECUTA O CÁLCULO DO RISCO
function pushbutton5_Callback(hObject, eventdata, handles)
global PR;
global VULNER;
global CUSTO;
global RISCO;
global PR_risco;
global ctr_risco;
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
RISCO = PR(:, :, PR_risco).*VULNER.*CUSTO;

warndlg('Análise do RISCO estabelecido com sucesso!', 'Operação realizada');

ctr_risco = 1;

function edit1_Callback(hObject, eventdata, handles)
global PR_risco;
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a double
PR_risco = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.

```

```

function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(get(hObject, 'parent'));

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1
a1 = imread('imag_custos.jpg');
image(a1);
axis off

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
a2 = imread('imag_risco.jpg');
image(a2);
axis off

function edit2_Callback(hObject, eventdata, handles)
global tam_pixel_custo;
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
tam_pixel_custo = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

A.7.2 CLASSE ESPECÍFICA PARA A EXECUÇÃO DO MODELO

```

classdef InputCusto
    %INPUTCUSTO Summary of this class goes here
    % Detailed explanation goes here

    properties
        tam;
        % Matrizes referentes ao valor das consequencias trazidas pelos
        % deslizamentos
        Matrix_posicao_Custo;
        Matrix_Valores_Custo;
    end
end

```

```

        matrix_Custo;
    end

    methods

        function obj = InputCusto(posicao,classes)
            obj.Matrix_posicao_Custo = posicao;
            obj.Matrix_Valores_Custo = classes;
            obj.tam = size(obj.Matrix_posicao_Custo);
            obj.matrix_Custo = zeros(obj.tam(1),obj.tam(2));
        end %end function

        %Método para entrada de dados sobre os custos caso uma ruptura
        %ocorra
        function obj = Input_Custo_Consequencias(obj)

            for i =1:obj.tam(1)
                for j = 1:obj.tam(2)

                    if obj.Matrix_posicao_Custo(i,j) ~= 0

                        obj.matrix_Custo(i,j) = obj.Matrix_Valores_Custo(obj.Matrix_posicao_Custo(i,j),1);

                    end %end if

                end %end for
            end %end for

        end % end function

    end %end methods

end % end class

```

A.8 INTERFACE GRÁFICA PARA EXPORTAÇÃO DOS DADOS

```

function varargout = Exporta(varargin)
% EXPORTA M-file for Exporta.fig
%   EXPORTA, by itself, creates a new EXPORTA or raises the existing
%   singleton*.
%
%   H = EXPORTA returns the handle to a new EXPORTA or the handle to
%   the existing singleton*.
%
%   EXPORTA('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in EXPORTA.M with the given input arguments.
%
%   EXPORTA('Property','Value',...) creates a new EXPORTA or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Exporta_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Exporta_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Exporta

% Last Modified by GUIDE v2.5 18-May-2014 16:04:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Exporta_OpeningFcn, ...
                  'gui_OutputFcn',  @Exporta_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

```

```

end
% End initialization code - DO NOT EDIT

% --- Executes just before Exporta is made visible.
function Exporta_OpeningFcn(hObject, eventdata, handles, varargin)
global ver;
global PR;
global ctrol_per;
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Exporta (see VARARGIN)

% OPÇÃO DEFAULT PARA A VISUALIZAÇÃO DOS RESULTADOS
%VER RESULTADOS DE PERIGO
ver = 1;

if ctrol_per ~= 0
    set(handles.text9,'String',strcat('De 1 a ',num2str(size(PR,3))));
end % end if

% Choose default command line output for Exporta
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Exporta wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Exporta_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in abrir.
function abrir_Callback(hObject, eventdata, handles)
global res;
% hObject    handle to abrir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename_salva pathname_salva] = uiputfile;

fullpathname_salva = strcat(pathname_salva,filename_salva);
set(handles.text6,'String',fullpathname_salva);
save([fullpathname_salva],'res','-ASCII');

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% BOTÃO SAIR
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(get(hObject, 'parent'));

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clear strings

f = figure('menu','none','toolbar','none');
set(f,'name','Ajuda','numbertitle','off')

fid = fopen('exporta_help.txt');
ph = uipanel(f,'Units','normalized','position',[0.05 0.05 0.9 0.9],'title',...
'Visualização e Exportação de Dados');
lbh = uicontrol(ph,'style','listbox','Units','normalized','position',...
[0 0 1 1],'FontSize',9);

indic = 1;
while 1
    tline = fgetl(fid);
    if ~ischar(tline),
        break
    end
    strings(indic)=tline;

```

```

        indic = indic + 1;
    end
    fclose(fid);
    set(lbh,'string',strings);
    set(lbh,'Value',1);
    set(lbh,'Selected','on');

% --- Executes on selection change in listBox2.
function listBox2_Callback(hObject, eventdata, handles)
% hObject    handle to listBox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listBox2 contents as cell array
%         contents{get(hObject,'Value')} returns selected item from listBox2

% --- Executes during object creation, after setting all properties.
function listBox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listBox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listBox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
a2 = imread('imag_salva.jpg');
image(a2);
axis off

function edit2_Callback(hObject, eventdata, handles)
global passo_ver;
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a double
passo_ver = str2double(get(hObject,'String'));

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on key press with focus on perigo and none of its controls.
function perigo_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to perigo (see GCBO)
% eventdata  structure with the following fields (see UICONTROL)
%   Key: name of the key that was pressed, in lower case
%   Character: character interpretation of the key(s) that was pressed
%   Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
% handles    structure with handles and user data (see GUIDATA)

% --- Executes when selected object is changed in uipanel8.
function uipanel8_SelectionChangeFcn(hObject, eventdata, handles)
global ver;
global PR;
global ctrol_per;
global ctr_vulnerability;
global ctr_risco;
% hObject    handle to the selected object in uipanel8
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)

```

```

% OldValue: handle of the previously selected object or empty if none was selected
% NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)

if hObject == handles.perigo
    ver = 1;

    if ctrol_per == 1
        set(handles.text9,'String',strcat('De 1 a ',num2str(size(PR,3))));
        set(handles.edit2,'String','', 'enable','on');
        set(handles.pushbutton4,'enable','on');
        set(handles.abrir,'enable','on');

    else
        set(handles.pushbutton4,'enable','off');
        set(handles.abrir,'enable','off');

    end % end if

elseif hObject == handles.vulnerabilidade
    ver = 2;

    set(handles.text9,'String','');
    set(handles.edit2,'String','', 'enable','off');

    if ctr_vulnerability == 1

        set(handles.pushbutton4,'enable','on');
        set(handles.abrir,'enable','on');
    else

        set(handles.pushbutton4,'enable','off');
        set(handles.abrir,'enable','off');
    end % end if

elseif hObject == handles.risco
    ver = 3;

    set(handles.text9,'String','');
    set(handles.edit2,'String','', 'enable','off');

    if ctr_risco == 1

        set(handles.pushbutton4,'enable','on');

    else

        set(handles.pushbutton4,'enable','off');
        set(handles.abrir,'enable','off');

    end % end if

end % end if

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
global passo_ver;
global ver;

% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
global passo_ver;
global ver;
global PR;
global res;
global VULNER;
global RISCO;

% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)

axes(handles.axes1)

switch ver

    case 1

        if passo_ver > size(PR,3)
            %handles.res = PR(:, :, passo_ver);
            warndlg('Não existem dados para este passo de tempo!', 'Valor Inválido');

        else

            res = PR(:, :, passo_ver);
            imagesc(res);
            axis off

        end % end if

        % resultado da VULNERABILIDADE
    case 2

        res = VULNER;
        imagesc(res);
        axis off

        % resultado do RISCO
    case 3

        res = RISCO;
        imagesc(res);
        axis off

end % end switch

```

```

% --- Executes during object creation, after setting all properties.
function axes3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes3

a = imread('imag_salva.jpg');
image(a);
axis off

```

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

A.9 INTERFACES GRÁFICAS DOS MENUS PRINCIPAIS

A.9.1 MENU INICIAL

```

function varargout = MAIN(varargin)
% MAIN M-file for MAIN.fig
%     MAIN, by itself, creates a new MAIN or raises the existing
%     singleton*.
%
%     H = MAIN returns the handle to a new MAIN or the handle to
%     the existing singleton*.
%
%     MAIN('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in MAIN.M with the given input arguments.
%
%     MAIN('Property','Value',...) creates a new MAIN or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before MAIN_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to MAIN_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

```

```

% Edit the above text to modify the response to help MAIN

% Last Modified by GUIDE v2.5 16-Aug-2014 22:45:34
%filename

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @MAIN_OpeningFcn, ...
                  'gui_OutputFcn',  @MAIN_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before MAIN is made visible.
function MAIN_OpeningFcn(hObject, eventdata, handles, varargin)

global ctrl_per;
global ctr_vulnerability;
global ctr_custos;
global ctrl_automato;
global ctr_risco;
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MAIN (see VARARGIN)
ctrl_per = 0;
ctrl_automato = 0;
ctr_vulnerability = 0;
ctr_custos = 0;
ctr_risco = 0;

% Choose default command line output for MAIN
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

handles.axes2 = imread('main.jpg');
image(handles.axes2);
axis off

handles.axes2 = imread('icone_saird.jpg');
image(handles.axes2);
axis off

% UIWAIT makes MAIN wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = MAIN_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
MODEL_PERIGO_handle = MODEL_PERIGO;

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
vulnerabilidade_handle = vulnerabilidade;

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Exporta_handle = Exporta;

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
resp = questdlg('Deseja realmente sair do sistema?','SAÍDA DO SAIRD');

if strcmp(resp,'Yes')

    delete(get(hObject, 'parent'));
end % end if

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CUSTO_RISCO2_handle = CUSTO_RISCO2;

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
a1 = imread('main.jpg');
image(a1);
axis off

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
a2 = imread('icone_saIRD.jpg');
image(a2);
axis off

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clear strings

f = figure('menu','none','toolbar','none');
set(f,'name','Ajuda','numbertitle','off')

fid = fopen('main_help.txt');
ph = uipanel(f,'Units','normalized','position',[0.05 0.05 0.9 0.9],'title',...
'Tela Inicial');
lbh = uicontrol(ph,'style','listbox','Units','normalized','position',...
[0 0 1 1],'FontSize',9);

indic = 1;
while 1
    tline = fgetl(fid);
    if ~ischar(tline),
        break
    end
    strings{indic}=tline;
    indic = indic + 1;
end
fclose(fid);
set(lbh,'string',strings);
set(lbh,'Value',1);
set(lbh,'Selected','on');

```

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
clear strings

f = figure('menu','none','toolbar','none');
set(f,'name','SOBRE','numbertitle','off')

fid = fopen('sobre.txt');
ph = uipanel(f,'Units','normalized','position',[0.05 0.05 0.9 0.9],'title',...
'Sobre RiskLab');
lbh = uicontrol(ph,'style','listbox','Units','normalized','position',...
[0 0 1 1],'FontSize',9);

indic = 1;
while 1
tline = fgetl(fid);
if ~ischar(tline),
break
end
strings{indic}=tline;
indic = indic + 1;
end
fclose(fid);
set(lbh,'string',strings);
set(lbh,'Value',1);
set(lbh,'Selected','on');

```

A.9.2 MENU DE AMEAÇA

```

function varargout = MODEL_PERIGO(varargin)
% MODEL_PERIGO M-file for MODEL_PERIGO.fig
% MODEL_PERIGO, by itself, creates a new MODEL_PERIGO or raises the existing
% singleton*.
%
% H = MODEL_PERIGO returns the handle to a new MODEL_PERIGO or the handle to
% the existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before MODEL_PERIGO_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to MODEL_PERIGO_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MODEL_PERIGO

% Last Modified by GUIDE v2.5 12-May-2015 14:10:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @MODEL_PERIGO_OpeningFcn, ...
'gui_OutputFcn', @MODEL_PERIGO_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before MODEL_PERIGO is made visible.
function MODEL_PERIGO_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to MODEL_PERIGO (see VARARGIN)

```

```

% Choose default command line output for MODEL_PERIGO
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MODEL_PERIGO wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = MODEL_PERIGO_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
global ctrol_per;
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
if ctrol_per == 0
    SHIA_COMPL2_handle = SHIA_COMPL2;
else
    resp = questdlg('Quer realizar outra análise de perigo?', 'ANÁLISE JÁ REALIZADA');

    if strcmp(resp, 'Yes')
        SHIA_COMPL2_handle = SHIA_COMPL2;
    end % end if

end % end if
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% BOTÃO SAIR
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
delete(get(hObject, 'parent'));

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
global ctrol_per;
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
if ctrol_per == 0
    CURVAS_NIVEL_IMPORTA_handle = CURVAS_NIVEL_IMPORTA;
else
    resp = questdlg('Quer realizar outra análise de perigo?', 'ANÁLISE JÁ REALIZADA');

    if strcmp(resp, 'Yes')
        CURVAS_NIVEL_IMPORTA_handle = CURVAS_NIVEL_IMPORTA;
    end % end if

end % end if

% --- Executes during object creation, after setting all properties.
function icone_CreateFcn(hObject, eventdata, handles)
% hObject handle to icone (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate icone
a1 = imread('imag_perigo.jpg');
image(a1);
axis off

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
global ctrol_per;
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
if ctrol_per == 0

```

```

MODELO_SLIDE_handle = MODELO_SLIDE;

else
    resp = questdlg('Quer realizar outra análise de perigo?', 'ANÁLISE JÁ REALIZADA');

    if strcmp(resp, 'Yes')
        MODELO_SLIDE_handle = MODELO_SLIDE;
    end % end if

end % end if

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

clear strings

f = figure('menu','none','toolbar','none');
set(f, 'name', 'Ajuda', 'numbertitle', 'off')

fid = fopen('perigo_help.txt');
ph = uipanel(f, 'Units', 'normalized', 'position', [0.05 0.05 0.9 0.9], 'title', ...
    'Modelos para Análise de Perigo');
lbh = uicontrol(ph, 'style', 'listbox', 'Units', 'normalized', 'position', ...
    [0 0 1 1], 'FontSize', 9);

indic = 1;
while 1
    tline = fgetl(fid);
    if ~ischar(tline),
        break
    end
    strings{indic}=tline;
    indic = indic + 1;
end
fclose(fid);
set(lbh, 'string', strings);
set(lbh, 'Value', 1);
set(lbh, 'Selected', 'on');

% --- Executes during object creation, after setting all properties.
function axes_per_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
peg = imread('imag_perigo.jpg');
image(peg);
axis off

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
global ctrol_per;
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if ctrol_per == 0
    FLUXO_DETRITOS_handle = FLUXO_DETRITOS;

else
    resp = questdlg('Quer realizar outra análise de perigo?', 'ANÁLISE JÁ REALIZADA');

    if strcmp(resp, 'Yes')
        FLUXO_DETRITOS_handle = FLUXO_DETRITOS;
    end % end if

end % end if

% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
peg_explica = imread('Perigo.jpg');
image(peg_explica);
axis off

```