



Rafael Otávio Alves Abreu

**Algoritmos para Integração e Calibração de
Modelos Elastoplásticos com Múltiplas Superfícies
de Plastificação**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da PUC-Rio como requisito parcial para obtenção do grau de Mestre em Engenharia Civil.

Orientadora: Profa. Deane de Mesquita Roehl
Coorientador: Dr. Eleazar Cristian Mejía Sanchez

Rio de Janeiro
Março de 2019



Rafael Otávio Alves Abreu

**Algoritmos para Integração e Calibração de
Modelos Elastoplásticos com Múltiplas Superfícies
de Plastificação**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Engenharia Civil da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

Profa. Deane de Mesquita Roehl

Orientadora

Departamento de Engenharia Civil e Ambiental – PUC-Rio

Dr. Eleazar Cristian Mejía Sanchez

Coorientador

Instituto Tecgraf/PUC-Rio

Prof. Raul Rosas e Silva

Departamento de Engenharia Civil e Ambiental – PUC-Rio

Prof. Ivan Fabio Mota de Menezes

Departamento de Engenharia Mecânica – PUC-Rio

Profa. Maria Fernanda Figueiredo de Oliveira

Universidade do Estado do Rio de Janeiro – UERJ

Rio de Janeiro, 14 de março de 2019

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Rafael Otávio Alves Abreu

Graduou-se em Engenharia Civil pela Universidade Estadual do Maranhão (2016), onde atuou como pesquisador de iniciação científica na área de estruturas de concreto. Em 2017, ingressou no Mestrado em Engenharia Civil ofertado pelo Departamento de Engenharia Civil e Ambiental da Pontifícia Universidade Católica do Rio de Janeiro, na área de concentração Estruturas. No ano de 2018, passou a integrar o grupo de Modelagem e Simulação Multifísica do Instituto Tecgraf/PUC-Rio como pesquisador. Atualmente, desenvolve pesquisas nas áreas de Plasticidade Computacional e Análise Inversa.

Ficha Catalográfica

Abreu, Rafael Otávio Alves

Algoritmos para integração e calibração de modelos elastoplásticos com múltiplas superfícies de plastificação / Rafael Otávio Alves Abreu ; orientadora: Deane de Mesquita Roehl ; coorientador: Eleazar Cristian Mejía Sanchez. – 2019.

163 f. : il. color. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Civil e Ambiental, 2019.

Inclui bibliografia

1. Engenharia Civil e Ambiental – Teses. 2. Plasticidade computacional. 3. Múltiplas superfícies de plastificação. 4. Otimização. 5. Análise inversa. 6. Calibração de parâmetros. I. Roehl, Deane de Mesquita. II. Sanchez, Eleazar Cristian Mejía. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Civil e Ambiental. IV. Título.

CDD: 624

*Dedico este trabalho a todos que me apoiaram
nesta jornada.*

Agradecimentos

Em primeiro lugar, agradeço a Deus, por todas as bênçãos que me foram concedidas.

Aos meus pais Maria Rita e Julio Cesar, pela educação, atenção e suporte oferecidos para atingir os meus objetivos.

À minha orientadora Deane Roehl, por toda confiança, estímulo e conhecimentos transmitidos ao longo de toda a pesquisa.

Ao meu coorientador Cristian Mejía, por todos os conselhos, sugestões e apoio, fundamentais para o desenvolvimento deste trabalho.

À moça que tomou meu coração para si, Gilcyvania Castro, que me acompanhou nesta jornada, longe de casa. Esteve ao meu lado em todos os momentos difíceis, sobretudo quando acidentei minha mão, na reta final da conclusão deste volume. Foram tempos difíceis, mas que superamos juntos, com apoio mútuo.

A todos familiares e amigos que contribuíram de alguma forma nesta minha jornada.

A todos os meus colegas do Instituto Tecgraf/PUC-Rio, que me deram apoio, sobretudo após ao acidente que sofri prestes a entregar este volume. Um agradecimento especial à Fernanda Pereira, Lourdes Maria, Roberto Quevedo, Nilthson Noreña, Pedro Firme, Bruna Silveira, Renan Costa, Marcello Congro, Francisco Dias, Renan Marks, Renato Escobar, Julio Rueda e Kaio César.

Aos professores da Universidade Estadual do Maranhão que incentivaram meu desenvolvimento acadêmico, em especial aos professores Eduardo Aguiar, João Celso e Carlos César.

Aos meus colegas da PUC-Rio.

Aos professores e funcionários da PUC-Rio, por todas as contribuições prestadas.

Ao CNPq e à PUC-Rio, por todos os auxílios concedidos, que possibilitaram o desenvolvimento deste trabalho.

Ao Instituto Tecgraf/PUC-Rio, pela oportunidade de desenvolver pesquisas desafiadoras e de grande relevância.

Resumo

Abreu, Rafael Otávio Alves; Roehl, Deane de Mesquita (Orientadora); Sanchez, Eleazar Cristian Mejía (Coorientador). **Algoritmos para Integração e Calibração de Modelos Elastoplásticos com Múltiplas Superfícies de Plastificação**. Rio de Janeiro, 2019. 163p. Dissertação de Mestrado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

A representação do comportamento de materiais elastoplásticos a partir de modelos com múltiplas superfícies de plastificação é uma alternativa para representar o comportamento de materiais como concreto, rochas e solos, que apresentam diferentes tipos de resposta não linear, a depender do estado de tensão atuante. No entanto, o emprego desses modelos requer a definição de muitos parâmetros que, por vezes, não possuem significado físico. Além disso, a implementação de modelos elastoplásticos com múltiplas envoltórias traz complexidades adicionais. O emprego desse tipo de modelo requer um esquema robusto de integração das equações de evolução das variáveis plásticas. Nessa perspectiva, é apresentado um algoritmo de mapeamento de retorno baseado em um método de otimização sem restrições, o método de Newton-Raphson com busca unidimensional. Propõe-se uma expressão para o tensor constitutivo elastoplástico consistente para modelos com múltiplas superfícies de plastificação. Adicionalmente, apresenta-se uma metodologia de calibração dos parâmetros de tais modelos a partir da solução de um problema de otimização, solucionado via algoritmo genético. Para melhor compreender os parâmetros envolvidos nesse algoritmo, desenvolve-se um estudo paramétrico, solucionando uma série de problemas de otimização global. A robustez e eficácia dos algoritmos são avaliadas por meio de aplicações, dentre elas algumas disponíveis na literatura, num modelo constitutivo idealizado para concreto, rochas e solos: *Cap Model*. Por fim, calibra-se tal modelo, considerando dados experimentais disponíveis na literatura. Assim, este trabalho tem como objetivo contribuir para viabilizar o emprego de modelos elastoplásticos complexos em problemas de engenharia.

Palavras-chave

Plasticidade Computacional; Múltiplas Superfícies de Plastificação; Otimização; Análise Inversa; Calibração de Parâmetros

Abstract

Abreu, Rafael Otávio Alves; Roehl, Deane de Mesquita (Advisor); Sanchez, Eleazar Cristian Mejía (Co-advisor). **Algorithms for Integration and Calibration of Multisurface Elastoplastic Models**. Rio de Janeiro, 2019. 163p. Dissertação de Mestrado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

Elastoplastic models with multiple plastic surfaces is an alternative to represent the nonlinear behavior of materials such as concrete, rocks and soils. The nonlinear response of these materials depends highly on the stress state. However, these models require the definition of many parameters which do not always have physical meaning. In addition, the implementation of elastoplastic models represented by multiple plastic surfaces brings additional complexities to the analysis. The use of this type of model requires a robust numerical integration scheme of the elastoplastic evolution equations. This work presents two contributions. The first contribution is a robust return mapping algorithm for a multisurface plasticity model in general stress space known as closest point projection algorithm. The return mapping algorithm is based on a numerical method for unconstrained optimization. In this scenario, it is adopted the Newton-Raphson method with line search. A consistent tangent modulus for multisurface plasticity is also proposed. The second contribution is a methodology for parameter calibration. This methodology is formulated as an optimization problem, with the solution obtained through a genetic algorithm. A parametric study is developed in order to better understand specific parameters of the algorithm, solving global optimization problems. Robustness and effectiveness of the proposed algorithm are evaluated through numerical examples applied to a constitutive model used for modelling concrete, rocks and soils: Cap Model. Applications available in the literature are analysed. Lastly, the parameters of this model are calibrated using experimental data available in the literature. Thus, this work aims at improving the feasibility of the use of complex elastoplastic models in engineering problems.

Keywords

Computational Plasticity; Multisurface Plasticity; Optimization; Inverse Analysis; Parameter Calibration

Sumário

1	Introdução	23
1.1.	Motivação	25
1.2.	Objetivos	25
1.3.	Estrutura da Dissertação	26
2	Tópicos de Teoria da Plasticidade	28
2.1.	Problema Elastoplástico Generalizado	29
2.1.1.	Decomposição do Tensor de Deformações	29
2.1.2.	Relação Elástica entre Tensão e Deformação	30
2.1.3.	Critério de Plastificação	31
2.1.4.	Regra de Fluxo Plástico e Lei de Endurecimento/Amolecimento	32
2.1.5.	Multiplicador Plástico	35
2.1.6.	Tensor Constitutivo Elastoplástico	36
2.2.	Critério de Plastificação com Múltiplas Superfícies de Plastificação: <i>Cap Model</i>	37
3	Desenvolvimento de um Algoritmo de Mapeamento de Retorno para Múltiplas Superfícies de Plastificação	40
3.1.	Definições Básicas	42
3.1.1.	Formulação do Algoritmo de Projeção ao Ponto mais Próximo	44
3.1.2.	Tensor Constitutivo Elastoplástico Consistente	50
3.2.	Abordagens para Aperfeiçoamento do Algoritmo de Projeção ao Ponto mais Próximo	54
3.2.1.	Inversão de Matriz via Decomposição em Valores Singulares	55
3.2.2.	Método de Newton-Raphson como Algoritmo de Otimização	57
4	Análise Inversa	74
4.1.	Calibração de Parâmetros	76
4.2.	Solução de Problemas Inversos	76

4.3. Abordagem Direta para Solução de Problemas Inversos	77
4.3.1. Problema de Otimização	78
4.3.2. Algoritmos Determinísticos	81
4.3.3. Algoritmos Meta-heurísticos	82
5 Desenvolvimento de um Algoritmo Genético	84
5.1. Função de Aptidão	85
5.2. Operador de Seleção	86
5.3. Operador de Cruzamento	87
5.3.1. Cruzamento Aritmético	87
5.3.2. Cruzamento Binário Simulado	88
5.4. Operador de Mutação	88
5.4.1. Mutação Não Uniforme	89
5.4.2. Mutação Randômica Dinâmica	89
5.5. Descrição do Algoritmo Genético	90
5.6. Estudo Paramétrico	94
5.7. Conclusões Parciais	99
6 Aplicações e Resultados	101
6.1. Estudo do Algoritmo de Mapeamento de Retorno considerando um Modelo Elastoplástico Perfeitamente Plástico	102
6.2. Estudo do Algoritmo de Mapeamento de Retorno considerando um Modelo Elastoplástico com Endurecimento	109
6.3. Ensaio de Compressão Hidrostática	113
6.4. Ensaio Edométrico	115
6.5. Modelagem em Elementos Finitos de uma Sapata Flexível	118
6.6. Calibração de Parâmetros do <i>Cap Model</i> a partir de Dados Experimentais	119
7 Conclusões e Sugestões para Trabalhos Futuros	126
7.1. Conclusões	126
7.2. Sugestões para Trabalhos Futuros	127
8 Referências bibliográficas	130

Apêndice A Resultados do Estudo Paramétrico do Algoritmo Genético	139
Apêndice B Equações Auxiliares para Implementação do <i>Cap Model</i>	151
B.1. Derivadas dos Invariantes do Tensor de Tensões	151
B.2. Derivadas das Funções de Plastificação do <i>Cap Model</i>	152
Apêndice C Curvas Tensão-Deformação Calibradas para o <i>Cap Model</i>	157

Lista de Figuras

Figura 2.1 – Domínio elástico para um modelo elastoplástico com múltiplas superfícies de plastificação	32
Figura 2.2 – Representação do <i>Cap Model</i>	39
Figura 3.1 – Processo de atualização de estado de um problema elastoplástico	42
Figura 3.2 – Interpretação geométrica do algoritmo de projeção ao ponto mais próximo	50
Figura 3.3 – Interpretação geométrica da regra de Armijo	61
Figura 3.4 – Interpretação gráfica do método da seção áurea	63
Figura 3.5 – Dependência linear entre as direções dos fluxos plásticos	70
Figura 4.1 – Esquema de solução de problemas diretos e inversos	77
Figura 4.2 – Fluxograma da análise inversa	80
Figura 5.1 – Fluxograma de um algoritmo genético	85
Figura 6.1 – Sapata flexível	118

Lista de Tabelas

Tabela 5.1 – Lista de problemas de otimização com dimensão fixada	95
Tabela 5.2 – Lista de problemas de otimização com dimensão variável	95
Tabela 5.3 – Parâmetros do estudo paramétrico	96
Tabela 5.4 – Parâmetros adotados nas aplicações deste trabalho	99
Tabela 6.1 – Parâmetros adotados para o ensaio de compressão hidrostática	103
Tabela 6.2 – Deformações totais ϵ_{n+1} empregadas no estudo	103
Tabela 6.3 – Tensões atualizadas para as deformações totais estudadas	104
Tabela 6.4 – Parâmetros adotados para o ensaio de compressão hidrostática	109
Tabela 6.5 – Deformações totais ϵ_{n+1} empregadas no estudo	109
Tabela 6.6 – Tensões atualizadas para as deformações totais estudadas	110
Tabela 6.7 – Parâmetros adotados para o ensaio de compressão hidrostática	114
Tabela 6.8 – Tempos relativos para cada metodologia de busca unidimensional para o ensaio de compressão hidrostática	115
Tabela 6.9 – Parâmetros adotados para o ensaio de edométrico	116
Tabela 6.10 – Ensaio de Salami (1986) empregados na calibração	120
Tabela 6.11 – Parâmetros elásticos para o concreto ensaiado por Salami (1986)	121
Tabela 6.12 – Domínio dos parâmetros do <i>Cap Model</i> para o concreto ensaiado por Salami (1986)	121
Tabela 6.13 – Parâmetros obtidos após a calibração	122
Tabela 6.14 – Melhores parâmetros obtidos após a calibração	122
Tabela A.1 – Quantidade e taxas de sucessos para os problemas de otimização com dimensão fixada	145

Tabela A.2 – Médias das funções erro para os problemas de otimização com dimensão fixada	146
Tabela A.3 – Desvios padrões das funções erro para os problemas de otimização com dimensão fixada	147
Tabela A.4 – Quantidade e taxas de sucessos para os problemas de otimização com dimensão variável	148
Tabela A.5 – Médias das funções erro para os problemas de otimização com dimensão variável	149
Tabela A.6 – Desvios padrões das funções erro para os problemas de otimização com dimensão variável	150

Lista de Quadros

Quadro 3.1 – Cálculo da pseudoinversa da matriz A a partir da decomposição em valores singulares	56
Quadro 3.2 – Regra de Armijo	62
Quadro 3.3a – Busca unidimensional via método da seção áurea (parte 1)	63
Quadro 3.3b – Busca unidimensional via método da seção áurea (parte 2)	64
Quadro 3.4a – Busca unidimensional via interpolação quadrática ou cúbica (parte 1)	67
Quadro 3.4b – Busca unidimensional via interpolação quadrática ou cúbica (parte 2)	68
Quadro 3.4c – Busca unidimensional via interpolação quadrática ou cúbica (parte 3)	69
Quadro 3.5a – Algoritmo de projeção ao ponto mais próximo com busca unidimensional para múltiplas superfícies de plastificação (parte 1)	71
Quadro 3.5b – Algoritmo de projeção ao ponto mais próximo com busca unidimensional para múltiplas superfícies de plastificação (parte 2)	72
Quadro 3.5c – Algoritmo de projeção ao ponto mais próximo com busca unidimensional para múltiplas superfícies de plastificação (parte 3)	73
Quadro 5.1a – Algoritmo genético (parte 1)	92
Quadro 5.1b – Algoritmo genético (parte 2)	93

Lista de Gráficos

Gráfico 5.1 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão fixa	97
Gráfico 5.2 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão fixa	97
Gráfico 5.3 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão fixa	98
Gráfico 6.1 – Histórico de correções das tensões para PP-2 com o método de Newton-Raphson sem busca unidimensional	104
Gráfico 6.2 – Histórico de correções das tensões para PP-2 com o método de Newton-Raphson com busca unidimensional	105
Gráfico 6.3 – Ampliação na solução final do histórico das correções das tensões para PP-2 com o método de Newton-Raphson com busca unidimensional	105
Gráfico 6.4 – Histórico de correções das tensões para PP-3 com o método de Newton-Raphson sem busca unidimensional	106
Gráfico 6.5 – Histórico de correções das tensões para PP-3 com o método de Newton-Raphson com busca unidimensional	106
Gráfico 6.6 – Ampliação na solução final do histórico das correções das tensões PP-3 com o método de Newton-Raphson com busca unidimensional	107
Gráfico 6.7 – Histórico de correções das tensões para PP-4 com o algoritmo de projeção ao ponto mais próximo	108
Gráfico 6.8 – Ampliação na solução final do histórico das correções das tensões para PP-4 com o algoritmo de projeção ao ponto mais próximo	108
Gráfico 6.9 – Histórico de correções das tensões para EE-1 com o método de Newton-Raphson sem busca unidimensional	110

Gráfico 6.10 – Histórico de correções das tensões para EE-1 com o método de Newton-Raphson com busca unidimensional	111
Gráfico 6.11 – Histórico de correções das tensões para EE-2 com o método de Newton-Raphson sem busca unidimensional	111
Gráfico 6.12 – Histórico de correções das tensões para EE2 com o método de Newton-Raphson com busca unidimensional	112
Gráfico 6.13 – Histórico de correções das tensões para EE-3 com o método de Newton-Raphson sem busca unidimensional	112
Gráfico 6.14 – Histórico de correções das tensões para EE-3 com o método de Newton-Raphson com busca unidimensional	113
Gráfico 6.15 – Evolução do primeiro invariante do tensor de tensões em relação às deformações volumétrica para o ensaio numérico de compressão hidrostática	114
Gráfico 6.16 – Evolução da tensão axial em relação às deformações axiais para o ensaio numérico edométrico	116
Gráfico 6.17 – Trajetória de tensões para o ensaio numérico edométrico	117
Gráfico 6.18 – Curva pressão-deslocamento para a sapata flexível	119
Gráfico 6.19 – Evolução do erro ao longo das análises para as soluções obtidas para $p_m = 0,05$	123
Gráfico 6.20 – Evolução do erro ao longo das análises para as soluções obtidas para $p_m = 0,1$	123
Gráfico 6.21 – Evolução do erro ao longo das análises para as soluções obtidas para $p_m = 0,15$	124
Gráfico 6.22 – Evolução do erro ao longo das análises para as melhores soluções	125
Gráfico A.1 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 5)	139
Gráfico A.2 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 10)	139
Gráfico A.3 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 20)	140
Gráfico A.4 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 30)	140

Gráfico A.5 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 40)	140
Gráfico A.6 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 5)	141
Gráfico A.7 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 10)	141
Gráfico A.8 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 20)	141
Gráfico A.9 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 30)	142
Gráfico A.10 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 40)	142
Gráfico A.11 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 5)	142
Gráfico A.12 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 10)	143
Gráfico A.13 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 20)	143
Gráfico A.14 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 30)	143
Gráfico A.15 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 40)	144
Gráfico C.1 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional	157
Gráfico C.2 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional	158
Gráfico C.3 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional	158

Gráfico C.4 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional	159
Gráfico C.5 – Curvas Tensão-Deformação para o Ensaio de Extensão Triaxial Reduzida	159
Gráfico C.6 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial	160
Gráfico C.7 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial	160
Gráfico C.8 – Curvas Tensão-Deformação para o Ensaio de Cisalhamento Simples	161
Gráfico C.9 – Curvas Tensão-Deformação para o Ensaio de Extensão Triaxial	161
Gráfico C.10 – Curvas Tensão-Deformação para o Ensaio de Carregamento Proporcional	162
Gráfico C.11 – Curvas Tensão-Deformação para o Ensaio de Trajetória Circular de Tensões	162
Gráfico C.12 – Curvas Tensão-Deformação para o Ensaio de Trajetória Arbitrária de Tensões	163

Lista de Símbolos

$(\dot{\cdot})$	Derivada de (\cdot) com respeito ao tempo
$\frac{\partial(\cdot)}{\partial x}$	Derivada parcial de (\cdot) com respeito a x
$\nabla(\cdot)$	Vetor gradiente de (\cdot)
$\partial_x(\cdot)$	Derivada de (\cdot) com respeito a x
$\partial_{xy}^2(\cdot)$	Derivada de segunda ordem mista de (\cdot) com respeito a x e y
$(\cdot)_i$	Valores de índice i da variável (\cdot)
$\langle \cdot \rangle$	$\langle \cdot \rangle$ denota os colchetes de Macaulay
$tr(\cdot)$	Traço de (\cdot)
$\Delta(\cdot)$	Incremento da variável (\cdot)
$\nabla^s(\cdot)$	Gradiente simétrico de um vetor de campo (\cdot)
$(\cdot)^{trial}$	Variável (\cdot) preditor(a)
$(\cdot)_{n+1}$	Valor de (\cdot) para o incremento $n + 1$
$(\cdot)^{(k)}$	Valor de (\cdot) na iteração k
$\frac{d(\cdot)}{dx}$	Derivada de (\cdot) com respeito a x
$(\cdot)'$	Dado (\cdot) normalizado
$\boldsymbol{\varepsilon}$	Tensor de deformações
$\boldsymbol{\varepsilon}^e$	Tensor de deformações elásticas
$\boldsymbol{\varepsilon}^p$	Tensor de deformações plásticas
t_0	Pseudotempo inicial
$\boldsymbol{\sigma}$	Tensor de tensões
W	Energia potencial elástica
\boldsymbol{x}	Posição de um ponto num corpo; ponto genérico em uma hipersuperfície
t	Tempo; número da geração
\boldsymbol{D}	Tensor constitutivo elástico
\boldsymbol{C}	Inverso do tensor constitutivo elástico

\mathbf{q}	Tensor de variáveis de endurecimento/amolecimento
$f(\boldsymbol{\sigma}, \mathbf{q})$	Função de plastificação
\mathbb{E}	Domínio elástico
\mathbb{Y}	Superfície de plastificação
$\bar{\mathbb{E}}$	Conjunto das tensões plasticamente admissíveis
m	Número de superfícies de plastificação do modelo; número de conjuntos de dados para a calibração de parâmetros
γ	Multiplicador plástico
$\mathbf{n}(\boldsymbol{\sigma}, \mathbf{q})$	Função que define a direção do fluxo plástico
$\mathbf{h}(\boldsymbol{\sigma}, \mathbf{q})$	Função que define o tipo de endurecimento/amolecimento
\mathbb{J}_{act}	Conjunto de superfícies de plastificação ativas
\mathbf{D}^{ep}	Tensor constitutivo elastoplástico
\emptyset	Conjunto vazio
$I_1(\boldsymbol{\sigma})$	Primeiro invariante do tensor de tensões
$J_2(\boldsymbol{\sigma})$	Segundo invariante do tensor de tensões desviadoras
$\mathbf{s}(\boldsymbol{\sigma})$	Tensões desviadoras
ε_v^p	Deformações plásticas volumétricas
\mathbf{r}_1	Resíduo da regra de fluxo plástico
\mathbf{r}_2	Resíduo da lei de endurecimento/amolecimento
\mathbf{r}	Vetor de resíduos
\mathbf{I}	Tensor identidade
\mathcal{J}	Matriz Jacobiana
$\psi(\mathbf{x}_{n+1})$	Função objetivo para a solução do sistema de equações não lineares
\mathbf{d}	Direção de descida
\mathcal{H}	Matriz Hessiana
α	Tamanho do passo
tol	Tolerância
φ	Inverso da razão áurea
\mathbf{p}	Vetor de parâmetros
Ω	Domínio do problema
$E^t(\mathbf{p})$	Função erro total médio
w	Peso das funções erro

$E_i(\mathbf{p})$	Função erro
N	Número de dados observados; tamanho da população
\mathbf{W}	Matriz de pesos
$F(\mathbf{x})$	Função objetivo genérica
\mathbf{u}	Limite inferior do domínio Ω dos parâmetros \mathbf{p} no problema de otimização
\mathbf{l}	Limite superior do domínio Ω dos parâmetros \mathbf{p} no problema de otimização
D	Dimensão da função objetivo (número de genes por indivíduo)
\mathbf{x}^*	Solução do problema de otimização
$\mathcal{F}(\mathbf{x})$	Função de aptidão
x_i^a, x_i^b	Genes de indivíduos
$\bar{x}_i^a, \bar{x}_i^b, \bar{x}_i$	Genes de novos indivíduos
K	Número de tentativas para gerar indivíduos no domínio Ω
T	Número máximo de gerações
\mathbf{P}_t	População na geração t
\mathbf{x}_t^k	Indivíduo k de uma população na geração t
$\bar{\mathbf{P}}_t$	População intermediária na geração t
M	Tamanho da população intermediária
p_c	Probabilidade de cruzamento
p_m	Probabilidade de mutação
\mathbf{x}_t^*	Melhor indivíduo de cada geração t
S	Número de gerações para cálculo do desvio padrão, empregado no critério de parada
E	Módulo de elasticidade
ν	Coefficiente de Poisson
p	Tensão hidrostática
q	Tensão equivalente de von Mises

1 Introdução

Formular uma lei constitutiva que represente adequadamente o comportamento real dos materiais de engenharia é uma tarefa complexa. Em especial, os geomateriais requerem a descrição fenômenos microestruturais. Uma alternativa para modelar geomateriais é utilizar modelos elastoplásticos, já que a fissuração e o colapso de poros são fenômenos da microescala que podem ser aproximados por modelos de deformação plástica na mesoescala.

Diversos modelos constitutivos são propostos para simular o comportamento de materiais como concreto, solos e rochas, cada um com suas peculiaridades. Dentre esses modelos, têm-se aqueles cujas envoltórias são definidas por múltiplas superfícies de plastificação. Um exemplo é o *Cap Model*. Em contraposição a modelos clássicos como Mohr-Coulomb e Drucker-Prager, o *Cap Model* apresenta delimitação das tensões hidrostáticas compressivas.

A aplicação da elastoplasticidade ganhou expressão com o avanço dos métodos numéricos. Dentre esses, o método dos elementos finitos é o mais largamente utilizado para solução de problemas com não linearidade do comportamento do material. Em se tratando dos algoritmos para atualização das tensões de materiais elastoplásticos, Ortiz e Popov (1985) introduzem duas famílias de algoritmos de integração ao contexto da elastoplasticidade. São elas: a regra de integração trapezoidal generalizada e a de integração no ponto médio generalizado. Por serem formulações genéricas, são capazes de representar a integração explícita e implícita de Euler.

Os métodos explícitos são largamente empregados em problemas geomecânicos. Sloan (1987) discute dois esquemas de integração explícitos com subincrementação para problemas elastoplásticos. A subincrementação confere mais acurácia ao algoritmo, todavia exige mais esforço computacional. Jakobsen e Lade (2002) apresentam um algoritmo explícito para atualização das tensões por meio de um algoritmo de integração explícito. Todavia os autores destacam que

para uma boa precisão do algoritmo são necessários pequenos incrementos de deformações.

Na década de 1980, os métodos de integração implícitos foram aprimorados. Por suas características de estabilidade e convergência, esses métodos ganharam interesse da comunidade científica, podendo ser citados os trabalhos de Simo e Taylor (1985) e Ortiz e Simo (1986). Todavia destaca-se que a implementação dessa classe de algoritmo é mais complexa do que os algoritmos explícitos.

Um ponto importante a ser destacado é que, à medida que a envoltória de plastificação torna-se mais complexa ou mais superfícies de plastificação são adicionadas ao critério de plasticidade, mais parâmetros são necessários para defini-lo. Isso pode ser um transtorno, pois definir esses parâmetros não é uma tarefa simples, principalmente quando esses não podem ser medidos direta ou indiretamente por meio de ensaios.

Pelas razões apresentadas acima, é muito comum encontrar trabalhos na literatura que descrevem o comportamento de materiais como concreto, rochas e solos por intermédio do critério de Mohr-Coulomb ou Drucker-Prager. Podemos citar por exemplo os trabalhos apresentados por Huang e Griffiths (2008), Huang e Griffiths (2009) e Yazdi, Kalantary e Yazdi (2012). A vantagem desses critérios é que os mesmos possuem apenas dois parâmetros que definem a envoltória de plastificação. Todavia, quando se deseja modelar comportamentos mais complexos do material, é necessário adotar modelos mais elaborados que possuem uma quantidade maior de parâmetros.

Historicamente, foram empregados, para determinação de parâmetros de modelos constitutivos, métodos de tentativa e erro, além de métodos de ajuste de curva. Em anos mais recentes, métodos de otimização trouxeram avanços na solução de problemas inversos, como por exemplo, na obtenção de parâmetros de tais modelos. Dentre esses métodos, destacam-se aqueles classificados como métodos de otimização estocástica por serem mais robustos que os algoritmos determinísticos na busca de uma solução global.

Dessa forma, de modo a viabilizar o emprego de modelos complexos para a descrição do comportamento de materiais, este trabalho se propõe a discutir um algoritmo de integração das equações de evolução elastoplásticas, considerando múltiplas superfícies de plastificação. Também é apresentada uma metodologia para calibração de parâmetros a partir dos princípios da análise inversa. É

empregado um algoritmo de otimização para solucionar um problema inverso e determinar os parâmetros desconhecidos a partir de dados observados, usualmente extraídos de ensaios.

1.1. Motivação

A opção por modelos constitutivos mais simples em problemas de engenharia se deve muitas vezes à dificuldade de formulação e implementação dos algoritmos necessários para modelos mais complexos e à dificuldade de determinar os valores dos parâmetros de tais modelos, já que frequentemente estes não têm significado físico.

Entretanto quando se deseja incorporar à descrição do comportamento do material fenômenos como esmagamento do concreto e dilatância de solos a complexidade dos modelos aumenta.

Em se tratando da implementação de algoritmos de integração das equações de evolução elastoplásticas, a convergência dos mesmos pode ser prejudicada à medida que o modelo constitutivo se torna complexo. Autores como Huang e Griffiths (2008) e Scherzinger (2017) reportam tais problemas em seus trabalhos.

Dessa forma, este trabalho tem o intuito de contribuir para a utilização de modelos constitutivos mais complexos, sobretudo modelos com múltiplas superfícies de plastificação em conjunto com o método dos elementos finitos.

1.2. Objetivos

Objetiva-se neste trabalho formular e desenvolver um algoritmo para integração das leis de evolução das variáveis plásticas com melhores características de convergência para modelos elastoplásticos com múltiplas superfícies de plastificação. Adicionalmente, de modo a permitir a aplicação de modelos com muitos parâmetros, propõe-se formalizar uma metodologia para calibração de parâmetros.

1.3. Estrutura da Dissertação

Este trabalho está estruturado em sete capítulos de modo a apresentar os aspectos básicos da integração e calibração de modelos constitutivos elastoplásticos com múltiplas superfícies de plastificação e os algoritmos desenvolvidos. Esta introdução compõe o Capítulo 1 do trabalho.

No Capítulo 2, são descritos os conceitos básicos da teoria da plasticidade empregados no trabalho. Ademais, apresentam-se as equações de evolução elastoplásticas, que precisam ser integradas para que seja possível determinar a evolução das variáveis de estado em uma análise incremental.

No Capítulo 3, é descrito o algoritmo de projeção ao ponto mais próximo, que é empregado na atualização das tensões e cálculo de outras variáveis de estado. Apresenta-se uma interpretação do algoritmo por meio de um problema de otimização. É possível adicionar o processo da busca unidimensional ao algoritmo para melhorar as características de convergência. Ainda é apresentada a decomposição em valores singulares, empregada na inversão de matrizes, para contornar problemas de mal condicionamento. Por fim, propõe-se uma expressão para o cálculo do tensor constitutivo elastoplástico consistente.

No Capítulo 4, descreve-se a metodologia empregada na solução de problemas inversos, em que os parâmetros de entrada são desconhecidos, mas as respostas não são. Apresenta-se uma revisão bibliográfica de trabalhos que empregam tal metodologia e os princípios básicos a serem seguidos.

No Capítulo 5, apresenta-se um algoritmo genético, empregado na análise inversa para minimizar uma função de erro, a fim de obter os parâmetros ótimos para o problema. Apresentam-se os conceitos básicos desse tipo de algoritmo e a implementação do mesmo. Conclui-se o capítulo com um estudo de alguns parâmetros necessários para o algoritmo genético, a partir de um grupo de problemas de otimização global.

No Capítulo 6, são apresentadas as aplicações desenvolvidas neste trabalho para os algoritmos apresentados. Para o algoritmo de integração das equações de evolução elastoplásticas, são analisados problemas locais (a nível do ponto de Gauss) e problemas globais (envolvendo solução via elementos finitos), aplicados ao modelo elastoplástico *Cap Model*. Os resultados são comparados com outras

metodologias de cálculo ou com resultados disponíveis na literatura. Para apresentar a metodologia de calibração de parâmetros estudada, realiza-se uma calibração do modelo *Cap Model*, a partir de dados experimentais.

No Capítulo 7, expõem-se as conclusões desta dissertação, além de sugestões para trabalhos futuros.

O trabalho também é composto por três apêndices. O primeiro, Apêndice A, apresenta resultados gráficos e tabulares obtidos durante o estudo paramétrico do algoritmo genético. O Apêndice B apresenta as equações necessárias para implementar o *Cap Model*. Por fim, o Apêndice C expõe a comparação das curvas tensão-deformação experimentais e calibradas para o *Cap Model*.

2 Tópicos de Teoria da Plasticidade

Os materiais empregados na Engenharia Civil, como concreto, rochas e solos estão submetidos a fenômenos em diversas escalas de observação. Os fenômenos micromecânicos são aqueles que ocorrem no nível dos grãos e das partículas, podendo ser representados por modelos físicos. Já no nível macromecânico, desprezam-se as heterogeneidades presentes no material, e emprega-se para a descrição do comportamento do material modelos fenomenológicos. Esta abordagem é comumente empregada devido à complexidade dos modelos que se baseiam nos fenômenos que ocorrem na micromecânica. Representar cada fenômeno que ocorre na microescala é uma tarefa complexa. Dessa forma, é interessante agrupar esses fenômenos e analisá-los na macroescala.

Dentre as abordagens fenomenológicas disponíveis, tem-se os modelos fundamentados na teoria da plasticidade. Esta teoria fornece as bases para descrever e aproximar o comportamento de diversos tipos de materiais, podendo ser citados, metais, rochas, concreto e solos. Fundamentalmente, esta teoria se dedica a descrever sólidos sujeitos a deformações permanentes (plásticas) quando submetidos a carregamento e descarregamento. O colapso de poros, em materiais porosos, é um fenômeno na microescala que pode ser representado como deformações plásticas. A teoria da plasticidade se limita a descrever problemas em que tais deformações não são dependentes da taxa de aplicação de carga.

O objetivo da teoria da plasticidade é fornecer modelos constitutivos capazes de descrever com suficiente precisão, de forma qualitativa e quantitativa, o comportamento fenomenológico de determinados materiais. À vista disso, a fundamentação teórica para tal são descritos nas seções subsequentes.

2.1. Problema Elastoplástico Generalizado

Souza Neto, Perić e Owen (2008) descrevem as componentes básicas da formulação matemática de um modelo constitutivo elastoplástico generalizado. Tais etapas são descritas abaixo:

- Decomposição do tensor de deformações;
- Relação elástica entre tensão e deformação;
- Critério de plastificação, descrito por funções (superfícies) de plastificação;
- Regra de fluxo plástico, que define a evolução da deformação plástica;
- Lei de endurecimento/amolecimento, que define a evolução dos limites da região elástica.

A seguir são apresentados cada um desses conceitos, aplicados à problemas bidimensionais e tridimensionais.

2.1.1. Decomposição do Tensor de Deformações

Considerando o regime de pequenas deformações, uma das hipóteses fundamentais da teoria da plasticidade clássica é que o tensor de deformações totais pode ser decomposto de forma aditiva em uma parcela elástica ($\boldsymbol{\varepsilon}^e$) e plástica ($\boldsymbol{\varepsilon}^p$), que representam a parcela reversível e irreversível das deformações totais, respectivamente. Dessa forma, tem-se a expressão a seguir:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \quad (2.1)$$

O tensor $\boldsymbol{\varepsilon}^e$ é denominado de tensor das deformações elásticas e $\boldsymbol{\varepsilon}^p$, tensor das deformações plásticas. Esta mesma relação aditiva pode ser posta em forma de taxa, conforma apresentado a seguir:

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^e + \dot{\boldsymbol{\varepsilon}}^p \quad (2.2)$$

Destaca-se que a expressão (2.2) exige uma condição inicial, definida por:

$$\boldsymbol{\varepsilon}(t_0) = \boldsymbol{\varepsilon}^e(t_0) + \boldsymbol{\varepsilon}^p(t_0) \quad (2.3)$$

em que t_0 representa um pseudotempo inicial.

2.1.2. Relação Elástica entre Tensão e Deformação

O tensor de tensões $\boldsymbol{\sigma}$ está relacionado com o tensor de deformações elásticas por intermédio da energia potencial elástica, dada pela função escalar $W(\boldsymbol{x}, \boldsymbol{\varepsilon}^e(\boldsymbol{x}, t))$, em que \boldsymbol{x} representa a posição de um ponto num corpo e t o tempo. Dessa forma, a relação hiperelástica é dada por:

$$\boldsymbol{\sigma}(\boldsymbol{x}, t) = \frac{\partial W(\boldsymbol{x}, \boldsymbol{\varepsilon}^e(\boldsymbol{x}, t))}{\partial \boldsymbol{\varepsilon}^e} \quad (2.4)$$

ou ainda na forma:

$$\boldsymbol{\sigma} = \nabla W(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) \quad (2.5)$$

Para a elasticidade linear, tem-se que $W(\boldsymbol{x}, \boldsymbol{\varepsilon}^e(\boldsymbol{x}, t))$ é uma forma quadrática dada por:

$$W = \frac{1}{2} \boldsymbol{\varepsilon}^e : \boldsymbol{D} : \boldsymbol{\varepsilon}^e \quad (2.6)$$

considerando que \boldsymbol{D} é tensor constitutivo elástico. Sendo assim, a relação (2.4) se torna:

$$\boldsymbol{\sigma} = \boldsymbol{D} : \boldsymbol{\varepsilon}^e \quad (2.7)$$

que também pode ser expressa, com base na decomposição das deformações plásticas (2.1), na forma que segue:

$$\boldsymbol{\sigma} = \boldsymbol{D} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) \quad (2.8)$$

É interessante também definir a inversa da matriz \mathbf{D} , para aplicações futuras, como $\mathbf{C} = \mathbf{D}^{-1}$, o tensor que transforma tensões em deformações elásticas:

$$\boldsymbol{\varepsilon}^e = \mathbf{C} : \boldsymbol{\sigma} \quad (2.9)$$

2.1.3. Critério de Plastificação

A delimitação do domínio elástico de um problema elastoplástico generalizado é feita por meio de um critério de plastificação. Tais critérios são definidos por hipersuperfícies no espaço das tensões. Estas são propostas de acordo com o comportamento de cada material.

Classicamente, critérios como os de von Mises e Drucker-Prager, por exemplo, são representados por uma superfície de plastificação $f(\boldsymbol{\sigma}, \mathbf{q}) = 0$, sendo \mathbf{q} um tensor relacionado ao fenômeno do endurecimento e amolecimento. Sendo assim, o domínio elástico de tais problemas pode ser definido da forma que se segue:

$$\mathbb{E} = \{(\boldsymbol{\sigma}, \mathbf{q}) \mid f(\boldsymbol{\sigma}, \mathbf{q}) < 0\} \quad (2.10)$$

A superfície de plastificação limita (ou restringe) o domínio elástico e é definida pelo conjunto:

$$\mathbb{Y} = \{(\boldsymbol{\sigma}, \mathbf{q}) \mid f(\boldsymbol{\sigma}, \mathbf{q}) = 0\} \quad (2.11)$$

Assim sendo, é inadmissível um estado de tensões que atenda a relação $f(\boldsymbol{\sigma}, \mathbf{q}) > 0$. Da união dos conjuntos representados por (2.10) e (2.11), tem-se o conjunto das tensões plasticamente admissíveis, conforme apresentado a seguir:

$$\bar{\mathbb{E}} = \{(\boldsymbol{\sigma}, \mathbf{q}) \mid f(\boldsymbol{\sigma}, \mathbf{q}) \leq 0\} \quad (2.12)$$

Dentre a grande variedade de modelos constitutivos elastoplásticos disponíveis na literatura, existem também aqueles que são definidos por múltiplas superfícies de plastificação $f_i(\boldsymbol{\sigma}, \mathbf{q}) = 0$, em que $i \in \{1, 2, \dots, m\}$. O domínio

elástico desses modelos é definido pela interseção de múltiplas regiões, como descrito a seguir:

$$\mathbb{E} = \{(\boldsymbol{\sigma}, \boldsymbol{q}) \mid f_i(\boldsymbol{\sigma}, \boldsymbol{q}) < 0, \forall i \in \{1, 2, \dots, m\}\} \quad (2.13)$$

Dessa forma, o contorno do domínio elástico composto pelas múltiplas superfícies de plastificação e o conjunto das tensões plasticamente admissíveis são representados, respectivamente, por:

$$\mathbb{Y} = \{(\boldsymbol{\sigma}, \boldsymbol{q}) \mid f_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0, \forall i \in \{1, 2, \dots, m\}\} \quad (2.14)$$

$$\bar{\mathbb{E}} = \{(\boldsymbol{\sigma}, \boldsymbol{q}) \mid f_i(\boldsymbol{\sigma}, \boldsymbol{q}) \leq 0, \forall i \in \{1, 2, \dots, m\}\} \quad (2.15)$$

A Figura 2.1 ilustra o domínio elástico de um problema elastoplástico com múltiplas superfícies de plastificação. Além disso, nota-se na imagem a existência de interseções não suaves entre as superfícies, que é um aspecto comum em diversos modelos constitutivos descritos na literatura.

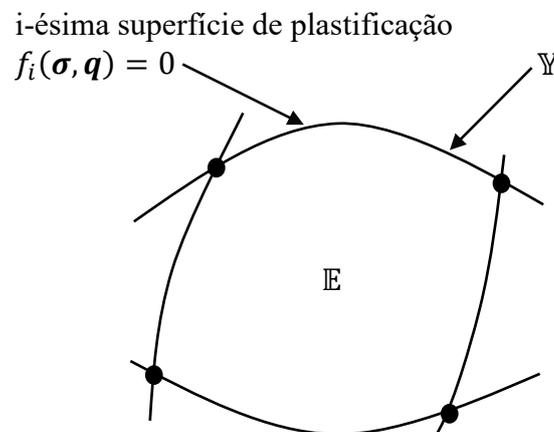


Figura 2.1 – Domínio elástico para um modelo elastoplástico com múltiplas superfícies de plastificação

2.1.4. Regra de Fluxo Plástico e Lei de Endurecimento/Amolecimento

A irreversibilidade do fenômeno da plastificação é representada por intermédio de equações que regem a evolução dos tensores $\boldsymbol{\varepsilon}^p$ e \boldsymbol{q} . Fenômenos dissipativos podem ser representados em um modelo constitutivo elastoplástico

mediante a uma lei de evolução das variáveis internas. O tensor das variáveis internas \mathbf{q} é representado, neste trabalho, no espaço das tensões. Tais variáveis têm a capacidade de modificar o domínio elástico, aumentando, diminuindo ou movendo o domínio no hiperespaço das tensões com a evolução do processo de deformação plástica.

A seguir são apresentadas as equações que governam tal evolução, conhecidas como regra de fluxo plástica e lei de endurecimento/amolecimento, para o problema de superfície de plastificação única:

$$\dot{\boldsymbol{\varepsilon}}^p = \gamma \mathbf{n}(\boldsymbol{\sigma}, \mathbf{q}) \quad (2.16)$$

$$\dot{\mathbf{q}} = \gamma \mathbf{h}(\boldsymbol{\sigma}, \mathbf{q}) \quad (2.17)$$

considerando que $\mathbf{n}(\boldsymbol{\sigma}, \mathbf{q})$ e $\mathbf{h}(\boldsymbol{\sigma}, \mathbf{q})$ são funções pré-definidas que definem a direção do fluxo plástico e o tipo de endurecimento/amolecimento. O parâmetro γ é não-negativo e conhecido como multiplicador plástico.

Uma possibilidade usual é adotar a regra de fluxo associada, na qual se considera que a variação de deformação plástica ocorre na direção normal à superfície de plastificação, ou seja:

$$\mathbf{n}(\boldsymbol{\sigma}, \mathbf{q}) = \partial_{\boldsymbol{\sigma}} f(\boldsymbol{\sigma}, \mathbf{q}) \quad (2.18)$$

As equações (2.16) e (2.17) devem satisfazer as condições de complementariedade, apresentadas a seguir:

$$f(\boldsymbol{\sigma}, \mathbf{q}) \leq 0 \quad (2.19)$$

$$\gamma \geq 0 \quad (2.20)$$

$$\gamma f(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad (2.21)$$

O conjunto de equações (2.19), (2.20) e (2.21) também são conhecidos por condições de carregamento/descarregamento ou por condições de Karush-Kuhn-Tucker. Por fim, em adição a tais expressões, tem-se a condição de consistência, descrita a seguir:

$$\gamma \dot{f}(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad (2.22)$$

Sendo inadmissível $f(\boldsymbol{\sigma}, \mathbf{q}) > 0$, as situações de carregamento que podem existir são:

$$\left\{ \begin{array}{l} f < 0 \leftrightarrow (\boldsymbol{\sigma}, \mathbf{q}) \in \mathbb{E} \rightarrow \gamma = 0 \text{ (regime elástico)} \\ f = 0 \leftrightarrow (\boldsymbol{\sigma}, \mathbf{q}) \in \mathbb{Y} \rightarrow \left\{ \begin{array}{l} \dot{f} < 0 \rightarrow \gamma = 0 \text{ (descarregamento elástico)} \\ \dot{f} = 0 \text{ e } \gamma = 0 \text{ (carregamento neutro)} \\ \dot{f} = 0 \text{ e } \gamma > 0 \text{ (carregamento plástico)} \end{array} \right. \end{array} \right. \quad (2.23)$$

Quando se adota um modelo constitutivo com múltiplas superfícies de plastificação $f_i(\boldsymbol{\sigma}, \mathbf{q})$, emprega-se a regra de Koiter, descrita inicialmente por Koiter (1953), para descrever a evolução das deformações plásticas. Adota-se também uma expressão similar desta regra para a evolução das variáveis internas. Em cada uma destas expressões, as funções $\mathbf{n}_i(\boldsymbol{\sigma}, \mathbf{q})$ e $\mathbf{h}_i(\boldsymbol{\sigma}, \mathbf{q})$ são ponderadas pelos multiplicadores plásticos γ_i , referentes à cada superfície de plastificação. As duas expressões são apresentadas a seguir:

$$\boldsymbol{\varepsilon}^p = \sum_{i=1}^m \gamma_i \mathbf{n}_i(\boldsymbol{\sigma}, \mathbf{q}) \quad (2.24)$$

$$\dot{\mathbf{q}} = \sum_{i=1}^m \gamma_i \mathbf{h}_i(\boldsymbol{\sigma}, \mathbf{q}) \quad (2.25)$$

Assim como no caso anterior, aplicam-se às expressões (2.24) e (2.25), as condições de complementariedade a cada superfície:

$$f_i(\boldsymbol{\sigma}, \mathbf{q}) \leq 0 \quad (2.26)$$

$$\gamma_i \geq 0 \quad (2.27)$$

$$\gamma_i f_i(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad (2.28)$$

e a condição de consistência:

$$\gamma_i \dot{f}_i(\boldsymbol{\sigma}, \mathbf{q}) = 0 \quad (2.29)$$

2.1.5. Multiplicador Plástico

O multiplicado plástico γ é um escalar que tem a função de ajustar o módulo dos incrementos de deformação plástica e das variáveis internas, sendo nulo durante o regime elástico. É possível obter o valor desse explorando a expressão (2.22), para o caso de uma superfície de plastificação, definindo-se a derivada de f em relação ao tempo:

$$\dot{f} = \partial_{\sigma} f : \dot{\boldsymbol{\sigma}} + \partial_{\mathbf{q}} f \cdot \dot{\mathbf{q}} \quad (2.30)$$

O símbolo “.” indica um produto apropriado entre $\partial_{\mathbf{q}} f$ e $\dot{\mathbf{q}}$. Assim, obtém-se:

$$\begin{aligned} \dot{f} &= \partial_{\sigma} f : \mathbf{D} : (\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p) + \partial_{\mathbf{q}} f \cdot \dot{\mathbf{q}} \\ &= \partial_{\sigma} f : \mathbf{D} : (\dot{\boldsymbol{\varepsilon}} - \gamma \mathbf{n}) + \partial_{\mathbf{q}} f \cdot \gamma \mathbf{h} \\ &= \partial_{\sigma} f : \mathbf{D} : \dot{\boldsymbol{\varepsilon}} - \gamma (\partial_{\sigma} f : \mathbf{D} : \mathbf{n} - \partial_{\mathbf{q}} f \cdot \mathbf{h}) \end{aligned} \quad (2.31)$$

Para o carregamento plástico, tem-se que $\dot{f} = 0$ para $\gamma > 0$. Logo, obtém-se a forma explícita do multiplicador plástico a partir do resultado obtido em (2.30):

$$\gamma = \frac{\langle \partial_{\sigma} f : \mathbf{D} : \dot{\boldsymbol{\varepsilon}} \rangle}{\partial_{\sigma} f : \mathbf{D} : \mathbf{n} - \partial_{\mathbf{q}} f \cdot \mathbf{h}} \quad (2.32)$$

devendo-se atender a inequação $\partial_{\sigma} f : \mathbf{D} : \mathbf{n} - \partial_{\mathbf{q}} f \cdot \mathbf{h} > 0$ para que $\gamma > 0$.

Quando se consideram múltiplas superfícies de plastificação, introduz-se o conceito de superfícies ativas. As equações (2.26) a (2.29) indicam quais as superfícies de plastificação (ou de um ponto de vista de otimização, restrições) estão ativas. Assim, o conjunto de superfícies de plastificação ativas é apresentado a seguir:

$$\mathbb{J}_{act} = \{j \in \{1, 2, \dots, m\} \mid f_j(\boldsymbol{\sigma}, \mathbf{q}) = 0 \text{ e } \dot{f}_j(\boldsymbol{\sigma}, \mathbf{q}) = 0\} \quad (2.33)$$

O cálculo da derivada em relação ao tempo de f_i é feito como se segue:

$$\begin{aligned}
\dot{f}_i &= \partial_{\sigma} f_i : \dot{\sigma} + \partial_{q} f_i \cdot \dot{q} \\
&= \partial_{\sigma} f_i : \mathbf{D} : (\dot{\epsilon} - \dot{\epsilon}^p) + \partial_{q} f_i \cdot \dot{q} \\
&= \partial_{\sigma} f_i : \mathbf{D} : \left(\dot{\epsilon} - \sum_{j=1}^m \gamma_j \mathbf{n}_j \right) + \partial_{q} f_i \cdot \sum_{j=1}^m \gamma_j \mathbf{h}_j \\
&= \partial_{\sigma} f_i : \mathbf{D} : \dot{\epsilon} - \sum_{j=1}^m \gamma_j (\partial_{\sigma} f_i : \mathbf{D} : \mathbf{n}_j - \partial_{q} f_i \cdot \mathbf{h}_j)
\end{aligned} \tag{2.34}$$

Para o carregamento plástico, tem-se que $\dot{f}_i = 0$ para $\gamma_i > 0$. Logo, obtém-se a forma explícita do multiplicador plástico referente à cada superfície de plastificação a partir do resultado de (2.34):

$$\gamma_i = \sum_{j=1}^m d^{ij} \langle \partial_{\sigma} f_j : \mathbf{D} : \dot{\epsilon} \rangle \tag{2.35}$$

em que:

$$d^{ij} = (\partial_{\sigma} f_i : \mathbf{D} : \mathbf{n}_j - \partial_{q} f_i \cdot \mathbf{h}_j)^{-1} \tag{2.36}$$

2.1.6. Tensor Constitutivo Elastoplástico

É importante determinar a relação entre a taxa de variação das tensões em termos da taxa de deformação total. Tal relação é definida por meio do tensor constitutivo elastoplástico \mathbf{D}^{ep} e expressa a seguir:

$$\dot{\sigma} = \mathbf{D}^{ep} : \dot{\epsilon} \tag{2.37}$$

Calcula-se \mathbf{D}^{ep} a partir da regra de fluxo plástico e na forma explícita do multiplicador plástico, obtidos anteriormente. Sendo assim, quando se considera superfície de plastificação única:

$$\dot{\sigma} = \mathbf{D} : (\dot{\epsilon} - \dot{\epsilon}^p) = \mathbf{D} : (\dot{\epsilon} - \gamma \mathbf{n}) \tag{2.38}$$

Por conseguinte, manipulando as expressões (2.32) e (2.38), obtém-se a seguinte expressão para \mathbf{D}^{ep} , observando a expressão (2.37):

$$\mathbf{D}^{ep} = \begin{cases} \mathbf{D}, & \text{se } \gamma = 0 \\ \mathbf{D} - \frac{\mathbf{D}:\mathbf{n} \otimes \mathbf{D}:\partial_{\sigma}f}{\partial_{\sigma}f:\mathbf{D}:\mathbf{n} - \partial_q f \cdot \mathbf{h}}, & \text{se } \gamma > 0 \end{cases} \quad (2.39)$$

Quando se considera um problema com múltiplas superfícies de plastificação, tem-se:

$$\dot{\boldsymbol{\sigma}} = \mathbf{D}:(\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p) = \mathbf{D}:\left(\dot{\boldsymbol{\varepsilon}} - \sum_{i=1}^m \gamma_i \mathbf{n}_i(\boldsymbol{\sigma}, \mathbf{q})\right) \quad (2.40)$$

Similarmente, manipulando as expressões (2.35) e (2.38), obtém-se a seguinte expressão para \mathbf{D}^{ep} :

$$\mathbf{D}^{ep} = \begin{cases} \mathbf{D}, & \text{se } \mathbb{J}_{act} = \emptyset \\ \mathbf{D} - \sum_{i \in \mathbb{J}_{act}} \sum_{j \in \mathbb{J}_{act}} d^{ij}(\mathbf{D}:\mathbf{n}_i) \otimes (\mathbf{D}:\partial_{\sigma}f_j), & \text{se } \mathbb{J}_{act} \neq \emptyset \end{cases} \quad (2.41)$$

2.2.

Critério de Plastificação com Múltiplas Superfícies de Plastificação: Cap Model

De acordo com Kojić e Bathe (2005), é observado experimentalmente que geomateriais coesivos sofrem deformações plásticas não apenas quando submetidos a forças cisalhantes, mas também quando existem cargas compressivas hidrostáticas agindo. Além disso, observa-se que estas últimas produzem um endurecimento no material e deformações volumétricas permanentes.

Modelos constitutivos como Mohr-Coulomb e Drucker-Prager, idealizados para esse tipo de material, não são capazes de representar tais comportamentos. Dessa forma, a aplicação de tais modelos conduz a um acréscimo de deformação plástica volumétrica excessivas quando comparadas com estudos experimentais. Partindo desta ideia, modelos específicos foram desenvolvidos para considerar tais comportamentos.

Cap Models são modelos compostos por uma superfície fixa, em geral, cônica (ou aproximadamente) e uma superfície que pode alterar de tamanho durante a plastificação. A introdução de uma superfície esférica (*cap*) foi feita por Drucker et al. (1975, *apud* Chen, 2007) no modelo que leva seu próprio nome, para controlar a deformação plástica volumétrica e a dilatação de solos. Desde então, vários outros modelos foram concebidos partindo desta ideia, como o modelo com o *cap* plano descrito por Chen (2007).

O *Cap Model* foi apresentado inicialmente por DiMaggio e Sandler (1971, *apud* Simo et al., 1988) e sua implementação foi discutida inicialmente por Sandler e Rubin (1979). Este modelo é adequado para modelar concreto, rochas e solos. Hofstetter, Simo e Taylor (1993) estudam uma modificação ao *Cap Model* original, considerando uma lei de endurecimento associada. Com esta modificação, é possível construir um tensor constitutivo elastoplástico simétrico; todavia todas as superfícies se tornam dependentes do parâmetro de endurecimento do modelo. Simo e Ju (1987a, 1987b) adicionaram ao modelo uma lei de dano, acoplando ao modelo elastoplástico a mecânica do dano para representar materiais frágeis.

Para definir um modelo constitutivo elastoplástico, é usual empregar os invariantes do tensor de tensões. Dessa forma, define-se:

$$I_1(\boldsymbol{\sigma}) = \text{tr}(\boldsymbol{\sigma}) \quad (2.42)$$

$$J_2(\boldsymbol{\sigma}) = \frac{1}{2} \text{tr}(\mathbf{s}^2(\boldsymbol{\sigma})) \quad (2.43)$$

em que $\mathbf{s}(\boldsymbol{\sigma})$ representa a tensão desviadora, dada por:

$$\mathbf{s}(\boldsymbol{\sigma}) = \boldsymbol{\sigma} - \frac{1}{3} I_1(\boldsymbol{\sigma}) \mathbf{I} \quad (2.44)$$

Das diversas representações desse modelo, apresenta-se o seguinte grupo de equações, que descrevem, respectivamente, a superfície de falha, ao *cap* elíptico e ao *cut-off* de tração, conforme apresentado na Figura 2.2:

$$f_1(\boldsymbol{\sigma}) = \sqrt{2J_2(\boldsymbol{\sigma})} - F_e(I_1(\boldsymbol{\sigma})) \quad (2.45)$$

$$f_2(\boldsymbol{\sigma}, \kappa) = \sqrt{2J_2(\boldsymbol{\sigma}) + \left(\frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R}\right)^2} - F_e(L(\kappa)) \quad (2.46)$$

$$f_3(\boldsymbol{\sigma}) = I_1(\boldsymbol{\sigma}) - T \quad (2.47)$$

considerando que:

$$F_e(I_1) = \alpha - \lambda \exp(\beta I_1) - \theta I_1 \quad (2.48)$$

$$L(\kappa) = \min(\kappa, \kappa_0) \quad (2.49)$$

$$X(\kappa) = L(\kappa) - R F_e(L(\kappa)) \quad (2.50)$$

em que α , β , λ , θ , κ_0 , R e T são parâmetros dos materiais e κ é a variável interna que controla o endurecimento do *cap* e, geometricamente, se refere ao centro desse. A superfície $f_1(\boldsymbol{\sigma})$ é definida para o domínio $L(\kappa) \leq I_1(\boldsymbol{\sigma}) < T$ e superfície $f_2(\boldsymbol{\sigma})$ é definida em $X(\kappa) \leq I_1(\boldsymbol{\sigma}) \leq L(\kappa)$.

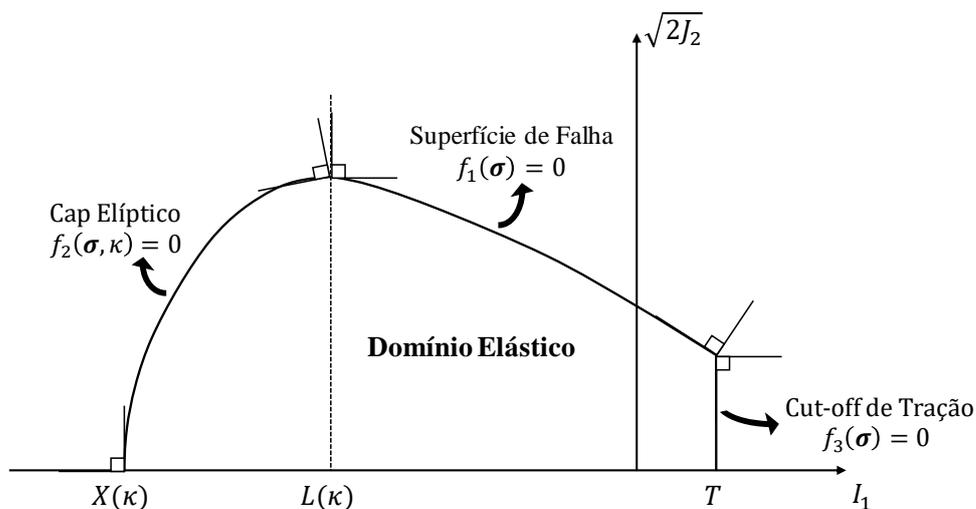


Figura 2.2 – Representação do Cap Model

O endurecimento do material está relacionado à lei de evolução das deformações plásticas volumétricas, conforme a expressão a seguir:

$$\varepsilon_v^p = W(\exp(D(X(\kappa) - X_0)) - 1) \quad (2.51)$$

em que W e D são parâmetros dos materiais e $X_0 = X(\kappa_0)$. As equações (2.45) a (2.51) consideram esforços de tração positivos e compressão, negativos.

3

Desenvolvimento de um Algoritmo de Mapeamento de Retorno para Múltiplas Superfícies de Plastificação

Para implementação de um modelo elastoplástico em um programa de elementos finitos é necessário definir um esquema de integração das equações de evolução elastoplásticas para atualizar as tensões e variáveis de estado do problema. Ademais, é necessário calcular o tensor constitutivo elastoplástico, essencial para a obtenção da matriz de rigidez do elemento finito.

Trabalhos como os de Simo e Taylor (1986), Simo e Hughes (1998) e Huang e Griffiths (2008, 2009) apresentam algoritmos necessários para problemas elastoplásticos com superfície de plastificação única. Scalet e Auricchio (2018) apresentam uma revisão do estado da arte dos algoritmos dos métodos computacionais aplicados à problemas elastoplásticos, discutindo as metodologias convencionais e não convencionais. Dentre os mais clássicos, tem-se o método do passo múltiplo, que abrange o método de Euler implícito, que foi adotado neste trabalho, por ser largamente empregado na literatura.

Em se tratando de problemas com múltiplas superfícies de plastificação, Simo, Kennedy e Govindjee (1988) apresentam um algoritmo para problemas e múltiplas superfícies, com a possibilidade da transição não suave entre cada uma destas. Através desse, problemas genéricos podem ser implementados, sem ser necessário particularizar o algoritmo para modelos distintos. Outras pesquisas que também desenvolvem avanços nesta linha são os trabalhos de Mackenzie-Helnwein, Eberhardsteiner e Mang (2003), Karaoulanis (2013) e Godio *et al.* (2016).

Alguns autores estudaram algoritmos aplicados a problemas com múltiplas superfícies de plastificação planas no espaço das tensões principais, podendo ser citados Pankaj e Bićanić (1997), Clausen, Damkilde e Andersen (2006) e Saksala (2009). A aplicação mais básica desses algoritmos está em resolver problemas que empregam os critérios clássicos de Tresca e Mohr-Coulomb.

Na literatura, encontram-se alguns estudos de algoritmos de integração de tensões para problemas com múltiplas superfícies de plastificação particularizados para problemas específicos. Dentre esses estão Lourenço e Rots (1997), que apresentam um algoritmo específico para duas superfícies de plastificação para estudar as interfaces de alvenarias; Lourenço, de Borst e Rots (1997), que estudam um algoritmo para o estado plano de tensões com duas superfícies de plastificação; e Souza Neto, Perić e Owen (2008), que apresenta algoritmos particularizados para os critérios de Tresca, Mohr-Coulomb e Drucker-Prager com *cap*.

Alguns autores desenvolvem seus estudos através de algoritmos de otimização para solucionar problemas de plasticidade. Eboli, Vaz e Vargas Júnior (1996) discutem estratégias de atualização das tensões para problemas elastoplásticos associados e não associados por meio do método de Newton-Raphson, programação linear complementar (algoritmo de Lemke) e programação quadrática sequencial (algoritmo de Han-Powell). Krabbenhoft *et al.* (2007) apresentam um algoritmo para solucionar problemas elastoplásticos com base no método dos pontos interiores, sendo o algoritmo capaz de tratar problemas com múltiplas superfícies de plastificação. Bruno *et al.* (2017) estudam o emprego da programação cônica para o problema de mapeamento de retorno para os critérios de von Mises, Tresca, Drucker-Prager e Mohr-Coulomb, formulados através de representações cônicas dos mesmos.

Adhikary *et al.* (2017) reportam que a dependência linear das direções dos fluxos plásticos entre si é um problema que deve ser tratado corretamente no algoritmo de integração. Essa situação também foi observada durante as implementações deste trabalho e o tratamento dado será discutida neste trabalho.

Isto posto, neste trabalho adota-se o algoritmo de integração implícita, conhecido como projeção ao ponto mais próximo (em inglês, *closest point projection*) e sua versão para múltiplas superfícies de plastificação genéricas, apresentado inicialmente por Simo, Kennedy e Govindjee (1988). A descrição do algoritmo é feita a seguir, juntamente às modificações propostas de acordo com observações encontradas na literatura e durante a implementação do mesmo. Frisa-se que a formulação do algoritmo apresentada neste trabalho é mais geral que a apresentada por Simo, Kennedy e Govindjee (1988), permitindo plasticidade e endurecimento/amolecimento não associados.

3.1. Definições Básicas

A solução numérica de problemas de plasticidade computacional é baseada em esquemas de integração das equações de evolução elastoplásticas. Duas famílias de algoritmos de integração são largamente abordadas na literatura: regra do ponto médio generalizado e regra do trapézio generalizado. Desses dois ainda é possível particularizar duas abordagens bastante usuais entre os algoritmos de integração: implícitos e explícitos.

Ortiz e Popov (1985) desenvolvem um estudo importante acerca da estabilidade dos algoritmos de integração empregados na atualização das tensões e variáveis internas. Segundo eles, métodos implícitos são incondicionalmente estáveis, diferentemente dos explícitos, cuja estabilidade depende do tamanho do incremento (variação das deformações totais).

Dessa forma, é importante selecionar um esquema de integração adequado a fim de atualizar tensões e variáveis de estado em um problema elastoplástico. A ideia básica da solução de um problema de plasticidade computacional é apresentada na Figura 3.1.

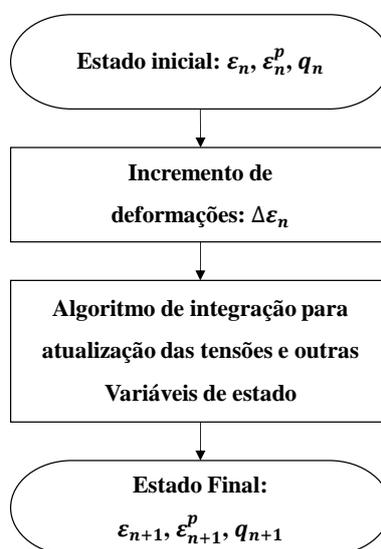


Figura 3.1 – Processo de atualização de estado de um problema elastoplástico

O algoritmo de integração a ser apresentado neste trabalho é baseado na estratégia preditor/corretor, que consiste em dois passos:

- Preditor elástico: Supor que todo incremento de deformações $\Delta \boldsymbol{\varepsilon}_n$ produz um estado de tensões $\boldsymbol{\sigma}_{n+1}$ que está contido no conjunto de tensões plasticamente admissíveis, ou seja, $\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p$ e $\mathbf{q}_{n+1} = \mathbf{q}_n$;
- Corretor plástico: Caso a suposição anterior não seja verdadeira, deve-se atualizar o estado de tensões e as variáveis internas, de modo que $f_i(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \leq 0$.

Considerando um problema mais geral com plasticidade não associada com encruamento para múltiplas superfícies de plastificação, o problema infinitesimal a ser resolvido pela estratégia de preditor-corretor é dado por:

$$\begin{array}{ccc}
 \text{Total:} & \text{Preditor elástico:} & \text{Corretor plástico:} \\
 \left. \begin{array}{l} \dot{\boldsymbol{\varepsilon}} = \nabla^s(\dot{\mathbf{u}}) \\ \dot{\boldsymbol{\varepsilon}}^p = \sum_{i=1}^m \gamma_i \mathbf{n}_i(\boldsymbol{\sigma}, \mathbf{q}) \\ \dot{\mathbf{q}} = \sum_{i=1}^m \gamma_i \mathbf{h}_i(\boldsymbol{\sigma}, \mathbf{q}) \end{array} \right\} = & \left. \begin{array}{l} \dot{\boldsymbol{\varepsilon}} = \nabla^s(\dot{\mathbf{u}}) \\ \dot{\boldsymbol{\varepsilon}}^p = \mathbf{0} \\ \dot{\mathbf{q}} = \mathbf{0} \end{array} \right\} + & \left. \begin{array}{l} \dot{\boldsymbol{\varepsilon}} = \mathbf{0} \\ \dot{\boldsymbol{\varepsilon}}^p = \sum_{i=1}^m \gamma_i \mathbf{n}_i(\boldsymbol{\sigma}, \mathbf{q}) \\ \dot{\mathbf{q}} = \sum_{i=1}^m \gamma_i \mathbf{h}_i(\boldsymbol{\sigma}, \mathbf{q}) \end{array} \right\} \quad (3.1)
 \end{array}$$

em que \mathbf{u} representa o campo de deslocamentos e m indica o número de superfícies de plastificação. Partindo das condições iniciais $\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{\varepsilon}_n^p$ e \mathbf{q}_n , pode-se aplicar o método de Euler implícito nas expressões de (3.1), conforme apresentado por Simo e Hughes (1998). Assim sendo, ao integrar-se implicitamente a regra de fluxo plástico e a lei de endurecimento/amolecimento, obtém-se:

$$\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p + \Delta t \sum_{i=1}^m \gamma_{i,n+1} \mathbf{n}_i(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \quad (3.2)$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \Delta t \sum_{i=1}^m \gamma_{i,n+1} \mathbf{h}_i(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \quad (3.3)$$

Por comodidade, define-se que $\Delta \gamma_{i,n+1} = \gamma_{i,n+1} \Delta t$, $\mathbf{n}_{i,n+1} = \mathbf{n}_i(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1})$ e $\mathbf{h}_{i,n+1} = \mathbf{h}_i(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1})$. Assim, as equações apresentadas em (3.1) se tornam, mediante ao método de Euler implícito:

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n + \nabla^S(\Delta \mathbf{u}) \quad (3.4)$$

$$\boldsymbol{\sigma}_{n+1} = \nabla W(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^p) \quad (3.5)$$

$$\boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p + \sum_{i=1}^m \Delta \gamma_{i,n+1} \mathbf{n}_{i,n+1} \quad (3.6)$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \sum_{i=1}^m \Delta \gamma_{i,n+1} \mathbf{h}_{i,n+1} \quad (3.7)$$

sujeito às condições:

$$f_i(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \leq 0 \quad (3.8)$$

$$\Delta \gamma_{i,n+1} \geq 0 \quad (3.9)$$

$$\Delta \gamma_{i,n+1} f_i(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) = 0 \quad (3.10)$$

Dessa forma, para calcular o preditor elástico, faz-se:

$$\boldsymbol{\varepsilon}_{n+1}^{e\ trial} = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n^p \quad (3.11)$$

$$\boldsymbol{\sigma}_{n+1}^{trial} = \nabla W(\boldsymbol{\varepsilon}_{n+1}^{e\ trial}) \quad (3.12)$$

$$f_{i,n+1}^{trial} = f_i(\boldsymbol{\sigma}_{n+1}^{trial}, \mathbf{q}_n) \quad (3.13)$$

Se $f_{i,n+1}^{trial} \leq 0$, o incremento de deformações levou a um estado elástico, ou seja, $\Delta \gamma_{i,n+1} = 0$. Caso $f_{i,n+1}^{trial} > 0$, o incremento de deformações é elastoplástico, logo, $\Delta \gamma_{i,n+1} > 0$. Para esta última situação, deve-se fazer a correção plástica, que é feita a partir do algoritmo apresentado a seguir.

3.1.1. Formulação do Algoritmo de Projeção ao Ponto mais Próximo

A formulação do algoritmo de projeção ao ponto mais próximo é obtida solucionando o sistema de equações não lineares a seguir:

$$\begin{cases} \boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p + \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \mathbf{n}_{i,n+1} \\ \mathbf{q}_{n+1} = \mathbf{q}_n + \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \mathbf{h}_{i,n+1} \\ \mathbf{f}_{n+1} = \mathbf{0} \end{cases} \quad (3.14)$$

considerando que:

$$\mathbf{f}_{n+1} = \begin{bmatrix} f_{i_1,n+1} \\ f_{i_2,n+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} f_{i_1}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \\ f_{i_2}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}) \\ \vdots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act} \quad (3.15)$$

Aplica-se o método de Newton-Raphson para solucionar o sistema não linear de equações. Para tal, definem-se os resíduos a ser minimizado no processo iterativo:

$$\begin{aligned} \mathbf{r}_{1,n+1}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) &= \mathbf{r}_{1,n+1} \\ &= \mathbf{C}_{n+1} : \boldsymbol{\sigma}_{n+1} - \boldsymbol{\varepsilon}_{n+1} + \boldsymbol{\varepsilon}_n^p + \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \mathbf{n}_{i,n+1} \end{aligned} \quad (3.16)$$

$$\begin{aligned} \mathbf{r}_{2,n+1}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) &= \mathbf{r}_{2,n+1} \\ &= -\mathbf{q}_{n+1} + \mathbf{q}_n + \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \mathbf{h}_{i,n+1} \end{aligned} \quad (3.17)$$

em que \mathbf{C}_{n+1} é o tensor que transforma tensões em deformações, conforme escrito na Seção 2.1.2, e:

$$\Delta\boldsymbol{\gamma}_{n+1} = \begin{bmatrix} \Delta\gamma_{1,n+1} \\ \Delta\gamma_{2,n+1} \\ \vdots \\ \Delta\gamma_{m,n+1} \end{bmatrix} \quad (3.18)$$

Assim, o sistema a ser solucionado se resume a:

$$\mathbf{r}_{n+1}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) = \mathbf{0} \quad (3.19)$$

em que:

$$\mathbf{r}_{n+1}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1}) = \begin{bmatrix} \mathbf{r}_{1,n+1} \\ \mathbf{r}_{2,n+1} \\ \mathbf{f}_{n+1} \end{bmatrix} \quad (3.20)$$

As equações apresentadas em (3.20) devem ser linearizadas a partir de séries de Taylor, a fim de se obter a fórmula recorrente do método de Newton-Raphson. Para isso, as variáveis $\Delta\boldsymbol{\sigma}_{n+1}^{(k)}$, $\Delta\mathbf{q}_{n+1}^{(k)}$ e $\Delta^2\boldsymbol{\gamma}_{n+1}^{(k)}$ representam as correções das tensões, variáveis internas e multiplicador plástico, respectivamente, da iteração k do algoritmo de projeção ao ponto mais próximo. Sendo assim, tem-se:

$$\begin{aligned} \mathbf{r}_{1,n+1} &\approx \mathbf{r}_{1,n+1}^{(k)} + \partial_{\boldsymbol{\sigma}}\mathbf{r}_{1,n+1}^{(k)}:\Delta\boldsymbol{\sigma}_{n+1}^{(k)} + \partial_{\mathbf{q}}\mathbf{r}_{1,n+1}^{(k)}\cdot\Delta\mathbf{q}_{n+1}^{(k)} \\ &\quad + \partial_{\Delta\boldsymbol{\gamma}}\mathbf{r}_{1,n+1}^{(k)}:\Delta^2\boldsymbol{\gamma}_{n+1}^{(k)} \end{aligned} \quad (3.21)$$

$$\begin{aligned} \mathbf{r}_{2,n+1} &\approx \mathbf{r}_{2,n+1}^{(k)} + \partial_{\boldsymbol{\sigma}}\mathbf{r}_{2,n+1}^{(k)}:\Delta\boldsymbol{\sigma}_{n+1}^{(k)} + \partial_{\mathbf{q}}\mathbf{r}_{2,n+1}^{(k)}\cdot\Delta\mathbf{q}_{n+1}^{(k)} \\ &\quad + \partial_{\Delta\boldsymbol{\gamma}}\mathbf{r}_{2,n+1}^{(k)}:\Delta^2\boldsymbol{\gamma}_{n+1}^{(k)} \end{aligned} \quad (3.22)$$

$$\mathbf{f}_{n+1} \approx \mathbf{f}_{n+1}^{(k)} + \partial_{\boldsymbol{\sigma}}\mathbf{f}_{n+1}^{(k)}:\Delta\boldsymbol{\sigma}_{n+1}^{(k)} + \partial_{\mathbf{q}}\mathbf{f}_{n+1}^{(k)}\cdot\Delta\mathbf{q}_{n+1}^{(k)} \quad (3.23)$$

em que:

$$\Delta^2\boldsymbol{\gamma}_{n+1}^{(k)} = \begin{bmatrix} \Delta^2\gamma_{i_1,n+1}^{(k)} \\ \Delta^2\gamma_{i_2,n+1}^{(k)} \\ \vdots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)} \quad (3.24)$$

$$\partial_{\boldsymbol{\sigma}}\mathbf{f}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{\sigma}}f_{i_1,n+1}^{(k)} & \partial_{\boldsymbol{\sigma}}f_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)} \quad (3.25)$$

$$\partial_{\mathbf{q}}\mathbf{f}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\mathbf{q}}f_{i_1,n+1}^{(k)} & \partial_{\mathbf{q}}f_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)} \quad (3.26)$$

Definindo-se \mathbf{I} como o tensor identidade, as derivadas presentes nas equações (3.21) e (3.22) são apresentadas a seguir:

$$\partial_{\boldsymbol{\sigma}}\mathbf{r}_{1,n+1}^{(k)} = \mathbf{C}_{n+1}^{(k)} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{\sigma}}\mathbf{n}_{i,n+1}^{(k)} \quad (3.27)$$

$$\partial_q \mathbf{r}_{1,n+1}^{(k)} = \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_q \mathbf{n}_{i,n+1}^{(k)} \quad (3.28)$$

$$\partial_{\Delta \gamma} \mathbf{r}_{1,n+1}^{(k)} = \mathbf{N}_{n+1}^{(k)} \quad (3.29)$$

$$\partial_{\sigma} \mathbf{r}_{2,n+1}^{(k)} = \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_{\sigma} \mathbf{h}_{i,n+1}^{(k)} \quad (3.30)$$

$$\partial_q \mathbf{r}_{2,n+1}^{(k)} = -\mathbf{I} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_q \mathbf{h}_{i,n+1}^{(k)} \quad (3.31)$$

$$\partial_{\Delta \gamma} \mathbf{r}_{n+1}^{2(k)} = \mathbf{H}_{n+1}^{(k)} \quad (3.32)$$

considerando que:

$$\mathbf{N}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{n}_{i_1,n+1}^{(k)} & \mathbf{n}_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)} \quad (3.33)$$

$$\mathbf{H}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{h}_{i_1,n+1}^{(k)} & \mathbf{h}_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)} \quad (3.34)$$

Já que $\mathbf{n}_{i,n+1}^{(k)}$ e $\mathbf{h}_{i,n+1}^{(k)}$ são tensores, $\mathbf{N}_{n+1}^{(k)}$ e $\mathbf{H}_{n+1}^{(k)}$ são representados computacionalmente por matrizes. Das equações (3.21), (3.22) e (3.23), é possível obter o sistema matricial que é empregado para atualizar as variáveis do problema.. Esse sistema é representado na equação (3.35), sendo $\mathcal{J}_{n+1}^{(k)}$ a matriz Jacobiana apresentada em (3.36), representados na forma matricial:

$$\mathbf{r}_{n+1}^{(k)} + \mathcal{J}_{n+1}^{(k)} \begin{bmatrix} \Delta \sigma_{n+1}^{(k)} \\ \Delta \mathbf{q}_{n+1}^{(k)} \\ \Delta^2 \boldsymbol{\gamma}_{n+1}^{(k)} \end{bmatrix} = \mathbf{0} \quad (3.35)$$

$$\mathcal{J}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{C}_{n+1}^{(k)} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_{\sigma} \mathbf{n}_{i,n+1}^{(k)} & \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_q \mathbf{n}_{i,n+1}^{(k)} & \mathbf{N}_{n+1}^{(k)} \\ \left(\sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_{\sigma} \mathbf{h}_{i,n+1}^{(k)} \right)^T & -\mathbf{I} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_q \mathbf{h}_{i,n+1}^{(k)} & \mathbf{H}_{n+1}^{(k)} \\ \left(\partial_{\sigma} \mathbf{f}_{n+1}^{(k)} \right)^T & \left(\partial_q \mathbf{f}_{n+1}^{(k)} \right)^T & \mathbf{0} \end{bmatrix} \quad (3.36)$$

A expressão (3.35) representa a fórmula para determinação das correções de $\boldsymbol{\sigma}_{n+1}^{(k)}$, $\mathbf{q}_{n+1}^{(k)}$ e $\Delta\boldsymbol{\gamma}_{n+1}^{(k)}$ por intermédio do método de Newton-Raphson. Esta equação pode ser reescrita, resultando na equação (3.37), que representa a fórmula de recorrência do problema:

$$\begin{bmatrix} \boldsymbol{\sigma}_{n+1}^{(k+1)} \\ \mathbf{q}_{n+1}^{(k+1)} \\ \Delta\boldsymbol{\gamma}_{n+1}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\sigma}_{n+1}^{(k)} \\ \mathbf{q}_{n+1}^{(k)} \\ \Delta\boldsymbol{\gamma}_{n+1}^{(k)} \end{bmatrix} - \left(\mathbf{J}_{n+1}^{(k)}\right)^{-1} \mathbf{r}_{n+1}^{(k)} \quad (3.37)$$

Partindo de (3.35), obtém-se o seguinte sistema de equações:

$$\begin{bmatrix} \left(\boldsymbol{\Psi}_{n+1}^{(k)}\right)^{-1} & \mathbf{E}_{n+1}^{(k)} \\ \left(\mathbf{G}_{n+1}^{(k)}\right)^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{n+1}^{(k)} \\ \Delta^2\boldsymbol{\gamma}_{n+1}^{(k)} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\rho}_{n+1}^{(k)} \\ \mathbf{f}_{n+1}^{(k)} \end{bmatrix} \quad (3.38)$$

em que:

$$\boldsymbol{\Psi}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{C}_{n+1}^{(k)} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{\sigma}} \mathbf{n}_{i,n+1}^{(k)} & \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1}^{(k)} \partial_{\mathbf{q}} \mathbf{n}_{i,n+1}^{(k)} \\ \left(\sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{\sigma}} \mathbf{h}_{i,n+1}^{(k)} \right)^T & -\mathbf{I} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1}^{(k)} \partial_{\mathbf{q}} \mathbf{h}_{i,n+1}^{(k)} \end{bmatrix}^{-1} \quad (3.39)$$

$$\boldsymbol{\rho}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{r}_{1,n+1}^{(k)} \\ \mathbf{r}_{2,n+1}^{(k)} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\varepsilon}_{n+1}^{(k)} + \boldsymbol{\varepsilon}_n^{(k)} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1}^{(k)} \mathbf{n}_{i,n+1}^{(k)} \\ -\mathbf{q}_{n+1}^{(k)} + \mathbf{q}_n^{(k)} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1}^{(k)} \mathbf{h}_{i,n+1}^{(k)} \end{bmatrix} \quad (3.40)$$

$$\boldsymbol{\delta}_{n+1}^{(k)} = \begin{bmatrix} \Delta\boldsymbol{\sigma}_{n+1}^{(k)} \\ \Delta\mathbf{q}_{n+1}^{(k)} \end{bmatrix} \quad (3.41)$$

$$\mathbf{G}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{\sigma}} \mathbf{f}_{n+1}^{(k)} \\ \partial_{\mathbf{q}} \mathbf{f}_{n+1}^{(k)} \end{bmatrix} \quad (3.42)$$

$$\mathbf{E}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{N}_{n+1}^{(k)} \\ \mathbf{H}_{n+1}^{(k)} \end{bmatrix} \quad (3.43)$$

Com isso, a solução do sistema (3.38) é dada por:

$$\Delta^2 \boldsymbol{\gamma}_{n+1} = \boldsymbol{\Lambda}_{n+1}^{(k)} \left(\boldsymbol{f}_{n+1}^{(k)} - \left(\boldsymbol{G}_{n+1}^{(k)} \right)^T \boldsymbol{\Psi}_{n+1}^{(k)} \boldsymbol{\rho}_{n+1}^{(k)} \right) \quad (3.44)$$

$$\boldsymbol{\delta}_{n+1}^{(k)} = -\boldsymbol{\Psi}_{n+1}^{(k)} \left(\boldsymbol{\rho}_{n+1}^{(k)} + \boldsymbol{E}_{n+1}^{(k)} \Delta^2 \boldsymbol{\gamma}_{n+1}^{(k)} \right) \quad (3.45)$$

considerando que:

$$\boldsymbol{\Lambda}_{n+1}^{(k)} = \left(\left(\boldsymbol{G}_{n+1}^{(k)} \right)^T \boldsymbol{\Psi}_{n+1}^{(k)} \boldsymbol{E}_{n+1}^{(k)} \right)^{-1} \quad (3.46)$$

Assim, à cada iteração, faz-se as seguintes correções das variáveis do problema:

$$\boldsymbol{\sigma}_{n+1}^{(k+1)} = \boldsymbol{\sigma}_{n+1}^{(k)} + \Delta \boldsymbol{\sigma}_{n+1}^{(k)} \quad (3.47)$$

$$\boldsymbol{q}_{n+1}^{(k+1)} = \boldsymbol{q}_{n+1}^{(k)} + \Delta \boldsymbol{q}_{n+1}^{(k)} \quad (3.48)$$

$$\Delta \boldsymbol{\gamma}_{i,n+1}^{(k+1)} = \Delta \boldsymbol{\gamma}_{i,n+1}^{(k)} + \Delta^2 \boldsymbol{\gamma}_{i,n+1}, \quad \forall i \in \mathbb{J}_{act}^{(k+1)} \quad (3.49)$$

As atualizações (3.47), (3.48) e (3.49) devem ser feitas até o momento que o resíduo (3.20) se torna desprezível. Dessa forma, é possível atualizar tensões, variáveis internas e multiplicador plástico. Para atualizar as deformações plásticas a cada iteração, faz-se:

$$\boldsymbol{\epsilon}_{n+1}^{p(k+1)} = \boldsymbol{\epsilon}_{n+1}^{p(k)} - \boldsymbol{C}_{n+1}^{(k)} \Delta \boldsymbol{\sigma}_{n+1}^{(k)} \quad (3.50)$$

Apresenta-se na Figura 3.2 uma interpretação geométrica para o algoritmo de projeção ao ponto mais próximo, considerando elasticidade linear, plasticidade ideal ($\boldsymbol{h}_i(\boldsymbol{\sigma}, \boldsymbol{q}) = 0$) e associada ($\boldsymbol{n}_i(\boldsymbol{\sigma}, \boldsymbol{q}) = \partial_{\boldsymbol{\sigma}} f(\boldsymbol{\sigma}, \boldsymbol{q})$). De fato, o que o algoritmo minimiza é a distância entre $\boldsymbol{\sigma}_{n+1}^{trial}$ e $\boldsymbol{\sigma}_{n+1}$, considerando a métrica definida pelo tensor constitutivo elástico. Observa-se que, a cada iteração, a solução atual retorna a uma superfície hipotética. Tal retorno é feito de forma perpendicular à mesma

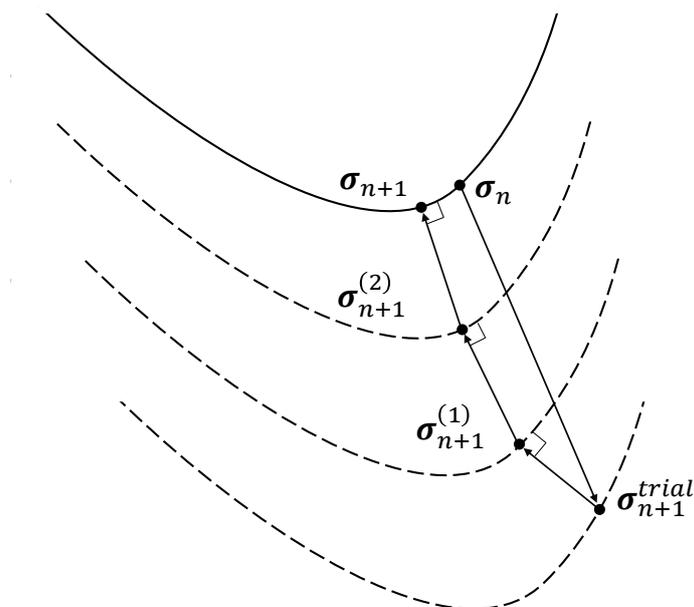


Figura 3.2 – Interpretação geométrica do algoritmo de projeção ao ponto mais próximo

3.1.2. Tensor Constitutivo Elastoplástico Consistente

Para conservar a convergência quadrática da solução global de equilíbrio no método dos elementos finitos, é necessário adotar um tensor constitutivo elastoplástico consistente, diferente do tensor constitutivo elastoplástico infinitesimal (2.41). O cálculo de um tensor constitutivo elastoplástico consistente foi apresentado inicialmente por Bathe *et al.* (1984, *apud* Huang e Griffiths, 2008). Objetiva-se calcular \mathbf{D}_{n+1}^{ep} , que representa:

$$\mathbf{D}_{n+1}^{ep} = \frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} \quad (3.51)$$

A abordagem empregada neste trabalho é similar ao apresentado por Foster *et al.* (2005) e Souza Neto, Perić e Owen (2008) que abordam o caso de superfície de plastificação única. Para o caso de múltiplas superfícies de plastificação, tem-se Lourenço, de Borst e Rots (1997) e Karoulanis (2013) que empregam abordagens similares.

Obtém-se \mathbf{D}_{n+1}^{ep} a partir do sistema de equações (3.14). Aplica-se a diferenciação nesse, sendo necessário empregar a regra da cadeia para obter as seguintes equações:

$$\begin{aligned}
d\boldsymbol{\varepsilon}_{n+1}^p &= \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \partial_{\boldsymbol{\sigma}} \mathbf{n}_{i,n+1} : d\boldsymbol{\sigma}_{n+1} + \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \partial_{\mathbf{q}} \mathbf{n}_{i,n+1} : d\mathbf{q}_{n+1} \\
&\quad + \sum_{i \in \mathbb{J}_{act}} d\Delta\gamma_{i,n+1} \mathbf{n}_{i,n+1}
\end{aligned} \tag{3.52}$$

$$\begin{aligned}
d\mathbf{q}_{n+1} &= \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \partial_{\boldsymbol{\sigma}} \mathbf{h}_{i,n+1} : d\boldsymbol{\sigma}_{n+1} + \sum_{i \in \mathbb{J}_{act}} \Delta\gamma_{i,n+1} \partial_{\mathbf{q}} \mathbf{h}_{i,n+1} : d\mathbf{q}_{n+1} \\
&\quad + \sum_{i \in \mathbb{J}_{act}} d\Delta\gamma_{i,n+1} \mathbf{h}_{i,n+1}
\end{aligned} \tag{3.53}$$

$$\partial_{\boldsymbol{\sigma}} f_{i,n+1} : d\boldsymbol{\sigma}_{n+1} + \partial_{\mathbf{q}} f_{i,n+1} : d\mathbf{q}_{n+1} = 0, \forall i \in \mathbb{J}_{act} \tag{3.54}$$

Sendo $d\boldsymbol{\varepsilon}_{n+1}^p = d\boldsymbol{\varepsilon}_{n+1} - d\boldsymbol{\varepsilon}_{n+1}^e$, é possível pôr o sistema formado pelas equações (3.52) a (3.54) na seguinte forma matricial:

$$\mathbf{J}_{n+1} \begin{bmatrix} d\boldsymbol{\sigma}_{n+1} \\ d\mathbf{q}_{n+1} \\ d\Delta\boldsymbol{\gamma}_{n+1} \end{bmatrix} = \begin{bmatrix} d\boldsymbol{\varepsilon}_{n+1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.55}$$

em que:

$$\mathbf{J}_{n+1} = \begin{bmatrix} \mathbf{C}_{n+1} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1} \partial_{\boldsymbol{\sigma}} \mathbf{n}_{i,n+1} & \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1} \partial_{\mathbf{q}} \mathbf{n}_{i,n+1} & \mathbf{N}_{n+1} \\ \left(\sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1} \partial_{\boldsymbol{\sigma}} \mathbf{h}_{i,n+1} \right)^T & -\mathbf{I} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta\gamma_{i,n+1} \partial_{\mathbf{q}} \mathbf{h}_{i,n+1} & \mathbf{H}_{n+1} \\ (\partial_{\boldsymbol{\sigma}} \mathbf{f}_{n+1})^T & (\partial_{\mathbf{q}} \mathbf{f}_{n+1})^T & \mathbf{0} \end{bmatrix} \tag{3.56}$$

considerando que \mathbf{J}_{n+1} é a matriz Jacobiana de (3.20), também empregada no método de Newton-Raphson para calcular as correções das tensões, variáveis internas e multiplicador plástico. Dessa forma:

$$\begin{bmatrix} d\boldsymbol{\sigma}_{n+1} \\ d\mathbf{q}_{n+1} \\ d\Delta\boldsymbol{\gamma}_{n+1} \end{bmatrix} = (\mathbf{J}_{n+1})^{-1} \begin{bmatrix} d\boldsymbol{\varepsilon}_{n+1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{3.57}$$

A obtenção do tensor constitutivo elastoplástico consistente parte da inversão da matriz \mathbf{J}_{n+1} . Dessa forma, define-se:

$$(\mathbf{J}_{n+1})^{-1} = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{D}_{13} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \mathbf{D}_{23} \\ \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} \end{bmatrix} \quad (3.58)$$

Assim, o sistema (3.57) se torna:

$$\begin{bmatrix} d\boldsymbol{\sigma}_{n+1} \\ d\mathbf{q}_{n+1} \\ d\Delta\boldsymbol{\gamma}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{D}_{13} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \mathbf{D}_{23} \\ \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} \end{bmatrix} \begin{bmatrix} d\boldsymbol{\varepsilon}_{n+1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (3.59)$$

Por fim, é possível obter o tensor constitutivo elastoplástico consistente por meio de (3.59), dado por:

$$\mathbf{D}_{n+1}^{ep} = \frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \mathbf{D}_{11} \quad (3.60)$$

Em resumo, deve-se inverter a matriz Jacobiana \mathbf{J}_{n+1} e isolar a matriz \mathbf{D}_{11} . A ordem dessa última matriz é igual ao número de componentes de tensão empregados na análise, conforme a notação de Voigt. Uma alternativa para obter \mathbf{D}_{n+1}^{ep} é inverter a matriz Jacobiana completa, depois selecionar a parcela de interesse. Todavia, como uma parte desta matriz já foi invertida ($\boldsymbol{\Psi}_{n+1}^{(k)}$), é interessante aproveitá-la para reduzir o volume de cálculos. Portanto, propõem-se, neste trabalho, empregar a forma de Banachiewicz-Schur para inversão de matrizes em bloco, conforme apresentado por Baksalary e Styan (2002), para uma dada matriz \mathbf{M}_0 :

$$\mathbf{M}_0 = \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 \\ \mathbf{C}_0 & \mathbf{D}_0 \end{bmatrix} \quad (3.61)$$

A inversa desta matriz é dada por:

$$\mathbf{M}_0^{-1} = \begin{bmatrix} \mathbf{A}_0^{-1} + \mathbf{A}_0^{-1}\mathbf{B}_0\mathbf{S}_0^{-1}\mathbf{C}_0\mathbf{A}_0^{-1} & \mathbf{A}_0^{-1}\mathbf{B}_0\mathbf{S}_0^{-1} \\ \mathbf{S}_0^{-1}\mathbf{C}_0\mathbf{A}_0^{-1} & \mathbf{S}_0^{-1} \end{bmatrix} \quad (3.62)$$

em que:

$$\mathbf{S}_0 = \mathbf{D}_0 - \mathbf{C}_0\mathbf{A}_0^{-1}\mathbf{B}_0 \quad (3.63)$$

Define-se então, baseado na matriz de coeficientes do sistema (3.38), que:

$$\mathbf{A}_0 = (\boldsymbol{\Psi}_{n+1})^{-1} \quad (3.64)$$

$$\mathbf{B}_0 = \mathbf{E}_{n+1} \quad (3.65)$$

$$\mathbf{C}_0 = (\mathbf{G}_{n+1})^T \quad (3.66)$$

$$\mathbf{D}_0 = \mathbf{0} \quad (3.67)$$

$$\mathbf{S}_0 = -(\mathbf{G}_{n+1})^T\boldsymbol{\Psi}_{n+1}\mathbf{E}_{n+1} = -(\boldsymbol{\Lambda}_{n+1})^{-1} \quad (3.68)$$

Assim, o primeiro termo da matriz \mathbf{M}_0^{-1} é dado por:

$$\begin{aligned} \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix} &= \mathbf{A}_0^{-1} + \mathbf{A}_0^{-1}\mathbf{B}_0\mathbf{S}_0^{-1}\mathbf{C}_0\mathbf{A}_0^{-1} = \mathbf{A}_0^{-1}(\mathbf{I} + \mathbf{B}_0\mathbf{S}_0^{-1}\mathbf{C}_0\mathbf{A}_0^{-1}) \\ &= \boldsymbol{\Psi}_{n+1}(\mathbf{I} - \mathbf{E}_{n+1}\boldsymbol{\Lambda}_{n+1}(\mathbf{G}_{n+1})^T\boldsymbol{\Psi}_{n+1}) \end{aligned} \quad (3.69)$$

Para obter apenas o termo \mathbf{D}_{11} e, conseqüentemente, o tensor constitutivo elastoplástico \mathbf{D}_{n+1}^{ep} , uma forma elegante e proposta neste trabalho é:

$$\mathbf{D}_{n+1}^{ep} = \boldsymbol{\Gamma}^T\boldsymbol{\Psi}_{n+1}(\mathbf{I} - \mathbf{E}_{n+1}\boldsymbol{\Lambda}_{n+1}(\mathbf{G}_{n+1})^T\boldsymbol{\Psi}_{n+1})\boldsymbol{\Gamma} \quad (3.70)$$

em que $\boldsymbol{\Gamma}$ é uma matriz identidade de ordem $A \times B$, sendo A igual ao número de componentes de tensão somado ao número de variáveis internas e B igual ao número de componentes de tensão. Ou seja, $\boldsymbol{\Gamma}$ tem a forma:

$$\mathbf{\Gamma} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad (3.71)$$

Dessa forma, tem-se uma forma eficiente de calcular o tensor constitutivo elastoplástico consistente. Este cálculo pode ser imediatamente realizado ao fim do algoritmo de mapeamento de retorno, já que todas as variáveis necessárias para o cálculo do tensor já estão devidamente definidas, inclusive aproveitando as inversões de matrizes previamente realizadas.

3.2. Abordagens para Aperfeiçoamento do Algoritmo de Projeção ao Ponto mais Próximo

Alguns autores citam dificuldades de convergência enfrentados na atualização de tensões realizadas por meio do algoritmo de projeção ao ponto mais próximo. Huang e Griffiths (2008) relatam que as dificuldades de convergências são comuns tanto para modelos complexos, que consideram endurecimento e amolecimento do material, quanto para modelos perfeitamente plásticos. Os autores experimentam tais problemas para o critério de Mohr-Coulomb.

Uma abordagem disponível na literatura para aumentar a taxa de sucesso do algoritmo de projeção ao ponto mais próximo é a busca unidimensional. Exemplos de aplicação, para problemas de plasticidade com superfície de plastificação única, são apresentados por Dutko, Perić e Owen (1993), Pérez-Foguet e Armero (2002), Seifert e Schmidt (2008), Li e Crouch (2010) e Lester e Scherzinger (2017).

Scherzinger (2017) reporta que o algoritmo de projeção ao ponto mais próximo sem busca unidimensional pode conduzir a situações de ciclos infinitos de iterações. Para solucionar tal problema, os autores incluem a busca unidimensional na implementação desse algoritmo, aumentando a taxa de sucesso do mesmo.

Adhikary *et al.* (2017) adotam a busca unidimensional em um algoritmo de mapeamento de retorno para múltiplas superfícies de plastificação. Todavia a função de resíduos empregada é distinta da empregada neste trabalho.

Outro problema que pode dificultar a convergência do algoritmo, observado por Godio *et al.* (2016), é a possibilidade da matriz Jacobiana (3.36) ser singular devido à dependência linear entre as direções de fluxo. Isso pode ocorrer devido redundância ou coincidência de superfícies de plastificação.

Redundância e coincidência de superfícies de plastificação não são problemas usuais, todavia observou-se, durante a implementação do *Cap Model*, a dependência linear entre a superfície do *cap* e do *cut-off*, pois os gradientes destas se tornam paralelos quando se aplica carregamento puramente hidrostático. Além disso, o endurecimento do material conduziu ao mal-condicionamento da matriz (3.36) e, conseqüentemente, a problemas no procedimento usual de inversão de matrizes.

Assim, apresenta-se a seguir duas metodologias empregadas no algoritmo de projeção ao ponto mais próximo para contornar eventuais problemas de convergência, descritos anteriormente, e aumentar a taxa de sucesso do algoritmo.

3.2.1. Inversão de Matriz via Decomposição em Valores Singulares

Apresenta-se, nesta seção, a inversão de matrizes a partir da decomposição em valores singulares, que permite o cálculo de pseudoinversa de matrizes retangulares. O algoritmo apresentado aqui é baseado em Press *et al.* (2007) e Godio *et al.* (2016). O método consiste em fatorar uma matriz retangular $A \times B$ genérica A em um produto de três matrizes S , U e V , conforme apresentado a seguir:

$$A = USV^T \quad (3.72)$$

em que U é uma matriz ortogonal de ordem $A \times A$, cujas colunas são formadas pelo autovetores de AA^T , V é uma matriz ortogonal de ordem $B \times B$, cujas colunas são formadas pelo autovetores de $A^T A$, e por fim, S é uma matriz diagonal positiva definida de ordem $A \times B$, cujos elementos S_{ii} correspondem às raízes quadradas dos autovalores não nulos de $A^T A$. Portanto, a inversa de A é dada por:

$$A^{-1} = VS^{-1}U^T \quad (3.73)$$

Quando \mathbf{A} é não singular, expressão (3.73) resulta na matriz calculada por metodologias tradicionais. Todavia, se \mathbf{A} é singular ou mal-condicionada, alguns dos elementos de \mathbf{S} são nulos ou aproximadamente nulos, sendo impossível aplicar diretamente à expressão (3.73). Assim, deve-se modificar a matriz \mathbf{S}^{-1} para $\bar{\mathbf{S}}^{-1}$. Sendo cada elemento da diagonal principal de $\bar{\mathbf{S}}^{-1}$ representado por \bar{S}_{ii}^{-1} , tem-se:

$$\bar{S}_{ii}^{-1} = \begin{cases} 0, & \text{se } S_{ii} < tol \\ \frac{1}{S_{ii}}, & \text{caso contrário} \end{cases} \quad (3.74)$$

Sendo \mathbf{A}^+ a pseudoinversa de \mathbf{A} , que representa a melhor aproximação para \mathbf{A}^{-1} , essa pode ser calculada por meio da expressão (3.75):

$$\mathbf{A}^+ = \mathbf{V}\bar{\mathbf{S}}^{-1}\mathbf{U}^T \quad (3.75)$$

O quadro resumo a seguir apresenta o algoritmo para obter a pseudoinversa de uma matriz retangular.

Quadro 3.1 – Cálculo da pseudoinversa da matriz \mathbf{A} a partir da decomposição em valores singulares

- 1) Dado de entrada: \mathbf{A} , tol
- 2) Computar a matriz \mathbf{U} , cujas colunas são definidas pelos autovetores do produto $\mathbf{A}\mathbf{A}^T$
- 3) Computar a matriz \mathbf{V} , cujas colunas são definidas pelos autovetores do produto $\mathbf{A}^T\mathbf{A}$
- 4) Computar a matriz diagonal \mathbf{S} , cujos elementos da diagonal principal S_{ii} correspondem às raízes quadradas dos autovalores de $\mathbf{A}^T\mathbf{A}$
- 5) Calcular a matriz diagonal $\bar{\mathbf{S}}^{-1}$, cujos elementos da diagonal principal \bar{S}_{ii}^{-1} são calculados por:

$$\bar{S}_{ii}^{-1} = \begin{cases} 0, & \text{se } S_{ii} < tol \\ \frac{1}{S_{ii}}, & \text{caso contrário} \end{cases}$$

- 6) Calcular e retornar a pseudoinversa de \mathbf{A} , dada por:

$$\mathbf{A}^+ = \mathbf{V}\bar{\mathbf{S}}^{-1}\mathbf{U}^T$$

Na aplicação apresentada neste trabalho, a matriz \mathbf{A} é sempre quadrada. Desse modo, as inversões de matrizes, necessárias nas equações (3.39) e (3.46), são realizadas a partir desta técnica, mesmo que tais matrizes não sejam singulares.

3.2.2.

Método de Newton-Raphson como Algoritmo de Otimização

Uma aplicação para algoritmos de otimização é a solução de sistemas não lineares. Dessa forma, é interessante aplicar tal abordagem à solução do sistema indicado por (3.19). Isso pode ser feito definindo a seguinte função objetivo $\psi(\mathbf{x}_{n+1})$ para o problema de otimização sem restrições, com base no vetor de resíduos $\mathbf{r}_{n+1}(\mathbf{x}_{n+1}) = \mathbf{r}_{n+1}(\boldsymbol{\sigma}_{n+1}, \mathbf{q}_{n+1}, \Delta\boldsymbol{\gamma}_{n+1})$, conforme (3.20):

$$\psi(\mathbf{x}_{n+1}) = \frac{1}{2} (\mathbf{r}_{n+1}(\mathbf{x}_{n+1}))^T \mathbf{r}_{n+1}(\mathbf{x}_{n+1}) \quad (3.76)$$

Considerando o problema de plasticidade para múltiplas superfícies de plastificação, o vetor \mathbf{x}_{n+1} é dado por:

$$\mathbf{x}_{n+1} = \begin{bmatrix} \boldsymbol{\sigma}_{n+1} \\ \mathbf{q}_{n+1} \\ \Delta\boldsymbol{\gamma}_{n+1} \end{bmatrix} \quad (3.77)$$

Sendo assim, define-se o problema de otimização sem restrições a ser resolvido como:

$$\min_{\mathbf{x}_{n+1}} \psi(\mathbf{x}_{n+1}) \quad (3.78)$$

Existe uma grande variedade de algoritmos que pode ser empregado para solucionar o problema (3.78). Podem ser citados o método do máximo declive, método de Fletcher-Reeves e métodos Quase-Newton. O método de Newton-Raphson também pode ser empregado nesse tipo de problema e será adotado neste trabalho, aproveitando a formulação apresentada nessa seção. Uma descrição desses métodos encontra-se em livros textos, como por exemplo, Nocedal e Wright (2006), Rao (2009) e Arora (2012).

Na otimização irrestrita, uma das abordagens empregadas é a combinação de métodos de busca unidimensional (em inglês, *line search*) e de direção de busca (em inglês, *descent direction*). Outra abordagem para esse tipo de problema é a do método de região de confiança (em inglês, *trust region*). Esta última abordagem é empregada por Lester e Scherzinger (2017) para solucionar problemas de plasticidade envolvendo uma superfície de plastificação. Os autores realizaram testes aplicando a busca unidimensional e a Região de Confiança em um problema de uma haste submetida a esforços de tração e cisalhamento, considerando o modelo elastoplástico proposto por Hosfold (1972, *apud* Lester e Scherzinger, 2017). A performance dos dois algoritmos foi bastante similar, considerando o tempo de processamento.

Sendo o método de Newton-Raphson um método de direção de busca, define-se a direção de descida $\mathbf{d}_{n+1}^{(k)}$, que representa a direção que se deve tomar a partir de um ponto $\mathbf{x}_{n+1}^{(k)}$ da hipersuperfície $\psi_{n+1} = \psi(\mathbf{x}_{n+1})$ para se obter uma solução mais próxima da solução ótima do problema. Obtida a partir de uma expansão por série de Taylor, a direção de busca é:

$$\mathbf{d}_{n+1}^{(k)} = -\left(\mathcal{H}_{n+1}^{(k)}\right)^{-1} \partial_x \psi_{n+1}^{(k)} \quad (3.79)$$

considerando que $\mathcal{H}_{n+1}^{(k)}$ representa a matriz Hessiana de $\psi_{n+1}^{(k)}$. Tendo em vista a equação (3.76) e a matriz Jacobiana $\mathcal{J}_{n+1}^{(k)}$ de $\psi_{n+1}^{(k)}$, tem-se que:

$$\partial_x \psi_{n+1}^{(k)} = \partial_x \psi(\mathbf{x}_{n+1}^{(k)}) = \left(\mathcal{J}_{n+1}^{(k)}\right)^T \mathbf{r}_{n+1}^{(k)} \quad (3.80)$$

Ademais, a matriz $\mathcal{H}_{n+1}^{(k)}$ é dada por:

$$\mathcal{H}_{n+1}^{(k)} = \left(\mathcal{J}_{n+1}^{(k)}\right)^T \mathcal{J}_{n+1}^{(k)} + \sum_i \mathbf{r}_{i,n+1}^{(k)} \partial_{xx}^2 \mathbf{r}_{i,n+1}^{(k)} \quad (3.81)$$

Segundo Nocedal e Wright (2006), a segunda parcela da expressão (3.81) pode ser considerada desprezível, sobretudo quando se aproxima da solução do

problema. Dessa forma, a parcela $\left(\mathcal{J}_{n+1}^{(k)}\right)^T \mathcal{J}_{n+1}^{(k)}$ tem a maior contribuição no cálculo da matriz Hessiana e por isso, adota-se, assim como é feito em diversos algoritmos de mínimos quadrados não lineares, a seguinte aproximação:

$$\mathcal{H}_{n+1}^{(k)} \approx \left(\mathcal{J}_{n+1}^{(k)}\right)^T \mathcal{J}_{n+1}^{(k)} \quad (3.82)$$

Dessa forma, a equação (3.79), com base em (3.80) e (3.82), torna-se

$$\mathbf{d}_{n+1}^{(k)} = -\left(\left(\mathcal{J}_{n+1}^{(k)}\right)^T \mathcal{J}_{n+1}^{(k)}\right)^{-1} \left(\mathcal{J}_{n+1}^{(k)}\right)^T \mathbf{r}_{n+1}^{(k)} = -\left(\mathcal{J}_{n+1}^{(k)}\right)^{-1} \mathbf{r}_{n+1}^{(k)} \quad (3.83)$$

Portanto, a equação (3.83) recai na solução do seguinte sistema de equações para obtenção da direção de descida $\mathbf{d}_{n+1}^{(k)}$:

$$\mathcal{J}_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)} = -\mathbf{r}_{n+1}^{(k)} \quad (3.84)$$

Observa-se que a equação (3.84) é idêntica à equação (3.35). Dessa forma, os incrementos de tensão, deformação plástica e multiplicador plástico representam fisicamente as componentes da direção de descida da função ψ_{n+1} em direção à solução do problema de plasticidade. Isto posto, esta direção é dada por:

$$\mathbf{d}_{n+1}^{(k)} = \begin{bmatrix} \Delta \sigma_{n+1}^{(k)} \\ \Delta \mathbf{q}_{n+1}^{(k)} \\ \Delta^2 \gamma_{n+1}^{(k)} \end{bmatrix} \quad (3.85)$$

Sendo assim, resta calcular:

$$\mathbf{x}_{n+1}^{(k+1)} = \mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)} \quad (3.86)$$

O tamanho do passo $\alpha_{n+1}^{(k)}$ é calculado a partir de alguma estratégia de busca unidimensional. Dados $\mathbf{x}_{n+1}^{(k)}$ e $\mathbf{d}_{n+1}^{(k)}$, isso é feito solucionando o seguinte problema:

$$\min_{\alpha_{n+1}^{(k)} > 0} \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)}) \quad (3.87)$$

O procedimento apresentado nesta seção recai no formulado na Seção 3.1.1 quando $\alpha_{n+1}^{(k)} = 1$, já que a atualização de cada uma das variáveis é feita pelos incrementos $\Delta \sigma_{n+1}^{(k)}$, $\Delta \mathbf{q}_{n+1}^{(k)}$ e $\Delta^2 \boldsymbol{\gamma}_{n+1}^{(k)}$ de forma integral. Quando se considera a busca unidimensional, apenas uma fração do vetor $\mathbf{d}_{n+1}^{(k)}$ é empregada para atualizar as variáveis do vetor $\mathbf{x}_{n+1}^{(k)}$, pois é usual adotar $0 < \alpha_{n+1}^{(k)} \leq 1$ para o método de Newton-Raphson, conforme apresentado por autores como Nocedal e Wright (2006), Press *et al.* (2007) e Scherzinger (2017).

Dentre as metodologias disponíveis para cálculo de $\alpha_{n+1}^{(k)}$, existem as exatas e inexatas. Neste trabalho, avaliam-se três métodos:

- Seção áurea (busca exata);
- Interpolação quadrática (busca inexata);
- Interpolação cúbica (busca inexata).

Neste trabalho, são apresentados os algoritmos dessas três metodologias, que atuam em conjunto com a regra de Armijo, que também é um método de busca unidimensional inexato, que indica se um valor de $\alpha_{n+1}^{(k)}$ é razoável para a iteração atual. Esta regra é descrita a seguir.

3.2.2.1. Regra de Armijo

Por comodidade, define-se:

$$\omega(\alpha_{n+1}^{(k)}) = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)}) \quad (3.88)$$

A regra de Armijo estabelece de forma inexata se o tamanho do passo dado decrementa suficientemente a função a ser minimizada. Considera-se o critério definido pela equações (3.89), conforme descrito por Armijo (1966):

$$\psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)}) \leq \psi(\mathbf{x}_{n+1}^{(k)}) + c_1 \alpha_{n+1}^{(k)} \left(\partial_x \psi(\mathbf{x}_{n+1}^{(k)}) \right)^T \mathbf{d}_{n+1}^{(k)} \quad (3.89)$$

em que c_1 é um parâmetro de controle da inclinação de busca, sugerido por diversos autores, como Nocedal e Wright (2006), com o valor de $c_1 = 10^{-4}$. A expressão pode também ser apresentada da seguinte forma, para o caso especial da função objetivo descrita pela equação (3.76):

$$\omega(\alpha_{n+1}^{(k)}) \leq \psi(\mathbf{x}_{n+1}^{(k)}) - 2c_1 \alpha_{n+1}^{(k)} \psi(\mathbf{x}_{n+1}^{(k)}) \quad (3.90)$$

A expressão definida em (3.90) tem a interpretação geométrica representada na Figura 3.3. As regiões denotadas como aceitável definem valores suficientes para $\alpha_{n+1}^{(k)}$. É importante ter um algoritmo que defina um valor adequado para $\alpha_{n+1}^{(k)}$, já que a equação (3.90) é atendida para pequenos valores deste parâmetro.

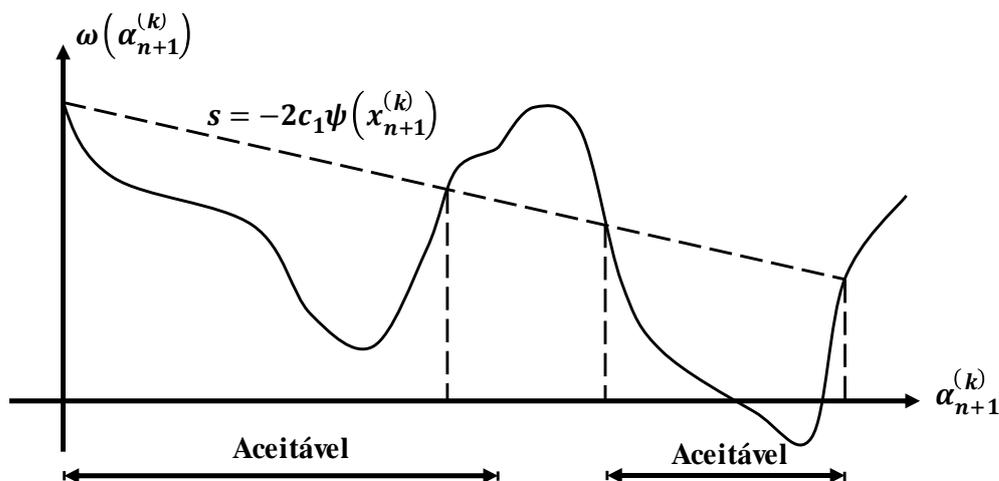


Figura 3.3 – Interpretação geométrica da regra de Armijo

Assim, o algoritmo inicia verificando se o tamanho do passo $\alpha_{n+1}^{(k)} = 1$ satisfaz a equação (3.89), ou seja, se é possível dar o passo máximo indicado. Caso não seja possível, uma das três abordagens a seguir pode ser adotada para o cálculo do valor ótimo para $\alpha_{n+1}^{(k)}$. Na opção por interpolação quadrática ou cúbica, a regra de Armijo também será empregada para avaliar se a solução em cada iteração é adequada. O Quadro 3.2 resume os passos necessários para aplicar esta regra.

Quadro 3.2 – Regra de Armijo

<p>1) Dados de entrada: $c_1, \mathbf{x}_{n+1}^{(k)}, \mathbf{d}_{n+1}^{(k)}, \alpha_{n+1}^{(k)}$</p> <p>2) Definição da função objetivo:</p> $\psi(\mathbf{x}_{n+1}^{(k)}) = \frac{1}{2} \left(\mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)}) \right)^T \mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)})$ $\mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)}) = \mathbf{r}_{n+1}(\boldsymbol{\sigma}_{n+1}^{(k)}, \mathbf{q}_{n+1}^{(k)}, \Delta \boldsymbol{\gamma}_{n+1}^{(k)}), \text{ conforme a equação (3.20)}$ $\omega(\alpha_{n+1}^{(k)}) = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)})$ <p>3) Calcular a inclinação de busca ao longo da direção $\mathbf{d}_{n+1}^{(k)}$:</p> $s = -2c_1 \psi(\mathbf{x}_{n+1}^{(k)})$ <p>4) Verificar regra de Armijo:</p> <p>Se $\omega(\alpha_{n+1}^{(k)}) \leq \psi(\mathbf{x}_{n+1}^{(k)}) + s\alpha_{n+1}^{(k)}$:</p> <p style="padding-left: 40px;">Retornar que a condição foi verificada</p> <p>Caso contrário:</p> <p style="padding-left: 40px;">Retornar que a condição não foi verificada</p>
--

3.2.2.2. Método da Seção Áurea

Neste ponto, considera-se que a solução do problema (3.87) está no intervalo $0 < \alpha_{n+1}^{(k)} \leq 1$. O método da seção áurea é um método de busca exata intervalar, que particiona esse intervalo múltiplas vezes até que se chega a um intervalo dentro de uma dada tolerância tol , podendo qualquer um dos limites ser considerado a solução do problema.

Uma abordagem usual é adotar subintervalos iguais. Para determinar qual dos dois intervalos descartar, ou seja, qual dos dois intervalos não contém a solução ótima, adota-se a média desses. Será descartado aquele intervalo que possuir o maior valor da função (3.88) avaliada no ponto médio. A estratégia de subdivisão continua no intervalo remanescente, até que o intervalo seja tão pequeno quanto uma tolerância definida. Esta metodologia é chamada de método da biseção. Uma forma mais eficiente de subdividir os intervalos é dividi-los com base na proporção áurea, com base na constante φ , definida por:

$$\varphi = \frac{\sqrt{5} - 1}{2} \quad (3.91)$$

Desse modo, a primeira divisão de intervalos gerará o subintervalo $[0, \varphi]$ e $[\varphi, 1]$. Assim como no método da biseção, aplica-se a mesma proporção φ para determinar a posição a ser usada para determinar qual intervalo será descartado. Independente do intervalo descartado, uma destas duas últimas posições permanecerá, devido à proporção áurea, proporcionando maior eficiência computacional, pois avaliações da função (3.88) podem ser aproveitadas de uma iteração para outra. A interpretação gráfica do método é apresentada na Figura 3.4. Com isso em vista, o algoritmo apresentado nos Quadros 3.3a e 3.3b resumem o método da seção áurea, que é empregado em conjunto com a regra de Armijo.

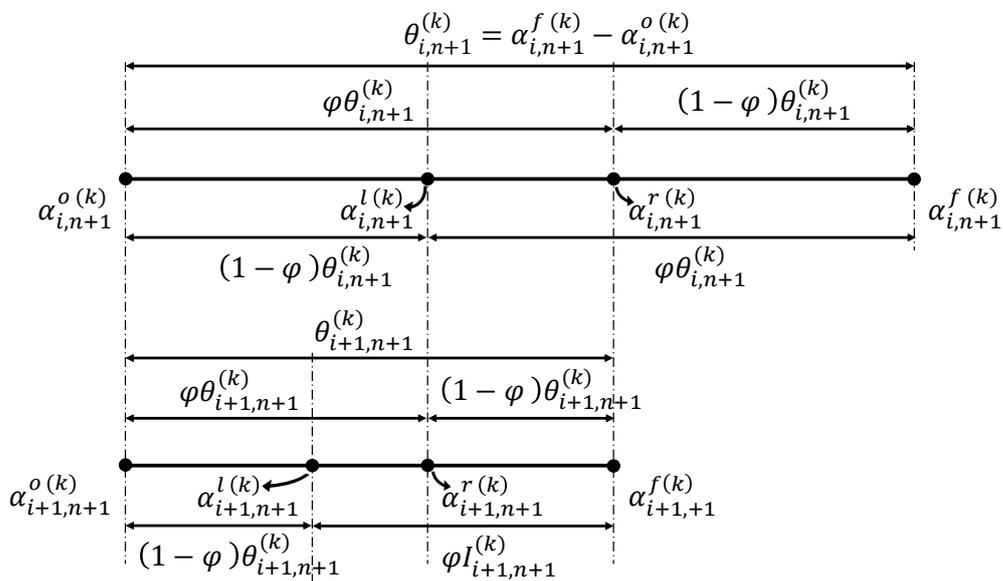


Figura 3.4 – Interpretação gráfica do método da seção áurea

Quadro 3.3a – Busca unidimensional via método da seção áurea (parte 1)

- 1) Dados de entrada: $\mathbf{x}_{n+1}^{(k)}$, $\mathbf{d}_{n+1}^{(k)}$, tol
 - 2) Inicializar: $i = 1$, $\alpha_{0,n+1}^{o(k)} = 0$, $\alpha_{0,n+1}^{f(k)} = 1$
 - 3) Definições iniciais:

$$\psi(\mathbf{x}_{n+1}^{(k)}) = \frac{1}{2} \left(\mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)}) \right)^T \mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)})$$

$$\mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)}) = \mathbf{r}_{n+1}(\boldsymbol{\sigma}_{n+1}^{(k)}, \mathbf{q}_{n+1}^{(k)}, \Delta \boldsymbol{\gamma}_{n+1}^{(k)}), \text{ conforme a equação (3.20)}$$

$$\partial_{\alpha} \omega(0) = -2\psi(\mathbf{x}_{n+1}^{(k)})$$

$$\varphi = \frac{\sqrt{5} - 1}{2}$$
- Ir para o Quadro 3.3b

Quadro 3.3b – Busca unidimensional via método da seção áurea (parte 2)

- 4) Verificar regra de Armijo para $\alpha_{n+1}^{(k)} = \alpha_{0,n+1}^{f(k)}$, conforme o Quadro 3.2.

Se a condição for verificada:

Retornar $\alpha_{n+1}^{(k)} = \alpha_{0,n+1}^{(k)}$ e encerrar o algoritmo

- 5) Primeira subdivisão do intervalo:

$$\alpha_{n+1}^{r(k)} = \varphi \alpha_{0,n+1}^{f(k)}$$

$$\alpha_{n+1}^{l(k)} = (1 - \varphi) \alpha_{0,n+1}^{f(k)}$$

$$\psi_{n+1}^{r(k)} = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{r(k)} \mathbf{d}_{n+1}^{(k)})$$

$$\psi_{n+1}^{l(k)} = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{l(k)} \mathbf{d}_{n+1}^{(k)})$$

- 6) Definição do intervalo que contém a solução:

Se $\psi_{n+1}^{r(k)} > \psi_{n+1}^{l(k)}$:

$$\alpha_{i,n+1}^{f(k)} = \alpha_{i,n+1}^{r(k)}$$

$$\theta_{i,n+1}^{(k)} = \alpha_{i,n+1}^{f(k)} - \alpha_{i,n+1}^{o(k)}$$

$$\alpha_{i,n+1}^{r(k)} = \alpha_{i,n+1}^{l(k)}$$

$$\alpha_{i,n+1}^{l(k)} = \alpha_{i,n+1}^{o(k)} + (1 - \varphi) \theta_{i,n+1}^{(k)}$$

$$\psi_{n+1}^{r(k)} = \psi_{n+1}^{l(k)}$$

$$\psi_{n+1}^{l(k)} = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{l(k)} \mathbf{d}_{n+1}^{(k)})$$

Caso contrário:

$$\alpha_{i,n+1}^{o(k)} = \alpha_{0,n+1}^{l(k)}$$

$$\theta_{i,n+1}^{(k)} = \alpha_{i,n+1}^{f(k)} - \alpha_{i,n+1}^{o(k)}$$

$$\alpha_{i,n+1}^{l(k)} = \alpha_{i,n+1}^{r(k)}$$

$$\alpha_{i,n+1}^{r(k)} = \alpha_{i,n+1}^{o(k)} + \varphi \theta_{i,n+1}^{(k)}$$

$$\psi_{n+1}^{l(k)} = \psi_{n+1}^{r(k)}$$

$$\psi_{n+1}^{r(k)} = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{r(k)} \mathbf{d}_{n+1}^{(k)})$$

- 7) Verificar o critério de parada:

Se $\theta_{i,n+1}^{(k)} < tol$:

$$\alpha_{n+1}^{(k)} = \frac{\alpha_{i,n+1}^{o(k)} + \alpha_{i,n+1}^{f(k)}}{2}$$

Retornar $\alpha_{n+1}^{(k)}$ e encerrar o algoritmo

Caso contrário:

Fazer $i := i + 1$

Retornar ao passo 6)

3.2.2.3. Interpolação Quadrática

A interpolação quadrática é um método de busca inexata intervalar. Autores que empregam esta abordagem são Pérez-Foguet e Armero (2002), Seifert e Schmidt (2008) e Scherzinger (2017). Neste método, parte-se do conhecimento prévio de três dados para construir uma aproximação parabólica da função a ser minimizada. Estes dados são: $\psi(\mathbf{x}_{n+1}^{(k)})$, $\partial_\alpha \omega(\alpha_{0,n+1}^{(k)})$ e $\psi(\mathbf{x}_{n+1}^{(k)} + \mathbf{d}_{n+1}^{(k)})$. Sendo assim, segundo Pérez-Foguet e Armero (2002), tem-se que:

$$\begin{aligned} \partial_\alpha \omega(\alpha_{n+1}^{(k)}) &= \left(\partial_x \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)}) \right)^T \mathbf{d}_{n+1}^{(k)} \\ &= -2\psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \mathbf{d}_{n+1}^{(k)}) \end{aligned} \quad (3.92)$$

A cada interpolação quadrática i realizada obtém-se um valor de $\alpha_{i,n+1}^{(k)}$. Com isso, deve-se verificar a regra de Armijo (3.89). Caso esta não seja atendida, faz-se uma nova interpolação considerando os seguintes dados: $\psi(\mathbf{x}_{n+1}^{(k)})$, $\partial_\alpha \omega(\alpha_{0,n+1}^{(k)})$ e $\psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{i,n+1}^{(k)} \mathbf{d}_{n+1}^{(k)})$. Se $\alpha_{0,n+1}^{(k)} = 1$, de acordo com Nocedal e Wright (2006), a interpolação de tais dados é feita por meio da seguinte expressão:

$$\alpha_{i,n+1}^{(k)} = \frac{\partial_\alpha \omega(0)(\alpha_{i-1,n+1}^{(k)})^2}{2 \left(\psi(\mathbf{x}_{n+1}^{(k)}) + \partial_\alpha \omega(0) \alpha_{i-1,n+1}^{(k)} - \omega(\alpha_{i-1,n+1}^{(k)}) \right)} \quad (3.93)$$

Conforme sugerido por Seifert e Schmidt (2008), adota-se:

$$0,1\alpha_{i,n+1}^{(k)} \leq \alpha_{i,n+1}^{(k)} \leq 0,5\alpha_{i,n+1}^{(k)} \quad (3.94)$$

a fim de que não sejam obtidos valores muito pequenos ou muito elevados para $\alpha_{i,n+1}^{(k)}$. Neste trabalho, para a primeira interpolação quadrática, considera-se o limite $\alpha_{i,n+1}^{(k)} < 1$, para que valores próximos da unidade possam vir a ser aceitos em um primeiro instante. O processo é encerrado quando a regra de Armijo for

atendida ou quando um número máximo de iterações for alcançado, conforme sugerido por Pérez-Foguet e Armero (2002).

3.2.2.4. Interpolação Cúbica

A interpolação cúbica é um método de busca inexata intervalar. Autores que empregam esta abordagem são Seifert e Schmidt (2008) e Adhikary *et al.* (2017). O método inicia realizando uma interpolação quadrática, conforme apresentado anteriormente, obtendo-se $\alpha_{i,n+1}^{(k)}$ caso a regra de Armijo não seja atendida. Assim, tem-se informações suficientes para obter uma interpolação cúbica para o problema, a partir de: $\psi(\mathbf{x}_{n+1}^{(k)})$, $\partial_\alpha \omega(\alpha_{0,n+1}^{(k)})$ e $\psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{i,n+1}^{(k)} \mathbf{d}_{n+1}^{(k)})$ e $\psi(\mathbf{x}_{n+1}^{(k)} + \mathbf{d}_{n+1}^{(k)})$.

A cada nova interpolação cúbica, verifica-se a regra de Armijo. Caso esta não seja atendida, novas interpolações são realizadas, mantendo como informação $\psi(\mathbf{x}_{n+1}^{(k)})$ e $\partial_\alpha \omega(\alpha_{0,n+1}^{(k)})$, substituindo sempre os dois últimos valores obtidos para $\alpha_{i,n+1}^{(k)}$. Segundo Nocedal e Wright (2006), a interpolação cúbica pode ser realizada a partir da seguinte expressão:

$$\alpha_{i,n+1}^{(k)} = \frac{-b + \sqrt{b^2 - 3a\partial_\alpha \omega(0)}}{3a} \quad (3.95)$$

considerando que:

$$\mu = \frac{1}{\left(\alpha_{i-2,n+1}^{(k)} \alpha_{i-1,n+1}^{(k)}\right)^2 \left(\alpha_{i-1,n+1}^{(k)} - \alpha_{i-2,n+1}^{(k)}\right)} \quad (3.96)$$

$$\Phi = \begin{bmatrix} \left(\alpha_{i-2,n+1}^{(k)}\right)^2 & -\left(\alpha_{i-1,n+1}^{(k)}\right)^2 \\ -\left(\alpha_{i-2,n+1}^{(k)}\right)^3 & \left(\alpha_{i-1,n+1}^{(k)}\right)^3 \end{bmatrix} \quad (3.97)$$

$$\vartheta = \begin{bmatrix} \omega\left(\alpha_{i-1,n+1}^{(k)}\right) - \psi\left(\mathbf{x}_{n+1}^{(k)}\right) - \partial_\alpha \omega(0) \alpha_{i-1,n+1}^{(k)} \\ \omega\left(\alpha_{i-2,n+1}^{(k)}\right) - \psi\left(\mathbf{x}_{n+1}^{(k)}\right) - \partial_\alpha \omega(0) \alpha_{i-2,n+1}^{(k)} \end{bmatrix} \quad (3.98)$$

$$\zeta = \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \mu \Phi \vartheta \quad (3.99)$$

As recomendações atribuídas à interpolação quadrática também serão seguidas na interpolação cúbica. Devido à similaridade das duas estratégias, foram elaborados os Quadros 3.4a a 3.4c, que contemplam o algoritmo para as duas metodologias de interpolação. Isto posto, define-se que *quad* indica uma interpolação quadrática e *cubic*, uma interpolação cúbica.

Durante a implementação desta interpolação, verificou-se a possibilidade de $b^2 < 3a\partial_\alpha\omega(0)$, o que impossibilita o cálculo de $\sqrt{b^2 - 3a\partial_\alpha\omega(0)}$. Dessa forma, propõe-se que o cálculo de $\alpha_{n+1}^{(k)}$ seja interrompido e seja retornado $\alpha_{i-1,n+1}^{(k)}$.

Quadro 3.4a – Busca unidimensional via interpolação quadrática ou cúbica (parte 1)

<p>1) Dados de entrada: $\mathbf{x}_{n+1}^{(k)}$, $\mathbf{d}_{n+1}^{(k)}$, <i>max</i>, <i>type</i></p> <p>2) Inicializar: $i = 1$, $\alpha_{0,n+1}^{(k)} = 1$</p> <p>3) Definições iniciais:</p> $\psi(\mathbf{x}_{n+1}^{(k)}) = \frac{1}{2} \left(\mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)}) \right)^T \mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)})$ $\mathbf{r}_{n+1}(\mathbf{x}_{n+1}^{(k)}) = \mathbf{r}_{n+1}(\boldsymbol{\sigma}_{n+1}^{(k)}, \mathbf{q}_{n+1}^{(k)}, \Delta\mathbf{y}_{n+1}^{(k)}), \text{ conforme a equação (3.20)}$ $\partial_\alpha\omega(0) = -2\psi(\mathbf{x}_{n+1}^{(k)})$ <p>4) Verificar regra de Armijo para $\alpha_{n+1}^{(k)} = \alpha_{0,n+1}^{(k)}$, conforme o Quadro 3.2: Se a condição for verificada: Retornar $\alpha_{n+1}^{(k)} = \alpha_{0,n+1}^{(k)}$ e encerrar o algoritmo</p> <p>5) Realizar a interpolação quadrática inicial:</p> $\omega(\alpha_{0,n+1}^{(k)}) = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{0,n+1}^{(k)} \mathbf{d}_{n+1}^{(k)})$ $\alpha_{1,n+1}^{(k)} = \frac{\psi(\mathbf{x}_{n+1}^{(k)})}{\psi(\mathbf{x}_{n+1}^{(k)}) + \omega(\alpha_{0,n+1}^{(k)})}$ <p>6) Verificar os limites de $\alpha_{1,n+1}^{(k)}$:</p> <p>Se $\alpha_{1,n+1}^{(k)} < 0,1\alpha_{1,n+1}^{(k)}$:</p> $\alpha_{1,n+1}^{(k)} = 0,1\alpha_{1,n+1}^{(k)}$ <p>Se $\alpha_{1,n+1}^{(k)} > 1$:</p> $\alpha_{1,n+1}^{(k)} = 1$ $\omega(\alpha_{1,n+1}^{(k)}) = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{1,n+1}^{(k)} \mathbf{d}_{n+1}^{(k)})$ <p>Ir para o Quadro 3.4b</p>
--

Quadro 3.4b – Busca unidimensional via interpolação quadrática ou cúbica (parte 2)

7) Verificar regra de Armijo para $\alpha_{n+1}^{(k)} = \alpha_{i,n+1}^{(k)}$, conforme o Quadro 3.2:

Se a condição for verificada:

Retornar $\alpha_{n+1}^{(k)} = \alpha_{i,n+1}^{(k)}$ e encerrar o algoritmo

Fazer $i := i + 1$

8) Realizar uma nova interpolação:

Se $type = quad$:

$$\alpha_{i,n+1}^{(k)} = \frac{\partial_{\alpha}\omega(0)\left(\alpha_{i-1,n+1}^{(k)}\right)^2}{2\left(\psi\left(\mathbf{x}_{n+1}^{(k)}\right) + \partial_{\alpha}\omega(0)\alpha_{i-1,n+1}^{(k)} - \omega\left(\alpha_{i-1,n+1}^{(k)}\right)\right)}$$

Se $type = cubic$:

$$\mu = \frac{1}{\left(\alpha_{i-2,n+1}^{(k)}\alpha_{i-1,n+1}^{(k)}\right)^2\left(\alpha_{i-1,n+1}^{(k)} - \alpha_{i-2,n+1}^{(k)}\right)}$$

$$\Phi = \begin{bmatrix} \left(\alpha_{i-2,n+1}^{(k)}\right)^2 & -\left(\alpha_{i-1,n+1}^{(k)}\right)^2 \\ -\left(\alpha_{i-2,n+1}^{(k)}\right)^3 & \left(\alpha_{i-1,n+1}^{(k)}\right)^3 \end{bmatrix}$$

$$\mathfrak{g} = \begin{bmatrix} \omega\left(\alpha_{i-1,n+1}^{(k)}\right) - \psi\left(\mathbf{x}_{n+1}^{(k)}\right) - \partial_{\alpha}\omega(0)\alpha_{i-1,n+1}^{(k)} \\ \omega\left(\alpha_{i-2,n+1}^{(k)}\right) - \psi\left(\mathbf{x}_{n+1}^{(k)}\right) - \partial_{\alpha}\omega(0)\alpha_{i-2,n+1}^{(k)} \end{bmatrix}$$

$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \mu\Phi\mathfrak{g}$$

Se $b^2 < 3a\partial_{\alpha}\omega(0)$:

Retornar $\alpha_{n+1}^{(k)} = \alpha_{i-1,n+1}^{(k)}$ e encerrar o algoritmo

$$\alpha_{i,n+1}^{(k)} = \frac{-\zeta_2 + \sqrt{\zeta_2^2 - 3\zeta_1\partial_{\alpha}\omega(0)}}{3\zeta_1}$$

9) Verificar os limites de $\alpha_{i,n+1}^{(k)}$:

Se $\alpha_{i,n+1}^{(k)} < 0,1\alpha_{i-1,n+1}^{(k)}$:

$$\alpha_{i,n+1}^{(k)} = 0,1\alpha_{i-1,n+1}^{(k)}$$

Se $\alpha_{i,n+1}^{(k)} > 0,5\alpha_{i-1,n+1}^{(k)}$:

$$\alpha_{i,n+1}^{(k)} = 0,5\alpha_{i-1,n+1}^{(k)}$$

Ir para o Quadro 3.4c

Quadro 3.4c – Busca unidimensional via interpolação quadrática ou cúbica (parte 3)

10) Verificar o limite de iterações:

Se $i < max$:

Retornar ao passo 7) do Quadro 3.4b

$$\omega(\alpha_{i,n+1}^{(k)}) = \psi(\mathbf{x}_{n+1}^{(k)} + \alpha_{i,n+1}^{(k)} \mathbf{d}_{n+1}^{(k)})$$

Caso contrário:

Retornar $\alpha_{n+1}^{(k)} = \alpha_{i,n+1}^{(k)}$ e encerrar o algoritmo

3.2.2.5.

Algoritmo de Projeção ao Ponto mais Próximo com Busca Unidimensional

Aqui são feitas algumas observações acerca do algoritmo originalmente apresentado por Simo, Kennedy e Govindjee (1988), referente ao algoritmo de projeção ao ponto mais próximo para múltiplas superfícies. Esses autores consideram que um grupo inicial de superfícies ativas $\mathbb{J}_{act}^{(k)}$ pode ter apenas superfícies removidas desse conjunto. Todavia experimentos numéricos desenvolvidos ao longo deste trabalho identificaram que pode ser necessário redefinir o conjunto $\mathbb{J}_{act}^{(k)}$ a cada correção das variáveis.

Outro problema observado no algoritmo original é a possibilidade de, durante o processo iterativo, todas as superfícies de plastificação se tornem desativadas. Sendo ainda necessário corrigir o estado de tensões, sugere-se que se reative as superfícies de plastificação correspondentes ao estado de tensões predictoras.

Por fim, observou-se que é possível que o algoritmo calcule um valor de multiplicador plástico para uma superfície que ao final do processo iterativo não está ativa. Isso pode ocorrer devido à dependência linear entre as direções dos fluxos plásticos, reportada por Adhikary *et al.* (2017).

Quando se analisa o *Cap Model*, pode-se observar a dependência linear entre as direções dos fluxos plásticos entre o *cut-off* de tração e o *cap* elíptico, conforme representado na Figura 3.5. Pode-se assumir que o *cap* elíptico está sempre desativado sempre que $I_1 > L(\kappa)$, o que impediria tal problema surgir. Todavia um algoritmo robusto deve ser capaz de solucionar esse tipo problema.

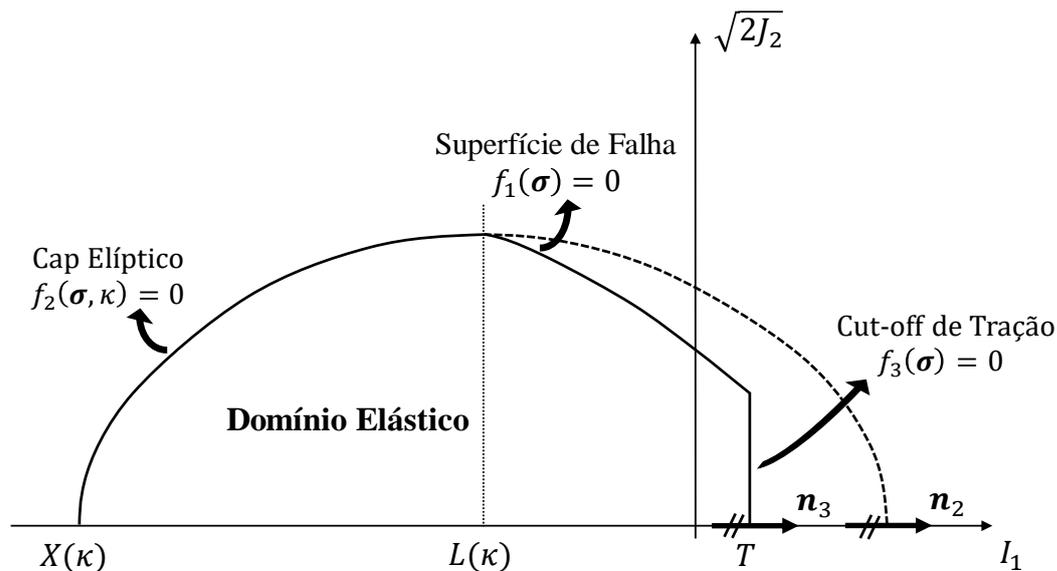


Figura 3.5 – Dependência linear entre as direções dos fluxos plásticos

Observa-se, na Figura 3.5, que as direções \mathbf{n}_2 e \mathbf{n}_3 são paralelas para um estado de tensões de tração ao longo do eixo I_1 (tensão desviadora nula). Claramente as tensões atualizadas devem estar de acordo com o *cut-off* de tração. Todavia, para um estado de tensões que viole o *cut-off* de tração e o *cap* elíptico, o algoritmo de projeção ao ponto mais próximo calculará um valor de multiplicador plástico para o *cap* elíptico, o que não faz sentido. Sugere-se que o algoritmo retorne apenas o valor de multiplicador plástico para as superfícies ativas ao final do processo iterativo. Além do mais, para a situação demonstrada na figura em questão, é necessário verificar constantemente a iteração em que o estado de tensões será corrigido para dentro ou sobre o *cap* elíptico, para tornar esta superfície desativada.

Dessa forma, o algoritmo de projeção ao ponto mais próximo empregado neste trabalho, para atender problemas com múltiplas superfícies de plastificação, é apresentado nos Quadros 3.5a ao 3.5c. Para tal, considera-se três tolerâncias: tol_1 , tol_2 e tol_3 . Cada uma corresponde a um elemento do vetor \mathbf{r}_{n+1} .

Frisa-se que as inversões de matrizes necessárias no Quadro 3.5b são feitas a partir do cálculo de pseudoinversa, com base no Quadro 3.1. A busca unidimensional foi incluída no algoritmo, como apresentado no Quadro 3.5c. Sugere-se selecionar um dos tipos de busca unidimensional apresentadas neste trabalho, nos Quadros 3.3a e 3.3b, ou Quadro 3.4a a 3.4b. Quando se considera

$\alpha_{n+1}^{(k)} = 1$, o algoritmo basicamente se torna o apresentado no trabalho de Simo, Kennedy e Govindjee (1988), com algumas ligeiras diferenças.

Quadro 3.5a – Algoritmo de projeção ao ponto mais próximo com busca unidimensional para múltiplas superfícies de plastificação (parte 1)

<p>1) Dados de entrada: $\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{\varepsilon}_n^p$, \mathbf{q}_n, $W(\boldsymbol{\varepsilon}^e)$, $f_i(\boldsymbol{\sigma}, \mathbf{q})$, $\mathbf{n}_i(\boldsymbol{\sigma}, \mathbf{q})$, $\mathbf{h}_i(\boldsymbol{\sigma}, \mathbf{q})$, m, tol_1, tol_2, tol_3</p> <p>2) Preditor elástico:</p> $\boldsymbol{\sigma}_{n+1}^{trial} = \nabla W(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n^p)$ $\mathbf{f}_{n+1}^{trial} = \begin{bmatrix} f_{1,n+1}^{trial} \\ f_{2,n+1}^{trial} \\ \vdots \\ f_{m,n+1}^{trial} \end{bmatrix}$ <p>3) Definir as superfícies de plastificação ativas para o preditor elástico:</p> $\mathbb{J}_{act}^{trial} = \{i \in \{1, 2, \dots, m\} \mid f_{i,n+1}^{trial} > 0\}$ <p>4) Checar a condição de plastificação:</p> <p>Se $\mathbb{J}_{act}^{trial} = \emptyset$:</p> $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{trial}, \mathbf{q}_{n+1} = \mathbf{q}_n, \Delta\boldsymbol{\gamma}_{n+1} = \mathbf{0}, \boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_n^p$ <p>Retornar $\boldsymbol{\sigma}_{n+1}$, \mathbf{q}_{n+1}, $\Delta\boldsymbol{\gamma}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}^p$ e encerrar o algoritmo</p> <p>Caso contrário:</p> $k = 0$ $\boldsymbol{\sigma}_{n+1}^{(k)} = \boldsymbol{\sigma}_{n+1}^{trial}, \mathbf{q}_{n+1}^{(k)} = \mathbf{q}_n, \Delta\boldsymbol{\gamma}_{n+1}^{(k)} = \mathbf{0}, \boldsymbol{\varepsilon}_{n+1}^{p(k)} = \boldsymbol{\varepsilon}_n^p$ $\mathbb{J}_{act}^{(k)} = \mathbb{J}_{act}^{trial}$ $\mathbf{f}_{n+1}^{(k)} = \begin{bmatrix} f_{i_1,n+1}^{trial} \\ f_{i_2,n+1}^{trial} \\ \vdots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)}$ <p>5) Avaliar as matrizes $\mathbf{N}_{n+1}^{(k)}$, $\mathbf{H}_{n+1}^{(k)}$ e $\mathbf{E}_{n+1}^{(k)}$:</p> $\mathbf{N}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{n}_{i_1,n+1}^{(k)} & \mathbf{n}_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)}$ $\mathbf{H}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{h}_{i_1,n+1}^{(k)} & \mathbf{h}_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)}$ $\mathbf{E}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{N}_{n+1}^{(k)} \\ \mathbf{H}_{n+1}^{(k)} \end{bmatrix}$ <p>Ir para o Quadro 3.5b</p>

Quadro 3.5b – Algoritmo de projeção ao ponto mais próximo com busca unidimensional para múltiplas superfícies de plastificação (parte 2)

6) Verificar o critério de parada:

$$\boldsymbol{\rho}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{r}_{1,n+1}^{(k)} \\ \mathbf{r}_{2,n+1}^{(k)} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\varepsilon}_{n+1}^p + \boldsymbol{\varepsilon}_n^p + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \mathbf{n}_{i,n+1}^{(k)} \\ -\mathbf{q}_{n+1}^{(k)} + \mathbf{q}_n + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \mathbf{h}_{i,n+1}^{(k)} \end{bmatrix}$$

Se $f_{i,n+1}^{(k)} \leq tol_1$, $\forall i \in \{1,2, \dots, m\}$, $\|\mathbf{r}_{1,n+1}^{(k)}\| \leq tol_2$ e

$$\|\mathbf{r}_{2,n+1}^{(k)}\| \leq tol_3:$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n+1}^{(k)}, \mathbf{q}_{n+1} = \mathbf{q}_{n+1}^{(k)}, \boldsymbol{\varepsilon}_{n+1}^p = \boldsymbol{\varepsilon}_{n+1}^{p(k)}$$

$$\Delta \gamma_{i,n+1} = \Delta \gamma_{i,n+1}^{(k)}, \forall i \in \mathbb{J}_{act}^{(k)}$$

$$\Delta \gamma_{i,n+1} = 0, \text{ tal que } \{i \in \{1,2, \dots, m\} \mid i \notin \mathbb{J}_{act}^{(k)}\}$$

Retornar $\boldsymbol{\sigma}_{n+1}$, \mathbf{q}_{n+1} , $\Delta \boldsymbol{\gamma}_{n+1}$, $\boldsymbol{\varepsilon}_{n+1}^p$ e encerrar o algoritmo

7) Obter o incremento do multiplicador plástico:

$$\mathbf{f}_{n+1}^{(k)} = \begin{bmatrix} f_{i_1,n+1}^{(k)} \\ f_{i_2,n+1}^{(k)} \\ \vdots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)}$$

$$\partial_{\boldsymbol{\sigma}} \mathbf{f}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{\sigma}} f_{i_1,n+1}^{(k)} & \partial_{\boldsymbol{\sigma}} f_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)}$$

$$\partial_{\mathbf{q}} \mathbf{f}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\mathbf{q}} f_{i_1,n+1}^{(k)} & \partial_{\mathbf{q}} f_{i_2,n+1}^{(k)} & \dots \end{bmatrix}, \text{ tal que } i_1, i_2, \dots \in \mathbb{J}_{act}^{(k)}$$

$$\mathbf{D}_{n+1}^{(k)} = \nabla^2 W(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^p), \mathbf{C}_{n+1}^{(k)} = (\mathbf{D}_{n+1}^{(k)})^{-1}$$

$$\boldsymbol{\Psi}_{n+1}^{(k)} = \begin{bmatrix} \mathbf{C}_{n+1}^{(k)} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{\sigma}} \mathbf{n}_{i,n+1}^{(k)} & \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_{\mathbf{q}} \mathbf{n}_{i,n+1}^{(k)} \\ \left(\sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_{\boldsymbol{\sigma}} \mathbf{h}_{i,n+1}^{(k)} \right)^T & -\mathbf{I} + \sum_{i \in \mathbb{J}_{act}^{(k)}} \Delta \gamma_{i,n+1}^{(k)} \partial_{\mathbf{q}} \mathbf{h}_{i,n+1}^{(k)} \end{bmatrix}^{-1}$$

$$\mathbf{G}_{n+1}^{(k)} = \begin{bmatrix} \partial_{\boldsymbol{\sigma}} \mathbf{f}_{n+1}^{(k)} \\ \partial_{\mathbf{q}} \mathbf{f}_{n+1}^{(k)} \end{bmatrix}$$

$$\boldsymbol{\Lambda}_{n+1}^{(k)} = \left((\mathbf{G}_{n+1}^{(k)})^T \boldsymbol{\Psi}_{n+1}^{(k)} \mathbf{E}_{n+1}^{(k)} \right)^{-1}$$

$$\Delta^2 \boldsymbol{\gamma}_{n+1}^{(k)} = \boldsymbol{\Lambda}_{n+1}^{(k)} \left(\mathbf{f}_{n+1}^{(k)} - (\mathbf{G}_{n+1}^{(k)})^T \boldsymbol{\Psi}_{n+1}^{(k)} \boldsymbol{\rho}_{n+1}^{(k)} \right)$$

Calcular $\boldsymbol{\Psi}_{n+1}^{(k)}$ e $\boldsymbol{\Lambda}_{n+1}^{(k)}$ por meio da metodologia exposta no Quadro 3.1

Ir para o Quadro 3.5c

Quadro 3.5c – Algoritmo de projeção ao ponto mais próximo com busca unidimensional para múltiplas superfícies de plastificação (parte 3)

8) Verificar condição de Karush-Kuhn-Tucker:

$$\Delta \bar{\gamma}_{i,n+1}^{(k+1)} = \Delta \gamma_{i,n+1}^{(k)} + \Delta^2 \gamma_{i,n+1}, \quad \forall i \in \mathbb{J}_{act}^{(k)}$$

Se $\Delta \bar{\gamma}_{i,n+1}^{(k+1)} \geq 0, \quad \forall i \in \{1, 2, \dots, m\}$:

$$\Delta \boldsymbol{\gamma}_{n+1}^{(k+1)} = \Delta \bar{\boldsymbol{\gamma}}_{n+1}^{(k+1)}$$

Caso contrário:

$$\mathbb{J}_{act}^{(k)} = \{i \in \mathbb{J}_{act}^{(k)} \mid \Delta \bar{\gamma}_{i,n+1}^{(k+1)} > 0\}$$

Ir para o passo 14)

9) Calcular incremento de tensões e variáveis internas:

$$\begin{bmatrix} \Delta \boldsymbol{\sigma}_{n+1}^{(k)} \\ \Delta \boldsymbol{q}_{n+1}^{(k)} \end{bmatrix} = -\boldsymbol{\Psi}_{n+1}^{(k)} \left(\boldsymbol{\rho}_{n+1}^{(k)} + \mathbf{E}_{n+1}^{(k)} \Delta^2 \boldsymbol{\gamma}_{n+1}^{(k)} \right)$$

10) Calcular o valor de $\alpha_{n+1}^{(k)}$ por meio da busca unidimensional (Quadro 3.3a e Quadro 3.3b, ou Quadro 3.4a e Quadro 3.4b)

11) Atualizar as tensões, variáveis internas e multiplicador plástico:

$$\boldsymbol{\sigma}_{n+1}^{(k+1)} = \boldsymbol{\sigma}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \Delta \boldsymbol{\sigma}_{n+1}^{(k)}$$

$$\boldsymbol{q}_{n+1}^{(k+1)} = \boldsymbol{q}_{n+1}^{(k)} + \alpha_{n+1}^{(k)} \Delta \boldsymbol{q}_{n+1}^{(k)}$$

$$\Delta \gamma_{i,n+1}^{(k+1)} = \Delta \gamma_{i,n+1}^{(k)} + \alpha_{n+1}^{(k)} \Delta^2 \gamma_{i,n+1}, \quad \forall i \in \mathbb{J}_{act}^{(k)}$$

12) Atualizar as deformações plásticas:

$$\boldsymbol{\varepsilon}_{n+1}^{p(k+1)} = \boldsymbol{\varepsilon}_{n+1}^{p(k)} - \alpha_{n+1}^{(k)} \mathbf{C}_{n+1}^{(k)} \Delta \boldsymbol{\sigma}_{n+1}^{(k)}$$

Fazer $k := k + 1$

13) Definir as superfícies de plastificação que permanecem ativas:

$$\boldsymbol{f}_{n+1}^{(k)} = \begin{bmatrix} f_{1,n+1}^{(k)} \\ f_{2,n+1}^{(k)} \\ \vdots \\ f_{m,n+1}^{(k)} \end{bmatrix}$$

$$\mathbb{J}_{act}^{(k)} = \{i \in \{1, 2, \dots, m\} \mid f_{i,n+1}^{(k)} > -tol_1\}$$

14) Reativar as superfícies de plastificação referentes ao preditor elástico, caso todas as superfícies se desativem:

Se $\mathbb{J}_{act}^{(k)} = \emptyset$:

$$\mathbb{J}_{act}^{(k)} = \mathbb{J}_{act}^{trial}$$

Retornar ao passo 5) do Quadro 3.5a

4 Análise Inversa

Nas últimas décadas, uma grande quantidade de processos de modelagem em engenharia vem sendo desenvolvidos para solucionar problemas diretos, ou seja, aqueles em que se conhece as causas e se deseja calcular os efeitos. Como tais processos requerem o conhecimento prévio de determinados parâmetros, por vezes é necessário resolver um problema inverso, ou seja, aquele para o qual se conhecem os resultados e se deseja calcular as causas. Para estudar esse tipo de problema diversos autores estudam metodologias dentro do que se chama análise inversa.

Em geral, a determinação de parâmetros de uma relação constitutiva é feita por intermédio de ensaios de laboratório de pequenas amostras ou testes realizados em campo. Também é comum calibrar modelos computacionais com os resultados obtidos em campo. Tal processo muitas vezes é conduzido mediante a tentativa e erro. Inúmeras análises são necessárias até que os resultados produzidos pelas simulações sejam condizentes com a realidade.

À medida que a modelagem se torna mais complexa, por vezes, a abordagem da tentativa e erro torna-se inviável. Dessa forma, é cada vez mais necessário desenvolver técnicas racionais e automáticas para realizar esse tipo de processo. Nesse âmbito, surgem as técnicas da análise inversa, cujo benefício fundamental é automatizar a obtenção de parâmetros que melhor ajustem os resultados previstos em relação aos observados. Como benefício adicional, reduz-se o tempo necessário para calibrar tais parâmetros em relação ao processo da tentativa e erro.

A análise inversa é um tipo de análise classicamente conhecida por ser um problema mal-posto. De acordo com Tikhonov e Arsenin (1977), não existe a garantia de existência e unicidade da solução, além da instabilidade desta, quando pequenas alterações nas condições iniciais e de contorno são aplicadas. Contudo a análise inversa se torna uma poderosa ferramenta para substituir o processo de tentativa e erro.

Bem difundida em problemas geotécnicos e geomecânicos, a análise inversa teve seus primeiros trabalhos abordando estas áreas nos anos 1980 (Gioda e Maier,

1980; Cividini, Jurina e Gioda, 1981; Arai, Ohta e Yasui, 1983). Na literatura, tal metodologia é aplicada a diversos problemas de engenharia, como escavações profundas (Calvello e Finno, 2004; Finno e Calvello, 2005; Tang e Kung, 2009, Zhao *et al.*, 2015), escavações de túneis (Ledesma, Gens e Alonso, 1996; Gens, Ledesma e Alonso, 1996; Swoboda *et al.*, 1999; Miranda *et al.* 2011; Vardakos, Gutierrez e Xia, 2012), análise de aterros em solos moles (Arai *et al.*, 1986), análise de barragens (Jia e Chi, 2015) e estimativa de parâmetros geomecânicos (Zhao e Yin, 2009; Song, Jiang e Jiang, 2015).

No contexto de modelos constitutivos, uma aplicação interessante para a análise inversa é obter os parâmetros de um determinado modelo, a partir de efeitos previamente conhecidos. Como vários desses parâmetros não têm um significado físico, torna-se difícil desenvolver ensaios para obtê-los. Alguns trabalhos relacionados são desenvolvidos por Pal, Wathugala e Kundu (1996), Zentar, Hicher e Moulin, (2001) Samarajiva, Macari e Wathugala (2005), Levasseur *et al.* (2008) e Yin *et al.* (2016).

Ju *et al.* (1985) calibram os parâmetros do *Cap Model* por meio de ensaios em corpos de prova de concreto. Tais ensaios são realizados a partir de trajetórias multiaxiais de tensões não convencionais, sendo necessário um algoritmo adequado para a calibração. Assim sendo, os autores adotam o método de Marquardt-Levenberg para a obtenção dos parâmetros do *Cap Model*. Simo e Ju (1987b) desenvolvem uma calibração similar, todavia considerando o *Cap Model* com uma lei de dano.

Melo (1995) desenvolve uma implementação do modelo elastoplástico proposto por Kim e Lade (1988, apud Melo, 1995) a fim de representar o comportamento tensão-deformação de solos em um programa de elementos finitos. Tal implementação é aplicada na análise da fase de construção de uma barragem. Inicialmente, os parâmetros desse modelo são obtidos experimentalmente, por intermédio de um procedimento convencional. Essa primeira estimativa não produziu resultados adequados, com respeito ao comportamento do material. Dessa forma, o autor um algoritmo evolucionário para determinar tais parâmetros. O autor conclui que os parâmetros obtidos através dessa abordagem representam melhor o problema real.

Assim, será apresentado como aplicar algoritmos de otimização para obter os parâmetros de modelos constitutivos elastoplásticos. A metodologia não é

particularizada para esse tipo de problema, podendo ser empregada de uma forma geral, para problemas de ajuste de curvas.

4.1. Calibração de Parâmetros

A análise inversa pode ser empregada em diversas aplicações, como determinar condições de contorno de um modelo matemático. Todavia o objetivo deste trabalho é determinar parâmetros de modelos por meio desta metodologia.

Calibrar adequadamente um modelo matemático a partir de um conjunto de dados que representam as respostas ou efeitos previamente conhecidos consiste em minimizar a diferença entre tais dados e os resultados gerados pelo modelo, produzindo um grupo de parâmetros que representa de forma consistente o fenômeno analisado. Dentre as metodologias disponíveis para calibração de parâmetros, a análise inversa é preferida, pois é um processo objetivo, mesmo quando não existe um significado físico para os parâmetros (Yin *et al.*, 2017). Além disso, os algoritmos de análise inversa permitem a calibração de múltiplos parâmetros simultaneamente.

Na análise inversa, é usual calibrar-se um determinado modelo por meio de um processo iterativo. Dependendo do método empregado, é necessário estimar parâmetros iniciais ou um intervalo que deve conter tal solução. Esta estimativa pode ser feita recorrendo à experiência, ao conhecimento prévio ou às informações disponíveis na literatura.

4.2. Solução de Problemas Inversos

Em problemas diretos, os dados de entrada e os parâmetros são previamente conhecidos. Em uma análise linear elástica via elementos finitos, por exemplo, tem-se os parâmetros do material: módulo de elasticidade e o coeficiente de Poisson. Os dados de entrada são geometria, condições de contorno, malha de elementos finitos, entre outros. O principal objetivo da análise é obter respostas (efeitos), como deslocamentos e tensões. Em um problema inverso, objetiva-se obter parâmetros desconhecidos para um conjunto conhecido de dados de entradas e respostas. Ambos os problemas são representados na Figura 4.1.

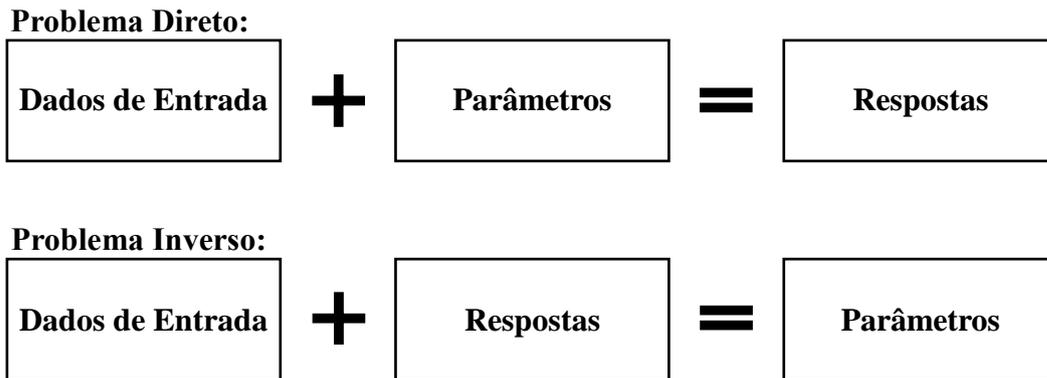


Figura 4.1 – Esquema de solução de problemas diretos e inversos

Existem duas abordagens para a solução de problemas inversos: a inversa e a direta. Na abordagem inversa, invertem-se as equações que regem o problema com respeito aos parâmetros desconhecidos que se deseja obter. Já na abordagem direta, empregam-se processos iterativos a fim de minimizar uma função erro total médio, sem que seja necessária inversão das equações do problema. Esse último caminho é mais adequado para modelos complexos e não-lineares, sobretudo quando não existem soluções fechadas para o problema (Knabe *et al.*, 2013). A metodologia apresentada neste trabalho segue a abordagem direta, já que os problemas tratados são demasiadamente complexos para aplicação da abordagem inversa.

4.3. Abordagem Direta para Solução de Problemas Inversos

Com o crescente avanço das técnicas de otimização nas últimas décadas, a abordagem direta na análise inversa ganhou grande relevância na engenharia. Baseado no exposto por Knabe *et al.* (2013), para realizar uma calibração de parâmetros de forma direta via algoritmos de otimização, os seguintes passos devem ser seguidos:

- Definir conjunto(s) de dados a ser(em) comparado(s) com os valores previstos pelo modelo matemático analisado. É usual que esses dados sejam experimentais, mas nada impede que dados sintéticos sejam empregados;
- Definir os dados de entrada do problema, como as condições de contorno e condições iniciais do modelo;

- Dependendo do algoritmo para solução do problema de otimização, é necessário definir uma solução inicial ou um intervalo que deve conter tal solução. Problemas mais complexos podem envolver (outras) restrições;
- Definir uma função objetivo que indicará um ajuste consistente entre os dados observados e os obtidos pelo modelo matemático;
- Selecionar e executar o algoritmo de otimização mais adequado;
- Avaliar a qualidade e confiança dos parâmetros obtidos. Dependendo das circunstâncias pode ser necessário refazer a execução do algoritmo selecionado.

4.3.1. Problema de Otimização

Na abordagem direta, os problemas a serem tratados neste trabalho podem ser formulados como problemas de otimização irrestritos, ou seja:

$$\min_{\mathbf{p}} E^t(\mathbf{p}) \quad (4.1)$$

em que \mathbf{p} são os parâmetros a serem calibrados. De acordo com o algoritmo de otimização empregado, pode ser necessário incluir restrições laterais na formulação do problema, que definem um domínio Ω . Os algoritmos adotados neste trabalho possuem esta natureza, de modo que o problema de otimização a ser resolvido tem o seguinte formato:

$$\min_{\mathbf{p} \in \Omega} E^t(\mathbf{p}) \quad (4.2)$$

Para um problema de análise inversa, a função objetivo deve avaliar o erro entre as medições observadas e as correspondentes previsões. Quando existe mais de uma medição de erro associada ao problema, existem duas abordagens que podem ser adotadas: otimização mono-objetivo ou multiobjetivo. A abordagem multiobjetivo é detalhada por Papon *et al.* (2012) e Jiang *et al.* (2018). Neste trabalho, a abordagem mono-objetivo é adotada, baseada nos trabalhos de Ye *et al.* (2016) e Yin *et al.* (2017). Esta abordagem combina as medidas de erro a partir de pesos para formar a função objetivo, como descrito a seguir:

$$E^t(\mathbf{p}) = \sum_{i=1}^m w_i E_i(\mathbf{p}) \quad (4.3)$$

considerando que w_i se refere aos pesos para cada medição de erro $E_i(\mathbf{p})$, respeitando $\sum w_i = 1$, m é o número de conjuntos de dados considerados e $E^t(\mathbf{p})$ é a medida de erro para cada conjunto de dados.

Para contornar problemas de escala que podem ocorrer devido a diferenças entre ordens de grandeza dos erros mensurados, propõe-se normalizar os dados empregados no cálculo dos erros. Dessa forma, cada resposta do modelo matemático e cada elemento do conjunto de dados empregados no ajuste, representados genericamente por y_i , são normalizados mediante aos máximos e mínimos valores dos dados observados, para cada conjunto de dados, de acordo com o que se segue:

$$a_i = \max(\mathbf{y}_i^{obs}) \quad (4.4)$$

$$b_i = \min(\mathbf{y}_i^{obs}) \quad (4.5)$$

$$y'_i = 2 \frac{y_i - b_i}{a_i - b_i} - 1 \quad (4.6)$$

Assim, cada elemento dos vetores \mathbf{y}_i^{obs} , que corresponde aos dados observados e \mathbf{y}_i^{pre} , que corresponde às previsões do modelo matemático, são normalizados. De acordo com a expressão (4.6), os valores de \mathbf{y}'_i^{obs} pertencerão ao intervalo $[-1, 1]$. O mesmo não ocorrerá com o vetor $\mathbf{y}'_i^{pre}(\mathbf{p})$, pois não se pode garantir um valor máximo e mínimo que tais respostas apresentarão para um vetor \mathbf{p} qualquer.

Diversos autores propõem formulações distintas para avaliar o erro $E_i(\mathbf{p})$. Baseado nos trabalhos de Papon *et al.* (2012), Buljak (2012) e Yin *et al.* (2017), emprega-se a seguinte formulação para a medida de erro:

$$E_i(\mathbf{p}) = \sqrt{\frac{1}{N_i} \mathbf{r}_i^T \mathbf{W}_i \mathbf{r}_i} \quad (4.7)$$

$$\mathbf{r}_i = \mathbf{y}'_i^{pre}(\mathbf{p}) - \mathbf{y}'_i^{obs} \quad (4.8)$$

em que N_i representa o número de dados observados para o conjuntos de dados i , $\mathbf{y}'_i{}^{pre}(\mathbf{p})$ é o vetor das respostas normalizadas do modelo matemático, $\mathbf{y}'_i{}^{obs}$ corresponde ao conjunto de dados normalizados empregados no ajuste e \mathbf{W}_i é uma matriz quadrada com os pesos atribuídos levando em consideração, por exemplo, possíveis incertezas nas medições. Neste trabalho a matriz \mathbf{W}_i é diagonal, para que cada elemento da diagonal se refira exclusivamente a um dado.

Nos últimos anos, diversos algoritmos de otimização vêm sendo estudados a fim de minimizar a função objetivo. Diversos estudos analisam as duas principais vertentes do ramo da otimização: determinísticas e meta-heurísticas. A Figura 4.2 descreve os passos necessários para realizar uma análise inversa a partir de um algoritmo de otimização genérico. A seguir é feita uma breve descrição desses algoritmos e de algumas aplicações e conclusões da literatura sobre tais metodologias.

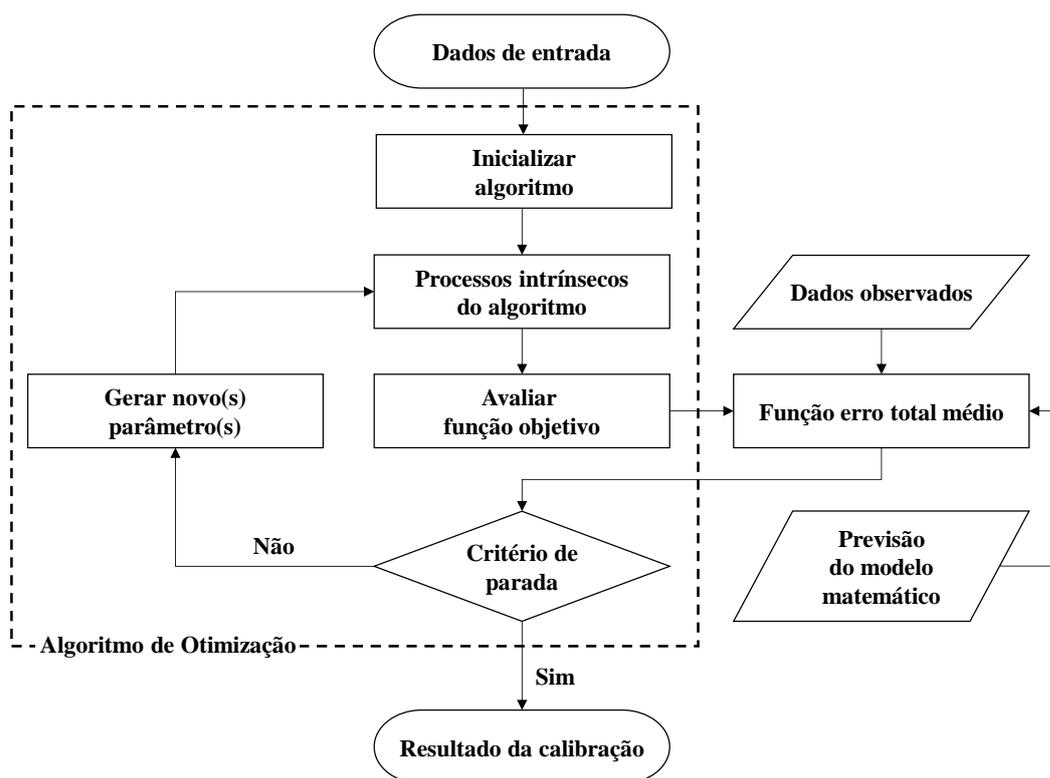


Figura 4.2 – Fluxograma da análise inversa

4.3.2. Algoritmos Determinísticos

Diversos trabalhos, sobretudo os primeiros artigos aplicados à engenharia civil, como Gioda e Maier (1980), Cividini, Jurina e Gioda (1981) e Arai, Ohta e Yasui (1983) empregam algoritmos determinísticos na calibração de parâmetros. Tais algoritmos, em geral, necessitam calcular derivadas da função objetivo, o que muitas vezes não é uma tarefa trivial.

Algoritmos determinísticos, em geral, são capazes de obter mínimos locais. Esses algoritmos necessitam de uma solução inicial para iniciar a análise. Dentre esses algoritmos, pode-se citar os métodos de Gauss-Newton, Levenberg-Marquardt e Quase-Newton, que são exemplos de algoritmos de otimização irrestrita. Tais algoritmos são discutidos por Nocedal e Wright (2006). Le, Fatahi e Khabbaz (2015) empregam um algoritmo do tipo região de confiança para calibrar um modelo viscoplástico a partir de dados experimentais para uma argila. O interessante é que o algoritmo empregado é capaz de considerar restrições durante o processo de otimização. Dessa forma, pode-se limitar o intervalo superior e inferior de cada parâmetro a ser calibrado, o que pode ser importante para que o parâmetro faça sentido fisicamente.

Swoboda *et al.* (1999) apresentam a solução de um problema inverso por meio do método de Gauss-Newton, aplicado a modelos geotécnicos. As derivadas da função objetivo são calculadas analiticamente, derivando a matriz de rigidez do elemento finito com relação aos parâmetros desconhecidos.

Tang e Kung (2009) reportam que durante seu estudo de calibração de parâmetros de escavações profundas, partindo de diferentes soluções iniciais, o algoritmo por eles empregado (método quase-Newton BFGS) produziu diferentes resultados para distintos pontos de partida.

Papon *et al.* (2012) desenvolvem em seu trabalho uma comparação entre um algoritmo determinístico (algoritmo de Nelder-Mead) e um método estocástico (algoritmo genético). Os autores destacam que os algoritmos estocásticos são mais robustos e não sofrem problemas de bacias de atração, como os determinísticos, que podem produzir diferentes soluções para distintas soluções iniciais.

Diversos autores comparam a eficiência, aplicabilidade e qualidade das soluções obtidas entre os algoritmos determinísticos e meta-heurísticos. A

conclusão mais básica é que os algoritmos determinísticos são mais eficientes computacionalmente que os algoritmos meta-heurísticos (Rechea, Levasseur e Finno, 2008). Todavia, Levasseur *et al.* (2008), Papon *et al.* (2012) e Moreira *et al.* (2013) reportam que os métodos determinísticos não são adequados quando a topologia da função objetivo é complexa. Esses algoritmos dependem da solução inicial, convergindo para o mínimo local mais próximo, e por isso devem ser empregados com cautela.

4.3.3. Algoritmos Meta-heurísticos

Nas últimas décadas, algoritmos conceitualmente diferentes dos métodos clássicos de otimização vêm sendo desenvolvidos. Esses métodos têm como base determinadas características e comportamentos de processos naturais, como evolução e seleção biológica, resfriamento de metais, comportamento de animais, entre outros. Em geral, eles são mais adequados para encontrar mínimos globais; por outro lado, necessitam de um grande esforço computacional. A seguir são apresentados dois desses métodos, citando algumas das aplicações disponíveis na literatura.

4.3.3.1. Otimização por Enxame de Partículas

A otimização por enxame de partículas é uma técnica desenvolvida por Kennedy e Eberhart (1995) e inspirada na inteligência coletiva dos animais, tais como pássaros e peixes. Nesse algoritmo, entidades chamadas de partículas são inicialmente geradas randomicamente no domínio do problema e se movimentam nesse espaço a partir de perturbações aleatórias.

Esse algoritmo apresenta inicialmente uma busca aleatória e desordenada, que é organizada a partir do padrão percebido pelas partículas, já que as posições das partículas são atualizadas com base em regras pré-estabelecidas, que levam em consideração o histórico de localização das partículas. À medida que o algoritmo evolui, as partículas se movem em direção à solução do problema.

O cálculo das novas posições das partículas para cada nova iteração é feito a partir do parâmetro denominado velocidade, que depende da melhor localização

que a partícula já possuiu e da melhor localização já obtida por qualquer partícula do enxame. Ademais, consideram-se fatores constantes e aleatórios no cálculo.

Algumas aplicações recentes em geomecânica e geotecnia foram desenvolvidos por Zhang, Gallipoli e Augarde (2009), Zhao e Yin (2009), Yazdi, Kalantary e Yazdi (2012), Knabe *et al.* (2013), Zhang, Gallipoli e Augarde (2013) e Jia e Chi (2015), revelando a vasta gama de problemas que podem ser resolvidos por meio desse algoritmo.

4.3.3.2. Algoritmos Genéticos

Em 1975, John Holland introduziu os algoritmos genéticos, inspirado nos mecanismos e processos da seleção natural e genética. Esse tipo de algoritmo busca uma solução ótima de uma função objetivo por intermédio da aplicação de mecanismos como seleção, cruzamento e mutação. O algoritmo simula computacional e iterativamente um processo evolutivo de várias gerações de uma população. Nesse contexto, uma geração representa uma iteração do processo, enquanto a população contém os candidatos à solução ótima do problema.

O algoritmo genético é iniciado com os indivíduos da população inicial gerados randomicamente, os quais evoluirão à cada iteração. Tal evolução ocorre mediante aos operadores de seleção, cruzamento e mutação, feitos à cada geração. Esse é um processo estocástico e não requer o cálculo de derivadas. Além disso, o algoritmo não necessita de uma solução inicial, mas de um intervalo para cada uma das variáveis a serem otimizadas, conforme apresentado na equação (4.2).

Algumas aplicações recentes em geotecnia e geomecânica são conduzidas por Hashash *et al.* (2010), Samarajiva, Macari e Wathugala (2010), Rokonuzzaman e Sakai (2010), Papon *et al.* (2012), Jin *et al.* (2016), Yin *et al.* (2016) e Jin *et al.* (2017). Esses estudos têm como objetivo calibrar parâmetros, validando a aplicação do algoritmo na determinação de parâmetros desconhecidos.

Por ser um dos algoritmos mais usuais para a solução de problemas de otimização via processo estocástico, esse algoritmo foi selecionado para ser estudado, e o mesmo será apresentado no capítulo a seguir.

5 Desenvolvimento de um Algoritmo Genético

Para realizar a calibração de parâmetros por meio da análise inversa, adotou-se o algoritmo genético como método de otimização. Diferentemente de outros algoritmos clássicos, esse é adequado para buscar solução(ões) global(is) de problemas de otimização. Tem-se, então, o objetivo de solucionar o seguinte problema de minimização, dada a função $F(\mathbf{x})$:

$$\min_{\mathbf{x} \in \Omega} F(\mathbf{x}) \quad (5.1)$$

em que \mathbf{x} é um vetor de variáveis contínuas e Ω é o domínio do problema, representado por $l_i \leq x_i \leq u_i, i \in \{1; 2; \dots; D\}$, sendo D a dimensão do problema. A solução de (5.1) é \mathbf{x}^* , tal que $F(\mathbf{x}^*) \leq F(\mathbf{x}), \forall \mathbf{x}^* \in \Omega$.

Os algoritmos genéticos mimetizam a evolução biológica a partir de operadores que simulam os mecanismos desta. Aplica-se a ideia da sobrevivência dos indivíduos mais aptos de uma população. Um algoritmo genético típico consiste em um grupo de potenciais soluções (chamados de indivíduos ou cromossomos) sujeitos à esquemas de seleção, cruzamento e mutação (denominados aqui de operadores), de acordo com uma função de aptidão, que identifica quais são os melhores e piores indivíduos da população. Cada indivíduo \mathbf{x} é composto por genes, em que cada um desses corresponde a uma das componentes do vetor. O fluxograma apresentado na Figura 5.1 sintetiza os passos de um algoritmo genético.

As primeiras implementações desenvolvidas consideravam uma codificação binária para representar esses indivíduos. Esse tipo de representação é robusta, porém pode exigir muito esforço computacional, especialmente quando o domínio do problema é grande (Goldberg, 1991).

O algoritmo apresentado neste trabalho é baseado nos trabalhos de Deep e Thakur (2007), Kaelo e Ali (2007), Sawyerr, Adewumi e Ali (2014) e Yin *et al.* (2016). A seguir são apresentados os conceitos que envolvem a implementação de um algoritmo genético.

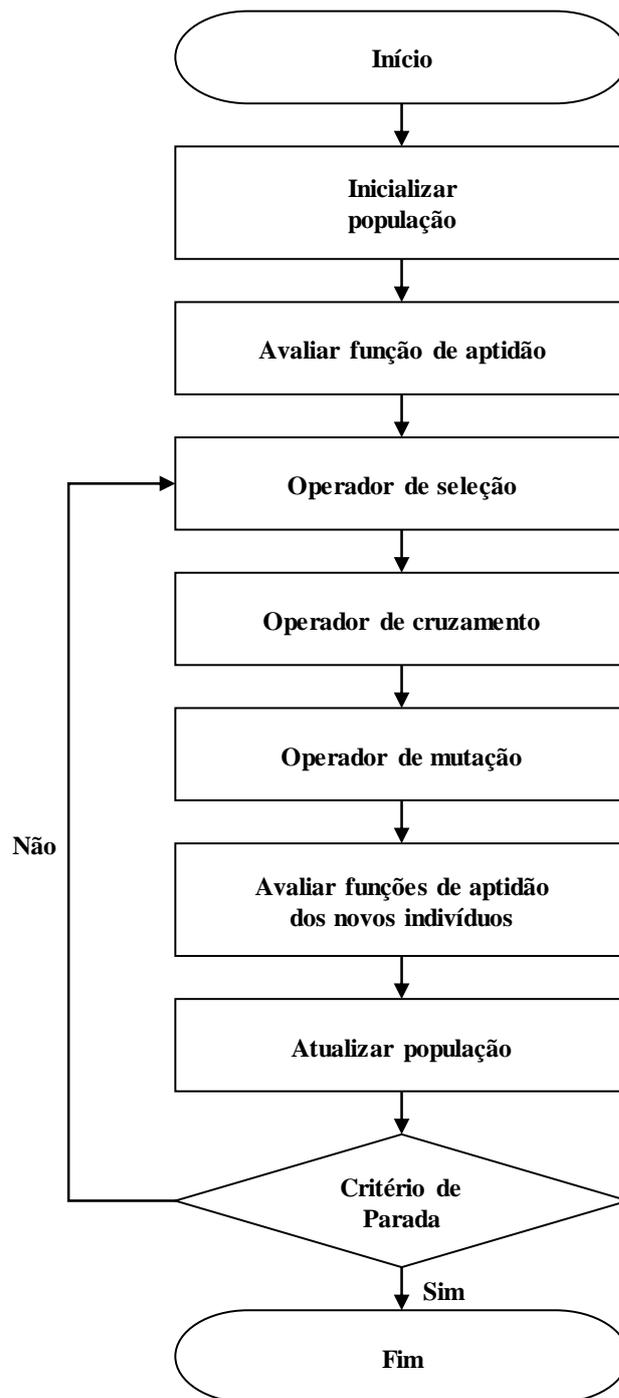


Figura 5.1 – Fluxograma de um algoritmo genético

5.1. Função de Aptidão

A função de aptidão define a qualidade de uma solução. Assim, quanto maior o valor desta função, melhor é o indivíduo. Esta função pode ser definida de

diversas formas, como apresentado por Rao (2009) e Arora (2012). Esta também depende da existência ou não de restrições.

Quando o problema é maximizar uma função sem restrições, pode-se empregar a própria função objetivo como função de aptidão. Quando se deseja minimizar uma função sem restrições, sugere-se, neste trabalho, em empregar o oposto da função objetivo, que é a mesma metodologia empregada para aplicar os algoritmos clássicos de otimização em problemas de maximização. Dessa forma, a função de aptidão $\mathcal{F}(\mathbf{x})$ empregada neste trabalho é dada por:

$$\mathcal{F}(\mathbf{x}) = -F(\mathbf{x}) \quad (5.2)$$

5.2. Operador de Seleção

Esse operador é crucial para a manutenção da diversidade populacional. Ele tem como objetivo selecionar indivíduos com o propósito de produzir cruzamentos e mutações. Os mecanismos de seleção devem ser capazes de favorecer a seleção dos melhores indivíduos para a realização dos cruzamentos. Dessa forma, tem-se o conceito de pressão seletiva, que segundo Bäck (1994), é usado para caracterizar a ênfase em selecionar os melhores indivíduos. Quanto maior for a pressão seletiva, maior será a quantidade de melhores indivíduos favorecidos. Todavia, Mokhadde e Kakde (2014) observam que quanto maior a pressão seletiva, menor é a diversidade da população.

Neste trabalho, optou-se por empregar a seleção por torneios, já que para esse tipo de seleção, pode-se listar os seguintes benefícios em relação a outros métodos: facilidade de implementação, permite trabalhar com arquiteturas paralelas e ajusta a pressão seletiva para se adaptar a diferentes domínios apenas modificando o número de indivíduos no torneio. Esses fatores vêm contribuindo para que esse tipo de seleção seja cada vez mais empregado nas implementações de algoritmos genéticos (Miller e Goldberg, 1995).

Na seleção por torneio, um grupo de indivíduos (chamado aqui de tamanho do torneio) é selecionado randomicamente da população e suas funções de aptidão são comparadas. O indivíduo que tem a maior função de aptidão do torneio é o vencedor desse, que é colocado em uma população intermediária, conhecida em

inglês por *mating pool*. Para manter a diversidade, neste trabalho, opta-se por adotar o tamanho de torneio igual a 2, conforme sugerido por Deb (2012) e Yin *et al.* (2016). Esse operador é acionado até que a população intermediária seja totalmente preenchida, de acordo com o tamanho pré-estabelecido pelo usuário.

5.3. Operador de Cruzamento

O cruzamento consiste em combinar os genes x_i^a e x_i^b de duas soluções (chamadas aqui de pais) a fim de gerar os genes \bar{x}_i^a e \bar{x}_i^b dos descendentes, com o objetivo de evoluir a população, introduzindo variações nesta. Esse é um operador fundamental e domina largamente a performance do algoritmo genético. Considera-se uma probabilidade pré-definida desse operador acontecer, para cada par de indivíduos selecionados. Como será apresentado no estudo paramétrico realizado neste trabalho, esse parâmetro de probabilidade pode ter grande impacto no sucesso do algoritmo.

Neste trabalho são avaliados dois operadores, apresentados a seguir. Para esses dois, existe a possibilidade de se gerar descendentes fora do domínio Ω . Dessa forma, sugere-se realizar um número pré-definido de tentativas K , caso esse domínio seja violado. Se por acaso, falha-se em gerar indivíduos dentro do domínio, os próprios pais são retornados como seus próprios descendentes.

5.3.1. Cruzamento Aritmético

Esse operador é definido como uma combinação linear entre os pais selecionados para o cruzamento. Isto posto, sendo α_i um número aleatório uniformemente distribuído no intervalo $[-1; 1]$, os genes \bar{x}_i^a e \bar{x}_i^b dos indivíduos podem ser calculados por meio de:

$$\bar{x}_i^a = \alpha_i x_i^a + (1 - \alpha_i) x_i^b \quad (5.3)$$

$$\bar{x}_i^b = (1 - \alpha_i) x_i^a + \alpha_i x_i^b \quad (5.4)$$

É importante frisar, que para cada gene, tem-se um α_i distinto.

5.3.2. Cruzamento Binário Simulado

Esse operador, proposto por Deb e Agrawal (1995), simula o comportamento do operador conhecido com cruzamento de um ponto, para indivíduos representados por codificação binária. É atribuída uma grande probabilidade para se gerar um ponto próximo a cada um dos pais. Os descendentes calculados por esse operador são definidos a partir de:

$$\bar{x}_i^a = 0,5 \left((1 + \beta_i)x_i^a + (1 - \beta_i)x_i^b \right) \quad (5.5)$$

$$\bar{x}_i^b = 0,5 \left((1 + \beta_i)x_i^a + (1 - \beta_i)x_i^b \right) \quad (5.6)$$

considerando que:

$$\beta_i = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & \text{se } r \leq 0,5 \\ \left(\frac{1}{2(1-u)} \right)^{\frac{1}{\eta+1}}, & \text{se } r > 0,5 \end{cases} \quad (5.7)$$

Em que r é um número aleatório uniformemente distribuído no intervalo $[0; 1]$, e η é um parâmetro, em que Yin *et al.* (2016) sugere ser igual a 20.

5.4. Operador de Mutação

Com o objetivo de prevenir que o algoritmo convirja para um mínimo local, um operador de mutação é necessário em um algoritmo genético. Esse operador tem como função alterar genes aleatoriamente de indivíduos selecionados, com o intuito de manter a diversidade da população. Isso é essencial na busca de uma solução global ótima. Cada gene do indivíduo possui uma probabilidade pré-definida de receber uma mutação.

Neste trabalho, dois operadores de mutação são analisados. Tais operadores têm a capacidade, em estágios iniciais, realizar uma busca uniforme pelo domínio do problema e, posteriormente, a busca se torna local, em estágios finais. Define-se

t como o número da geração (iteração) atual e T como o número máximo de gerações definidos pelo usuário.

Similarmente ao operador de cruzamento, existe a possibilidade de se gerar descendentes fora do domínio do problema. Assim, adota-se a mesma abordagem, em que se tenta um número pré-definido de vezes gerar um indivíduo com mutação e quando não se tem sucesso, devolve-se o indivíduo original como resultado do operador.

5.4.1. Mutaç o N o Uniforme

Esse operador considera os limites para cada par metro x_i , definidos por u_i e l_i , respectivamente, limite superior e inferior. Dessa forma, cada par metro sujeito   mutaç o \bar{x}_i   dado por:

$$\bar{x}_i = \begin{cases} x_i + \xi(t, u_i - x_i), & \text{se } r_1 \leq 0,5 \\ x_i + \xi(t, x_i - l_i), & \text{se } r_1 > 0,5 \end{cases} \quad (5.8)$$

em que:

$$\xi(t, \chi) = \chi \left(1 - r_2 \left(1 - \frac{t}{T} \right)^b \right) \quad (5.9)$$

considerando que r_1 e r_2 s o n meros rand micos uniformemente distribu dos no intervalo $[0; 1]$ e b   um par metro positivo fornecido pelo usu rio, que determina o grau de n o uniformidade. Neste trabalho, emprega-se $b = 5$, conforme adotado por Elsayed, Sarker e Essam (2010).

5.4.2. Mutaç o Rand mica Din mica

Esse operador, desenvolvido por Chuang, Chen e Hwang (2015),   definido pela seguinte regra para cada gene \bar{x}_i :

$$\bar{x}_i = x_i + s_m \phi_0 r (u_i - l_i) \quad (5.10)$$

considerando que:

$$s_m = \left(1 - \frac{t}{T}\right)^a \quad (5.11)$$

em que r é um número randômico uniformemente distribuídos no intervalo $[0, 1]$, a é um parâmetro positivo fornecido pelo usuário para controlar o decaimento do parâmetro s_m e ϕ_0 é um parâmetro fornecido pelo usuário, no intervalo $(0, 1]$, para controlar o tamanho de uma perturbação randômica provocada por r . Adota-se os valores 4 e 0,5 para os parâmetros, respectivamente, a e ϕ_0 , conforme apresentado por Chuang, Chen e Hwang (2015).

5.5. Descrição do Algoritmo Genético

A seguir é descrito o algoritmo genético desenvolvido neste trabalho. Por ser um algoritmo populacional, uma população de N potenciais soluções $\mathbf{P}_t = \{\mathbf{x}_t^1; \mathbf{x}_t^2; \dots; \mathbf{x}_t^N\}$ é mantida à cada geração t . Para inicializar o algoritmo em $t = 0$, gera-se uma população \mathbf{P}_0 , de forma randômica, dentro de um domínio Ω , definido por intervalos superiores e inferiores para cada variável, definidos, respectivamente, pelos vetores \mathbf{u} e \mathbf{l} . Sawyerr, Adewumi e Ali (2015) sugerem que o número de indivíduos que compõe a população seja $\min(100, 10D)$, em que D corresponde ao número de genes por indivíduo (dimensão do problema).

Após isto, faz-se $t := t + 1$ e aplica-se um operador de seleção para eleger M indivíduos de \mathbf{P}_{t-1} para compor uma população intermediária $\bar{\mathbf{P}}_t$, considerando que $M \leq N$. Neste trabalho adota-se M como um número par, já que são gerados dois descendentes por cruzamento. Dentre esses indivíduos, é permitido existir cópias do mesmo indivíduo. Os indivíduos desta população serão pareados e submetidos aos operadores de cruzamento e mutação.

Cada operador de cruzamento e mutação é acionado a partir de uma probabilidade. Dessa forma, define-se que r_1 e r_2 são números randômicos uniformemente distribuídos no intervalo $[0; 1]$ gerados à cada vez que se verifica a aplicação de um operador de cruzamento ou mutação.

O operador de cruzamento é aplicado quando $p_c \leq r_1$ for verificado verdadeiro. O operador é acionado um número de vezes suficientes para gerar M descendentes. Dessa forma, aplica-se as expressões (5.3) e (5.4) ou (5.5) e (5.6). Gerados os descendentes, copia-se esses para um novo grupo de indivíduos $\hat{\mathbf{P}}_t$. Se a expressão $p_c \leq r_1$ não é atendida, os pais selecionados não realizam cruzamento e são diretamente copiados para $\hat{\mathbf{P}}_t$.

O operador de mutação é aplicado quando $p_m \leq r_2$ for verificado verdadeiro, sendo esta verificação realizada para cada gene dos indivíduos contidos em $\hat{\mathbf{P}}_t$. Isto posto, aplica-se a expressão (5.8) ou (5.10). Cada indivíduo que recebeu mutação é introduzido num novo grupo de indivíduos $\tilde{\mathbf{P}}_t$. Mesmo que o indivíduo não receba mutação, ele é introduzido no grupo $\tilde{\mathbf{P}}_t$.

Ao fim desta geração, os indivíduos contidos em $\tilde{\mathbf{P}}_t$ são reintroduzidos na população. Dessa forma, unem-se os $N - M$ melhores indivíduos de \mathbf{P}_{t-1} e os indivíduos de $\tilde{\mathbf{P}}_t$ para compor a população \mathbf{P}_t . Dessa forma, sendo $M < N$, tem-se que os melhores indivíduos de uma geração sobrevivem para a próxima, sendo esta estratégia conhecida com elitismo.

No algoritmo adotado nesse trabalho, considera-se como critério de parada o número máximo de iterações. Ainda pode-se considerar o cálculo do desvio padrão dos melhores indivíduos das últimas gerações. Considera-se uma quantidade pré-definida S de gerações para tal cálculo e quando esse desvio for menor que uma tolerância, pode-se encerrar o algoritmo. A seguir são apresentados os Quadros 5.1a e 5.1b, com o resumo do algoritmo descrito aqui.

O algoritmo genético descrito nessa seção foi implementado na linguagem Python 3.6.7 (2018). Além dos exemplos apresentados na próxima seção, esta implementação é adotada para a solução dos problemas de otimização nas análises inversas apresentadas no próximo capítulo.

Quadro 5.1a – Algoritmo genético (parte 1)

- 1) Dados de entrada: $F(\mathbf{x})$, \mathbf{u} , \mathbf{l} , D , N , M , T , S , tol , K , p_c , p_m
- 2) Inicializar: $t = 0$
- 3) Gerar população inicial $\mathbf{P}_t = \{\mathbf{x}_t^1; \mathbf{x}_t^2; \dots; \mathbf{x}_t^N\}$, sendo $l_i \leq x_{i,t}^k \leq u_i$, tal que $i \in \{1, 2, \dots, D\}$
- 4) Calcular a função de aptidão para todos os indivíduos \mathbf{x}_t^k de \mathbf{P}_t :
 $\mathcal{F}(\mathbf{x}_t^k) = -F(\mathbf{x}_t^k)$, tal que $k \in \{1; 2; \dots; N\}$
- 5) Ordenar indivíduos de \mathbf{P}_t :
 Relacionar os indivíduos \mathbf{P}_t conforme as funções de aptidão. Selecionar o melhor indivíduo da população e inserir em \mathbf{x}_t^* . Fazer $t := t + 1$.
- 6) Operador de seleção:
 Selecionar M indivíduos da população \mathbf{P}_{t-1} , para construir a população intermediária $\bar{\mathbf{P}}_t$, por meio do operador de seleção descrito na Seção 5.2
- 7) Operador de cruzamento:
 Para i de 1 até $M/2$:
 Gerar um número aleatório r_1 uniformemente distribuídos no intervalo $[0; 1]$
 Se $p_c \leq r_1$:
 Aplicar operador de cruzamento aos indivíduos de índice $2i - 1$ e $2i$ da população $\bar{\mathbf{P}}_t$, conforme descrito na Seção 5.3. Deve-se verificar se o indivíduo respeita $l_j \leq x_{j,t}^k \leq u_j$, $j \in \{1; 2; \dots; D\}$.
 Caso contrário, reiniciar o último operador no máximo K vezes.
 Caso o número de reinícios se iguale a K , sem gerar descendentes dentro do domínio do problema, retornar os pais como resultado do cruzamento
 Caso contrário:
 Os pais são retornados como os indivíduos gerados pelo cruzamento
 Inserir os indivíduos gerados pelo cruzamento na população $\hat{\mathbf{P}}_t$
 Ir para o Quadro 5.1b

Quadro 5.1b – Algoritmo genético (parte 2)

8) Operador de mutação:

Para i de 1 até M :

Para j de 1 até D :

Gerar um número aleatório r_2 uniformemente distribuídos no intervalo $[0; 1]$

Se $p_m \leq r_2$:

Aplicar operador de mutação ao indivíduo de índice i da população $\hat{\mathbf{P}}_t$, no gene j conforme descrito na Seção 5.4

Deve-se verificar se o indivíduo respeita $l_j \leq x_{j,t}^k \leq u_j$. Caso contrário, reiniciar a aplicação de mutação no indivíduo no máximo K vezes. Caso o número de reinícios se iguale a K , sem gerar um indivíduo dentro do domínio do problema, retornar o indivíduo original como resultado da mutação

Inserir o indivíduo gerado pela mutação na população $\tilde{\mathbf{P}}_t$

9) Calcular a função de aptidão para todos os indivíduo \mathbf{x}_t^k de $\tilde{\mathbf{P}}_t$:

$\mathcal{F}(\mathbf{x}_t^k) = -F(\mathbf{x}_t^k)$, tal que $k \in \{1; 2; \dots; M\}$

10) Atualizar população, unindo os $N - M$ melhores indivíduos de \mathbf{P}_{t-1} e os indivíduos de $\tilde{\mathbf{P}}_t$ para gerar a população \mathbf{P}_t **11) Critério de parada:**

Se $t \geq S$:

$$\bar{F}^* = \frac{1}{S} \sum_{i=t-S+1}^t F(\mathbf{x}_i^*)$$

$$dev = \sqrt{\frac{1}{S-1} \sum_{i=t-S+1}^t (F(\mathbf{x}_i^*) - \bar{F}^*)^2}$$

Se $dev \leq tol$:

Retornar o melhor indivíduo \mathbf{x}_t^* e encerrar o algoritmo

Se $t < T$:

Retornar ao passo 5) do Quadro 5.1a

Caso contrário:

Retornar o melhor indivíduo \mathbf{x}_T^* e encerrar o algoritmo

5.6. Estudo Paramétrico

Para avaliar a performance do algoritmo apresentado e estudar o impacto de alguns parâmetros na solução final, foram escolhidos 28 problemas de otimização global com soluções conhecidas. Destas funções, tem-se 6 funções escaláveis (de dimensão variável) que são analisadas separadamente das outras funções de dimensão fixas (ou fixadas). Um breve resumo sobre esses problemas é apresentado nas Tabelas 5.1 e 5.2. Maiores informações sobre esses problemas são apresentadas por Ali, Khompatraporn e Zabinsky (2005).

As análises apresentadas aqui são baseadas no estudo desenvolvido por Sawyerr, Adewumi e Ali (2014) para avaliar a performance das implementações desenvolvidas pelos autores. Seguindo a metodologia descrita por eles, cada problema de otimização é executado 25 vezes. Um sucesso do algoritmo é computado quando a diferença absoluta entre os valores mínimos da função retornado pelo algoritmo e a(s) solução(ões) conhecida(s), respectivamente, F_{min} e $F(\mathbf{x}^*)$, é menor que uma tolerância. Os autores seguem o critério de tolerância conforme apresentado por Sawyerr, Ali e Adewumi (2011), ou seja:

$$|F(\mathbf{x}^*) - F_{min}| \leq 0,009 \quad (5.12)$$

Para este estudo, conforme Sawyerr, Adewumi e Ali (2014), fixou-se um número de gerações a serem respeitados por todos os processos de otimização. Dessa forma, os parâmetros S e tol não são levados em consideração neste estudo.

Foram desenvolvidos estudos paramétricos de três parâmetros: probabilidade de mutação, probabilidade crossover e a taxa da população renovada a cada geração (que se refere à fração que a população intermediária representa em relação à população total). Ademais, avaliou-se o impacto dos diferentes operadores de cruzamento e mutação descritos neste trabalho na solução final do problema.

As funções estudadas apresentam diversos níveis de dificuldade e por isso considera-se um bom grupo de funções para determinar os parâmetros a serem empregados nas aplicações de calibração de modelos desenvolvidas neste trabalho.

Tabela 5.1 – Lista de problemas de otimização com dimensão fixada

Nº	Problema	D	Intervalo	$F(x^*)$
1	Aluffi-Pentini	2	$-10 \leq x_1, x_2 \leq 10$	-0,3523
2	Becker and Lago	2	$-10 \leq x_1, x_2 \leq 10$	0
3	Bohachevsky 1	2	$-50 \leq x_1, x_2 \leq 50$	0
4	Bohachevsky 2	2	$-50 \leq x_1, x_2 \leq 50$	0
5	Branin	2	$-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$	0,397887358
6	Cosine Mixture	4	$-1 \leq x_1, x_2, x_3, x_4 \leq 1$	-0,4
7	Dekkers and Aarts	2	$-20 \leq x_1, x_2 \leq 20$	-24776,5183
8	Goldstein and Price	2	$-2 \leq x_1, x_2 \leq 2$	3
9	Gulf Reseach	3	$0,1 \leq x_1 \leq 100, 0 \leq x_2 \leq 25,6, 0 \leq x_3 \leq 5$	0
10	Hartman 3	3	$0 \leq x_1, x_2, x_3 \leq 1$	-3,862782
11	Hartman 6	6	$0 \leq x_i \leq 1, i \in \{1; 2; 3; 4; 5; 6\}$	-3,322368
12	Helical Valley	3	$-10 \leq x_1, x_2, x_3 \leq 10$	0
13	Hosaki	2	$0 \leq x_1 \leq 5, 0 \leq x_2 \leq 6$	-2,3458
14	Kowalik	4	$0 \leq x_i \leq 0,42, i \in \{1; 2; 3; 4\}$	0,00030748
15	Multi-Gaussian	2	$-2 \leq x_1, x_2 \leq 2$	-1,29695
16	Periodic	2	$-10 \leq x_1, x_2 \leq 10$	0,9
17	Powell's Quadratic	4	$-10 \leq x_i \leq 10, i \in \{1; 2; 3; 4\}$	0
18	Schubert	2	$-10 \leq x_1, x_2 \leq 10$	-186,7309
19	Shekel 5	4	$0 \leq x_i \leq 10, i \in \{1; 2; 3; 4\}$	-10,1532
20	Shekel 7	4	$0 \leq x_i \leq 10, i \in \{1; 2; 3; 4\}$	-10,4029
21	Shekel 10	4	$0 \leq x_i \leq 10, i \in \{1; 2; 3; 4\}$	-10,5364
22	Sinusoidal	20	$0 \leq x_i \leq 180, i \in \{1; 2 \dots; 20\}$	-3,5

Tabela 5.2 – Lista de problemas de otimização com dimensão variável

Nº	Problema	D	Intervalo	$F(x^*)$
23	Ackley	5 a 40	$-30 \leq x_i \leq 30, i \in \{1; 2 \dots; D\}$	0
24	Spherical	5 a 40	$-5,12 \leq x_i \leq 5,12, i \in \{1; 2 \dots; D\}$	0
25	Griewank	5 a 40	$-600 \leq x_i \leq 600, i \in \{1; 2 \dots; D\}$	0
26	Rastrigin	5 a 40	$-5,12 \leq x_i \leq 5,12, i \in \{1; 2 \dots; D\}$	0
27	Rosenbrock	5 a 40	$-30 \leq x_i \leq 30, i \in \{1; 2 \dots; D\}$	0
28	Schwefel	5 a 40	$-500 \leq x_i \leq 500, i \in \{1; 2 \dots; D\}$	0

A fim de checar a eficiência do algoritmo desenvolvido neste trabalho, os resultados gerados são comparados com os apresentados por Sawyerr, Adewumi e Ali (2014), que desenvolveram duas implementações de algoritmos genéticos: um algoritmo típico e um que considera uma estratégia de busca local, respectivamente denominados de RCGA e RCGAu. Vale destacar que a abordagem com busca local necessita de um maior número de avaliações da função objetivo, o que faz com que o método seja mais custoso computacionalmente. Os autores apresentam apenas um

grupo de resultados para cada algoritmo, considerando um único conjunto de parâmetros. A comparação não é totalmente justa, já que os parâmetros adotados pelos autores são diferentes. Mas a comparação é válida para demonstrar que mesmo os algoritmos genéticos sendo considerados metodologias de otimização global, nem sempre detectam uma solução ótima global.

Para este estudo paramétrico, foram definidos os parâmetros expostos na Tabela 5.3, exceto para aqueles parametrizados. Os Gráficos 5.1 a 5.3 apresentam o percentual de sucesso do algoritmo implementado para resolver os problemas descritos na Tabela 5.1. Cada conjunto de parâmetros é analisado para quatro combinações dos operadores de cruzamento e mutação: Cruzamento Aritmético (CA), Cruzamento Binário Simulado (CBS), Mutação Não Uniforme (MNU) e Mutação Randômica Dinâmica (MRD).

Tabela 5.3 – Parâmetros do estudo paramétrico

Parâmetro	Valor
Tamanho da População (N)	100
Tamanho da População Intermediária (M)	80
Número Máximo de Gerações (T)	1000
Número de Tentativas de Cruzamento ou Mutação (K)	10
Probabilidade de Cruzamento (p_c)	0,7
Probabilidade de Mutação (p_m)	0,05

O Gráfico 5.1 sintetiza os resultados obtidos mantendo os parâmetros apresentados na Tabela 5.3, exceto a probabilidade de mutação, a qual foi atribuído os valores 0,001, 0,005, 0,01, 0,05, 0,1 e 0,15. Observa-se que não foi possível solucionar os problemas em sua totalidade. Todavia obteve-se uma ótima taxa de acertos. O maior percentual de acertos do algoritmo apresentado neste trabalho foi 93,45%, quando o algoritmo genético padrão de Sawyerr, Adewumi e Ali (2014) atingiu 80,91% e a implementação com busca local, 92,91%. Observou-se uma maior taxa de acertos para o operador de cruzamento aritmético em relação ao cruzamento binário simulado. Por fim, o comportamento que se observou foi o aumento da taxa de acertos à medida que a probabilidade de mutação foi incrementada. Isso acontece, pois, o aumento das mutações introduzidas na população permite uma maior busca do domínio do problema.

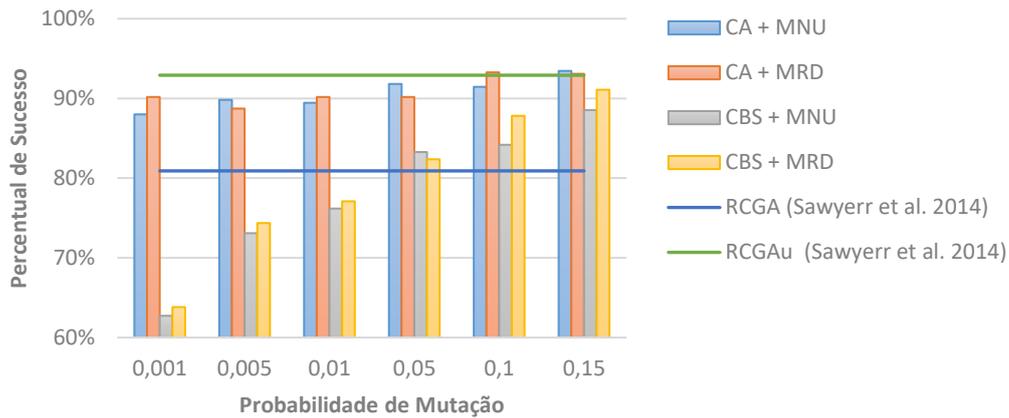


Gráfico 5.1 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão fixa

Para o estudo paramétrico referente à probabilidade de cruzamento, apresenta-se o Gráfico 5.2. Dessa forma, mantiveram-se os parâmetros apresentados na Tabela 5.3, com exceção desta probabilidade, na qual foram atribuídos os valores 0,5, 0,6, 0,7, 0,8, 0,9 e 1. Novamente, observou-se que o cruzamento aritmético produziu as maiores taxas de acerto, sendo a maior delas igual a 96%.

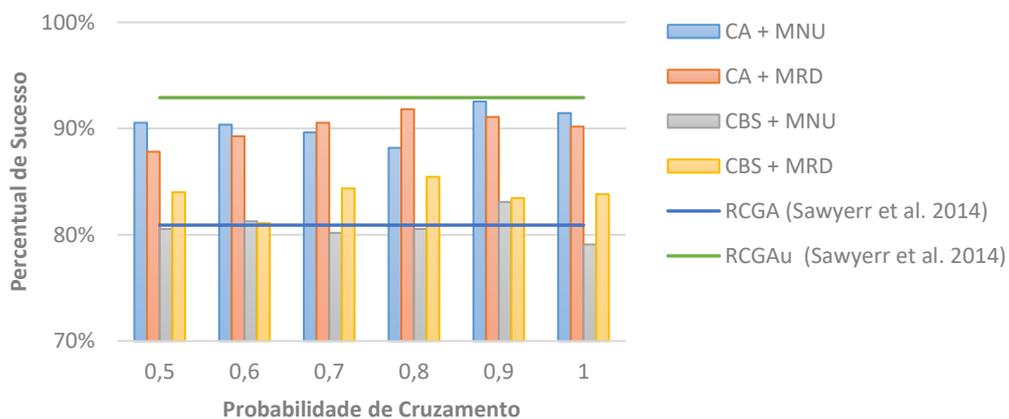


Gráfico 5.2 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão fixa

O Gráfico 5.3 se refere ao estudo do tamanho da população intermediária (*mating pool*). Dessa forma, considerou-se neste estudo $0,1N$, $0,2N$, $0,3N$, $0,4N$, $0,5N$, $0,6N$, $0,7N$, $0,8N$ e $0,9N$ para o tamanho desta população. Como o número

de gerações foi fixado para todos as análises, para populações intermediárias pequenas, o percentual de acerto foi baixo, já que o número de cruzamentos e mutações foi bem reduzido. Todavia, a partir de uma certa taxa, o percentual de sucesso se mantém aproximadamente igual, mesmo com o número de cruzamentos e mutações maior. Dessa forma, é interessante não empregar um valor tão grande para esses parâmetros a fim de reduzir o número de avaliações da função objetivo, e conseqüentemente, o esforço computacional.

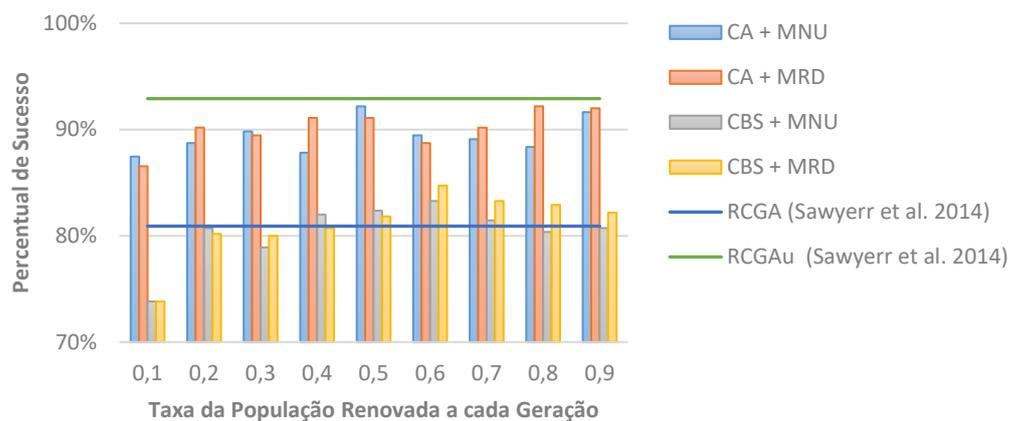


Gráfico 5.3 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão fixa

No Apêndice A, são apresentados outros gráficos, que se referem às funções indicadas na Tabela 5.2. Numa visão geral, observa-se que o cruzamento aritmético e o cruzamento binário simulado produziram resultados similares, quando combinados à mutação não uniforme, e superiores às demais combinações de operadores na maior parte das análises. De uma forma geral, o melhor comportamento do algoritmo ocorreu quando se combinou o cruzamento aritmético à mutação não uniforme.

Um comportamento interessante foi observado no estudo da probabilidade de mutação para as funções escaláveis com dimensão 20, também apresentado no Apêndice A; em que a taxa de acertos foi reduzida para as probabilidades de mutação mais baixas ou muito altas. Melhores resultados foram observados para valores intermediários, o que indica que valores elevados de mutação podem introduzir variações na população que atrapalham a convergência do algoritmo.

Observou-se que o impacto da probabilidade de mutação na solução final do problema é muito dependente do tipo de problema, já que diversos comportamentos foram observados ao variar tal parâmetro, conforme apresentado no Apêndice A. Já para o estudo paramétrico que estudou o impacto tamanho da população intermediária, verifica-se que a partir de um certo ponto, o aumento da população não tem grande impacto na solução final do problema. Por fim, observa-se que as taxas de sucesso para os problemas indicados na Tabela 5.2 foram bem reduzidas em relação aos problemas apresentados na Tabela 5.1, devido ao alto grau de complexidade dos mesmos.

5.7. Conclusões Parciais

Não é uma tarefa trivial definir parâmetros ótimos a serem adotados em um algoritmo genético, já que isso depende da natureza do problema. Todavia, a partir do estudo paramétrico desenvolvido e recomendações obtidas na literatura, sugere-se os parâmetros apresentados na Tabela 5.4. Esses parâmetros podem ser adotados inicialmente, e depois modificados de acordo com a dificuldade do problema analisado.

Tabela 5.4 – Parâmetros adotados nas aplicações deste trabalho

Parâmetro	Valor
Tamanho da População Intermediária (M)	0,8N
Número de Tentativas de Cruzamento ou Mutação (K)	5
Probabilidade de Cruzamento (p_c)	0,7
Probabilidade de Mutação (p_m)	0,01; 0,05 ou 0,15

Além disso, opta-se o operador de cruzamento aritmético e o operador de mutação não uniforme, por terem se comportado de forma mais estável ao longo de todas as análises.

O tamanho da população intermediária foi um parâmetro que teve comportamento estável ao longo do estudo paramétrico. Todavia a probabilidade de mutação não apresenta tal comportamento e, por isso, sugere-se realizar múltiplas análises, considerando os valores recomendados na Tabela 5.4. Faz-se esta sugestão, pois a natureza do problema geralmente é desconhecida, não sendo possível determinar previamente a melhor probabilidade de mutação. No estudo da

probabilidade de cruzamento diversos comportamentos bem distintos são observados ao longo das análises. Entretanto observa-se que o parâmetro foi pouco sensível na maioria dos resultados para os operadores de cruzamento aritmético e mutação não uniforme. Dessa forma, adota-se um valor intermediário para o mesmo. Assim, os próximos estudos de calibração de parâmetros serão realizados com base nesses parâmetros. É importante frisar que os parâmetros apresentados são apenas sugestões para uma análise inicial.

Por intermédio de alguns experimentos, não observou-se grande impacto do parâmetro K na solução dada pelo algoritmo. Dessa forma, reduziu-se o valor adotado na Tabela 5.3 para $K = 5$. O número de iterações e o critério de parada deve ser avaliado pelo usuário, já que esses são muito dependentes do problema.

Por fim, para avaliar os resultados obtidos a partir de tais parâmetros, os problemas de otimização estudados foram submetidos ao algoritmo genético empregando-os. Dessa vez, será avaliada também a qualidade das soluções a partir das médias e desvio padrão dos erros calculados por (5.12). Tais tabelas podem ser encontradas no Apêndice A, em que é feita uma comparação entre os resultados encontrados para a implementação apresentada e os resultados de Sawyerr, Adewumi e Ali (2014). Assim como esses últimos autores, o cálculo da média e desvio padrão foram feitos apenas considerando as análises bem sucedidas, conforme o critério (5.12).

Nestas análises, observou-se a qualidade das soluções produzidas pelo algoritmo, sobretudo para os problemas de dimensão fixada, é muito boa. Para esses problemas, as médias e desvios padrão dos erros eram inferiores à 10^{-4} . Já para os problemas com dimensão variável, assim como observado no estudo paramétrico, a taxa de sucesso era relativamente baixa. Alguns problemas desses não obtiveram uma resposta adequada, mesmo com 25 tentativas, o que mostra que, dependendo da natureza do problema, o algoritmo implementado neste trabalho pode não ser capaz de resolvê-lo.

De uma forma geral, a probabilidade de mutação $p_m = 0,15$ foi a que teve o melhor comportamento. Todavia, deve-se ter cautela, pois não é extrapolar esta informação para outros grupos de problemas.

6 Aplicações e Resultados

A seguir são apresentadas algumas aplicações acerca dos algoritmos discutidos neste trabalho. Os estudos são feitos a partir do modelo constitutivo elastoplástico *Cap Model*, conforme descrito na Seção 2.2. Todas as análises apresentadas neste trabalho consideram a elasticidade linear, considerando o módulo de elasticidade E e coeficiente de Poisson ν . De acordo com o algoritmo de mapeamento de retorno estudado, necessita-se de determinadas derivadas das funções de plastificação. Dessa forma, tais derivadas são apresentadas no Apêndice B.

A atualização das tensões foi implementada na plataforma MATLAB R2018b (2018), para que os testes a nível local (a nível do ponto de Gauss) fossem realizados. Dessa forma, a implementação desenvolvida está de acordo com os Quadros 3.5a a 3.5c. Para o mesmo, é necessário informar as deformações totais $\boldsymbol{\varepsilon}_{n+1}$, as deformações plásticas $\boldsymbol{\varepsilon}_n^p$, e as variáveis internas \boldsymbol{q}_n .

Ademais, a metodologia descrita neste trabalho foi implementada no *framework* GeMA, a fim de obter soluções via elementos finitos. Proposto inicialmente por Mendes (2016) e desenvolvido no Instituto Tecgraf/PUC-Rio, o *framework* GeMA (*Geo Modelling Analysis*) é uma biblioteca para suporte do desenvolvimento de simuladores multifísicos. Assim, esse permite aos desenvolvedores concentrarem-se na implementação de uma simulação física, enquanto o GeMA gerencia os dados e funções necessários para o desenvolvimento de programas eficientes. O *framework* foi desenvolvido na linguagem C++, suportando múltiplos paradigmas de simulação e acoplamento, com enfoque principal no método dos elementos finitos.

Á vista disso, os algoritmos apresentados no Capítulo 3 foram implementados no GeMA. Além disso, para o cálculo da matriz de rigidez de cada elemento finito, emprega-se a expressão (3.70). Maiores detalhes sobre a implementação de um algoritmo de mapeamento de retorno em um programa de elementos finitos são apresentados por Simo e Hughes (1998) e Souza Neto, Perić e Owen (2008).

Para solucionar problemas de análise inversa, foi desenvolvido um programa, na linguagem Python 3.6.7 (2018). O programa foi batizado de Minerva e idealizado para que seja possível tratar problemas genéricos de calibração de parâmetros, sendo necessário que o usuário defina o problema a ser resolvido em um arquivo externo, também em linguagem Python. O programa foi desenvolvido com base no exposto no Capítulo 4. A implementação do algoritmo genético, para minimização da equação (4.3), segue o apresentado no Capítulo 5. Devido ao algoritmo de atualização das tensões ter sido desenvolvido na plataforma MATLAB, estabeleceu-se uma comunicação entre o Python e o MATLAB.

O Minerva está disponível no Portal Eras (Lima *et al.*, 2018), que é um ambiente de apoio a pesquisadores e estudantes que trabalham em projetos voltados à geomecânica. O portal é desenvolvido pelo Instituto Tecgraf/PUC-Rio e pode ser acessado em: <https://eras.tecgraf.puc-rio.br/>.

6.1.

Estudo do Algoritmo de Mapeamento de Retorno considerando um Modelo Elastoplástico Perfeitamente Plástico

Considera-se neste estudo apenas o comportamento Perfeitamente Plástico (PP) para o *Cap Model*, ou seja, considerando que o parâmetro $L(\kappa)$ constante ao longo da análise. Dessa forma, esse parâmetro, denominado para esta análise somente por L , se tornou um dado de entrada do programa. Considerar esse comportamento torna o modelo associado e permite que a atualização das tensões seja tratada como um problema de otimização com restrições, conforme descrito por Simo e Hughes (1998). Isto posto, tem-se o seguinte problema de otimização a ser resolvido:

$$\begin{cases} \min_{\sigma_{n+1}} & \frac{1}{2}(\sigma_{n+1}^{trial} - \sigma_{n+1}) : \mathbf{C} : (\sigma_{n+1}^{trial} - \sigma_{n+1}) \\ \text{s. t.} & f_i(\sigma_{n+1}) \leq 0 \end{cases} \quad (6.1)$$

O MATLAB disponibiliza a ferramenta *Optimization Toolbox*, que fornece uma série de funções voltadas à problemas de otimização. Dessa forma, emprega-se esta ferramenta para ajudar a validar o algoritmo desenvolvido. Adota-se o método conhecido por programação quadrática sequencial, que é capaz de

solucionar problemas de otimização com restrições por meio de múltiplas aproximações de problemas de programação quadrática. Emprega-se, como solução inicial para o algoritmo, a tensão preditora. Sendo o *Cap Model* um modelo com múltiplas superfícies de plastificação, $f_i(\sigma_{n+1}) \leq 0$ representa três restrições para o problema.

Os parâmetros empregados nestas análises são apresentados na Tabela 6.1. Além dos parâmetros do modelo constitutivo, define-se o módulo de elasticidade E e o coeficiente de Poisson ν . Para este estudo, são consideradas as deformações totais ϵ_{n+1} apresentadas na Tabela 6.2, considerando que as deformações plásticas ϵ_n^p são nulas.

Tabela 6.1 – Parâmetros adotados para o ensaio de compressão hidrostática

Parâmetro	Valor	Unidade
E	30	MPa
ν	0,3	-
α	200	kPa
β	0,001	kPa^{-1}
λ	100	kPa
θ	0,2	-
R	2,5	-
L	-400	kPa
T	50	kPa

Tabela 6.2 – Deformações totais ϵ_{n+1} empregadas no estudo

Análise	$\epsilon_{xx,n+1}$	$\epsilon_{yy,n+1}$	$\epsilon_{zz,n+1}$	$\epsilon_{xy,n+1}$	$\epsilon_{xz,n+1}$	$\epsilon_{yz,n+1}$
PP-1	0,08	0,08	0,08	0,0866	0,0866	0,0866
PP-2	-0,06	-0,06	-0,06	0,26	0,26	0,26
PP-3	-0,02	-0,02	-0,02	0,26	0,26	0,26
PP-4	-0,12	-0,12	-0,12	0,0866	0,0866	0,0866
PP-5	0,08	0,08	0,08	0,013	0,013	0,013

Os resultados para estas análises são apresentados por intermédio da tensão hidrostática, dada por $p(\sigma) = \frac{1}{3}I_1(\sigma)$, e a tensão equivalente de von Mises, dada por $q(\sigma) = \sqrt{3J_2(\sigma)}$. Assim, calcula-se $p(\sigma_{n+1})$ e $q(\sigma_{n+1})$ empregando algoritmo proposto neste trabalho (denominado Newton-Raphson com busca unidimensional), a versão tradicional do algoritmo de projeção ao ponto mais próximo (denominado de Newton-Raphson sem busca unidimensional) e a

programação quadrática sequencial. Como método de busca unidimensional, adota-se o método da seção áurea. As três metodologias produziram os mesmos resultados, exceto para PP-2 com o Newton-Raphson sem busca unidimensional, já que para esse problema a convergência não foi atingida. A Tabela 6.3 resume as soluções obtidas para as cinco análises.

Tabela 6.3 – Tensões atualizadas para as deformações totais estudadas

Análise	p	q
PP-1	16,67	103,95
PP-2	-207,69	236,83
PP-3	-158,72	258,15
PP-4	-300,57	87,28
PP-5	16,67	103,95

Para entender o impacto da busca unidimensional no algoritmo de projeção ao ponto mais próximo, são apresentados a seguir gráficos que determinam os históricos de correções para cada tipo de algoritmo. O primeiro resultado analisado é exposto nos Gráficos 6.1 e 6.2, referente à PP-2. Destaca-se que esses gráficos e os subsequentes possuem a indicação da ordem que as soluções surgem durante o processo de atualização das tensões, sendo que a solução 1 indica a tensão preditora.

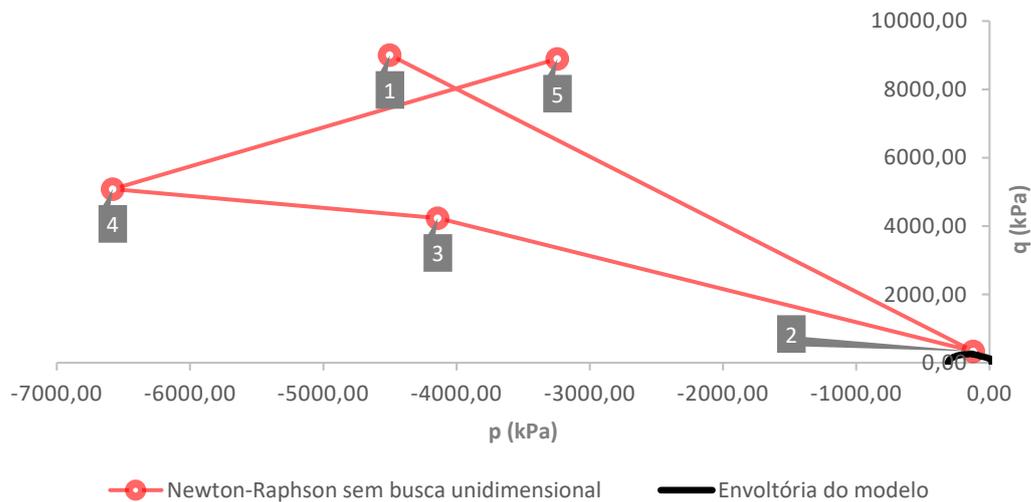


Gráfico 6.1 – Histórico de correções das tensões para PP-2 com o método de Newton-Raphson sem busca unidimensional

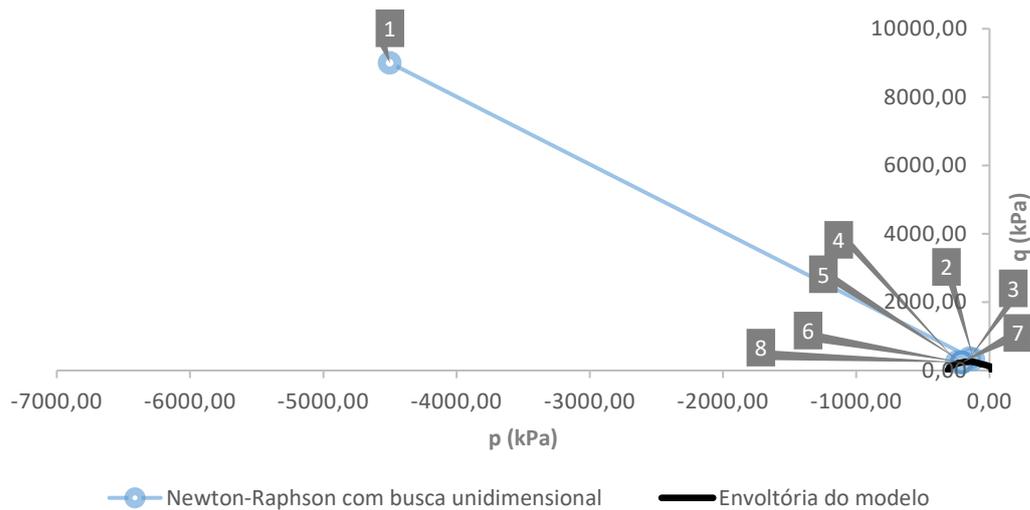


Gráfico 6.2 – Histórico de correções das tensões para PP-2 com o método de Newton-Raphson com busca unidimensional

Observa-se, como já dito anteriormente, que o Newton-Raphson sem busca unidimensional não foi capaz de solucionar esse problema, já que na quinta correção, o algoritmo estagnou. Ao se aproximar da envoltória do *Cap Model*, a solução diverge para um ponto muito distante da superfície. Então, chegou-se em um estado de tensões em que o cálculo de $\Delta^2 \gamma_{n+1}^{(k)}$ sempre produzia valores negativos, sendo impossível atualizar tal estado. A busca unidimensional tornou possível a obtenção de tal resultado. Apresenta-se, no Gráfico 6.3, uma ampliação da solução final do problema.

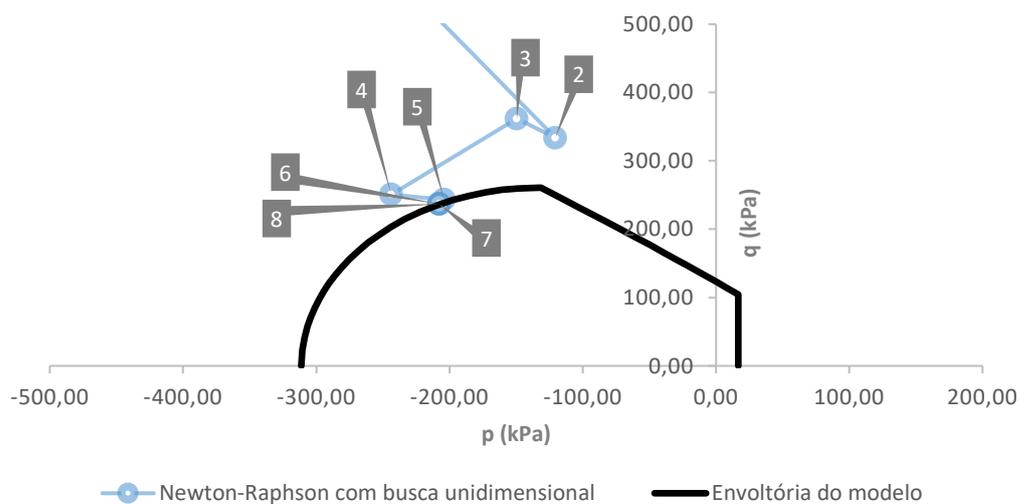


Gráfico 6.3 – Ampliação na solução final do histórico das correções das tensões para PP-2 com o método de Newton-Raphson com busca unidimensional

Considerando a PP-3, as abordagens com e sem busca unidimensional para o método de Newton-Raphson foram bem-sucedidas. Todavia a quantidade de iterações necessárias durante o emprego da busca unidimensional foi menor em relação à abordagem sem busca unidimensional, conforme observado nos Gráficos 6.4 e 6.5. Nos experimentos desenvolvidos ao longo deste trabalho, observou-se que, na maioria das vezes, a abordagem com busca unidimensional itera no máximo a quantidade de vezes da abordagem sem busca unidimensional. Ademais, geralmente, quando a abordagem sem busca unidimensional converge, a abordagem com busca unidimensional também o faz.

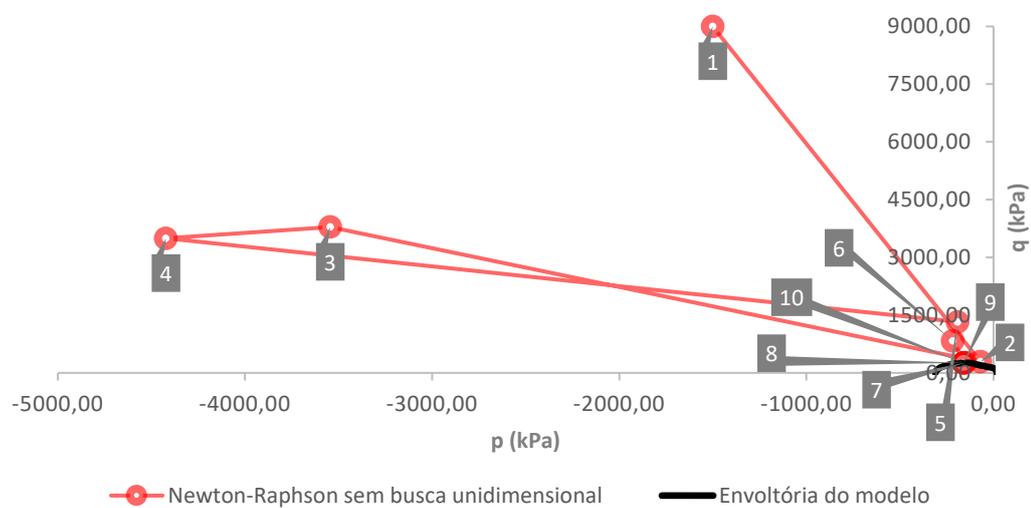


Gráfico 6.4 – Histórico de correções das tensões para PP-3 com o método de Newton-Raphson sem busca unidimensional

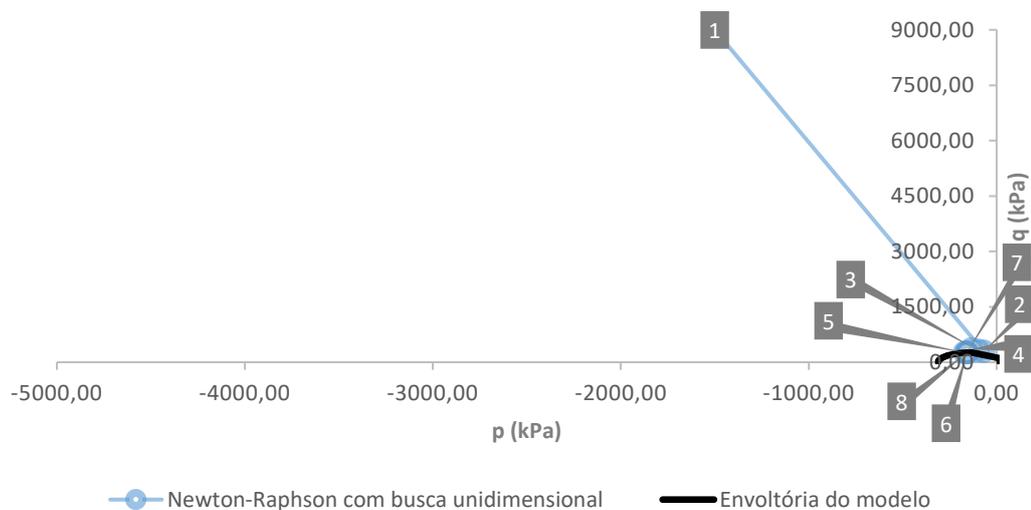


Gráfico 6.5 – Histórico de correções das tensões para PP-3 com o método de Newton-Raphson com busca unidimensional

Outro problema observado durante esta análise é que nas duas primeiras correções para o algoritmo sem busca unidimensional, a solução corrente se aproximava da envoltória do problema. Todavia, subitamente, a solução se afasta da envoltória, para depois retornar em direção à mesma. Isso não ocorre quando se emprega a busca unidimensional, já que o parâmetro $\alpha_{n+1}^{(k)}$ controla a correção das tensões. No Gráfico 6.6, apresenta-se uma ampliação da solução final da PP-3.

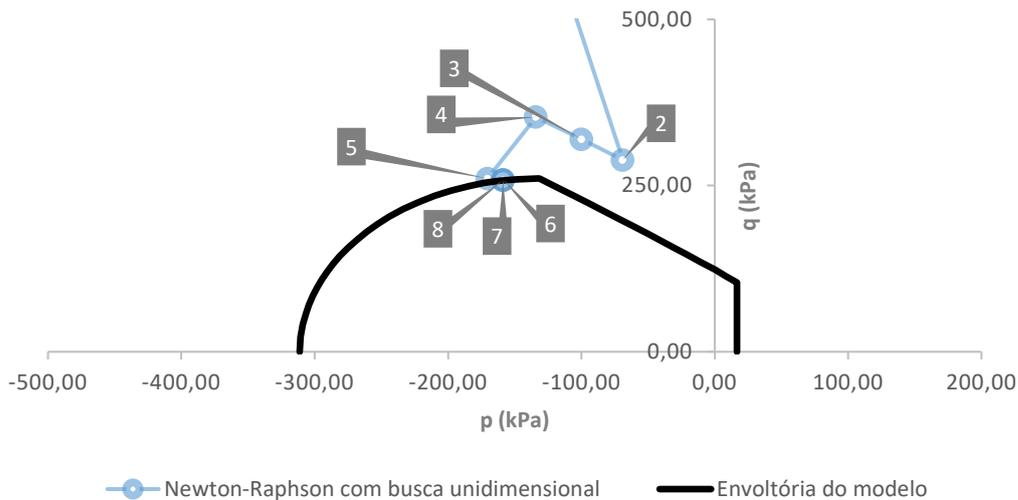


Gráfico 6.6 – Ampliação na solução final do histórico das correções das tensões PP-3 com o método de Newton-Raphson com busca unidimensional

Por fim, apresenta-se os resultados da PP-4, em que as duas abordagens produziram a mesma resposta, por intermédio das mesmas correções. Dessa forma, denota-se ambas as soluções por Newton-Raphson, apresentadas no Gráfico 6.7 e 6.8, que, respectivamente, representam o resumo o histórico de correções calculadas e uma ampliação em torno da solução final. Para todas as iterações, a primeira verificação da regra de Armijo foi atendida, fazendo $\alpha_{n+1}^{(k)} = 1$, dessa forma, o método da seção áurea não foi acionado.

É interessante frisar que estas análises têm o caráter de avaliar a robustez e eficiência do algoritmo proposto, tendo em vista o alto nível de tensões predictoras a serem corrigidas. Em um problema de elementos finitos, tensões intermediárias seriam obtidas a fim de se determinar uma solução global adequada. Todavia, esses testes são interessantes para verificar o quão robusto é o algoritmo apresentado.

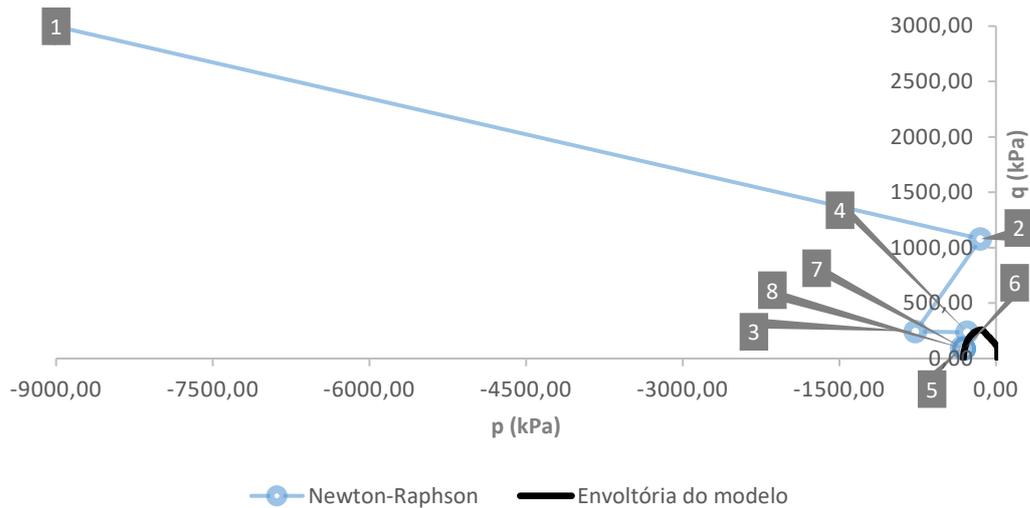


Gráfico 6.7 – Histórico de correções das tensões para PP-4 com o algoritmo de projeção ao ponto mais próximo

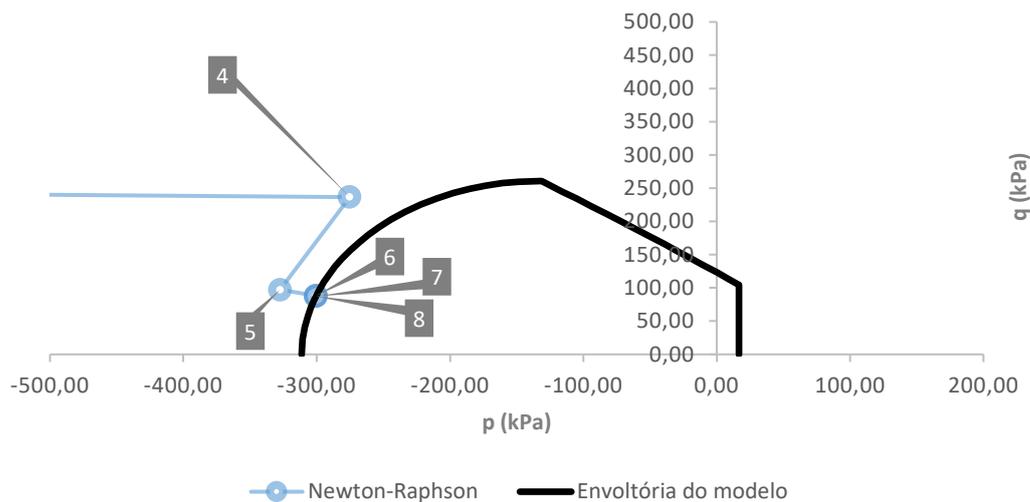


Gráfico 6.8 – Ampliação na solução final do histórico das correções das tensões para PP-4 com o algoritmo de projeção ao ponto mais próximo

Outra nota importante é que, para incrementos de deformações menores, a tendência é que não se tenham grandes problemas para atualizar o estado de tensões. Dessa forma, a fim de observar os comportamentos apresentados, foram adotados incrementos tão grandes. Todavia, isto depende do modelo constitutivo estudado. Scherzinger (2017) observa complicações para convergência mesmo para pequenos incrementos de deformações para o modelo de Hosfold (1972, *apud* Scherzinger, 2017).

6.2.

Estudo do Algoritmo de Mapeamento de Retorno considerando um Modelo Elastoplástico com Endurecimento

Foram realizadas análises considerando o comportamento Elastoplástico com Endurecimento (EE). Para isso, seguiu-se uma metodologia similar ao apresentado na seção anterior. Comparou-se o comportamento do método de Newton-Raphson com e sem busca unidimensional, empregando o método da seção áurea. Para estas análises, não se comparou com a programação quadrática sequencial, já que a formulação apresentada por Simo e Hughes (1998) não trata problemas não associados.

Os dados empregados para o *Cap Model* são apresentados na Tabela 6.4. As deformações totais empregadas em cada análise, são definidas na Tabela 6.5, considerando-se as deformações plásticas iniciais nulas.

Tabela 6.4 – Parâmetros adotados para o ensaio de compressão hidrostática

Parâmetro	Valor	Unidade
E	30	MPa
ν	0,3	-
α	700	kPa
β	0,0005	kPa^{-1}
λ	650	kPa
θ	0	-
R	6	-
D	0,001	kPa^{-1}
W	0,08	-
κ_0	-450	kPa
T	30	kPa

Tabela 6.5 – Deformações totais ϵ_{n+1} empregadas no estudo

Análise	$\epsilon_{xx,n+1}$	$\epsilon_{yy,n+1}$	$\epsilon_{zz,n+1}$	$\epsilon_{xy,n+1}$	$\epsilon_{xz,n+1}$	$\epsilon_{yz,n+1}$
EE-1	-0,05	0,08	-0,05	0,01	0,01	0,01
EE-2	-0,03	0,05	-0,05	-0,001	-0,001	-0,001
EE-3	-0,04	0,01	-0,025	0	0	0

Dentre estas análises, apenas a EE-1 foi bem-sucedida para ambas abordagens. Para as demais, apenas o método de Newton-Raphson com busca

unidimensional foi capaz de solucionar o problema. A Tabela 6.6 sintetiza as soluções para cada um dos problemas.

Tabela 6.6 – Tensões atualizadas para as deformações totais estudadas

Análise	p	q
EE-1	-247,91	234,98
EE-2	-359,30	229,28
EE-3	696,88	154,44

Apresentam-se nos Gráficos 6.9 e 6.10 o histórico de correções que os algoritmos determinam para a análise EE-1. Observa-se, como já dito anteriormente, que apenas a versão do algoritmo de projeção ao ponto mais próximo com busca unidimensional foi capaz de solucionar o problema, já que a implementação sem busca unidimensional entrou em um ciclo infinito de iterações, oscilando entre aproximadamente os mesmos estados de tensão. Encerrou-se a análise na vigésima terceira correção. Similar ao observado por Scherzinger (2017), esse exemplo elucida bem como a busca unidimensional soluciona este problema, que pode vir a ocorrer na implementação sem busca unidimensional.

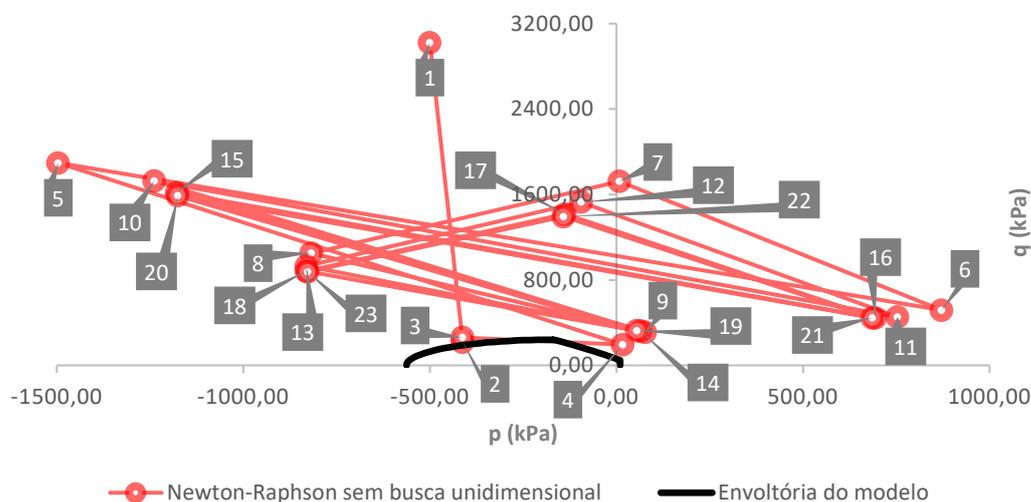


Gráfico 6.9 – Histórico de correções das tensões para EE-1 com o método de Newton-Raphson sem busca unidimensional

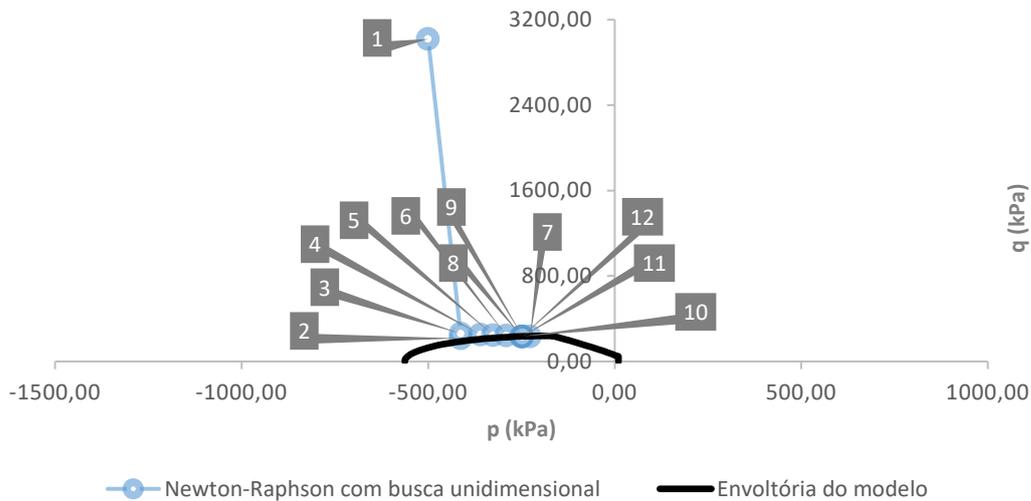


Gráfico 6.10 – Histórico de correções das tensões para EE-1 com o método de Newton-Raphson com busca unidimensional

O próximo exemplo apresentado, referente à EE-2, foi o único solucionado pelas duas abordagens. Todavia, mais iterações foram necessárias para a abordagem sem busca unidimensional, conforme apresentado nos Gráficos 6.11 e 6.12. Mais uma vez, não é uma regra geral que a busca unidimensional reduza a quantidade de iterações do processo.

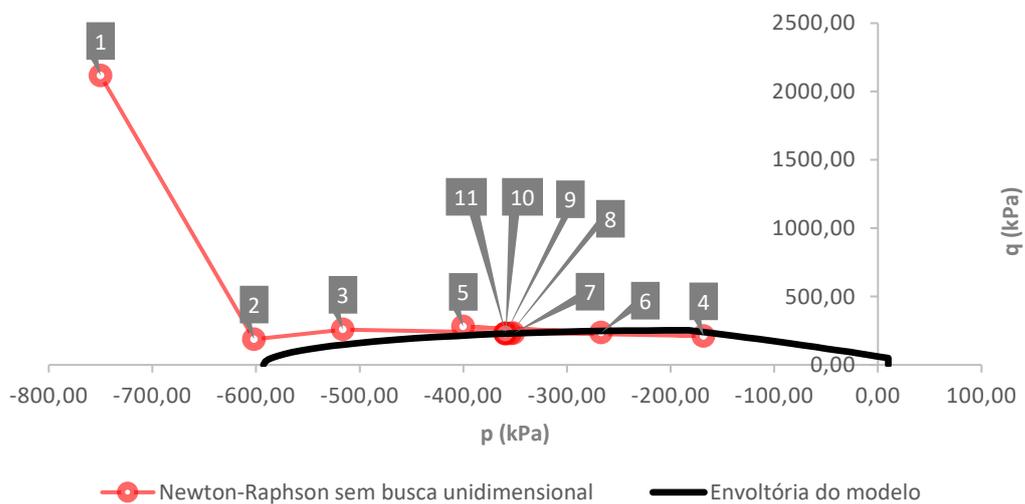


Gráfico 6.11 – Histórico de correções das tensões para EE-2 com o método de Newton-Raphson sem busca unidimensional

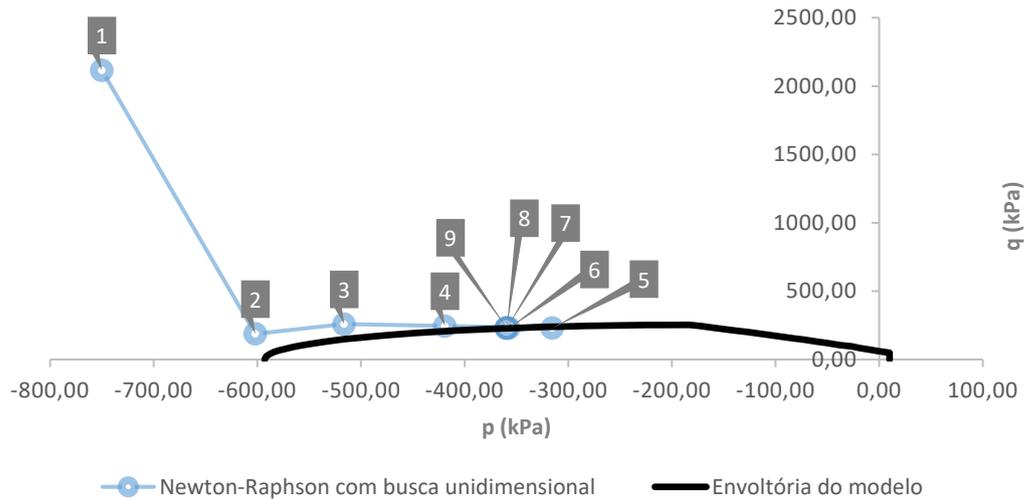


Gráfico 6.12 – Histórico de correções das tensões para EE2 com o método de Newton-Raphson com busca unidimensional

Por fim, a EE-3, o algoritmo sem busca unidimensional não foi mais capaz de corrigir as tensões a partir de um ponto, já que o cálculo de $\Delta^2 \gamma_{n+1}^{(k)}$ sempre produzia valores negativos. Dessa forma, a análise estagnou e teve que ser interrompida na correção 21, conforme apontado no Gráfico 6.13. Quando se empregou a busca unidimensional, foi possível obter uma solução para o problema, conforme apresentado no Gráfico 6.14.

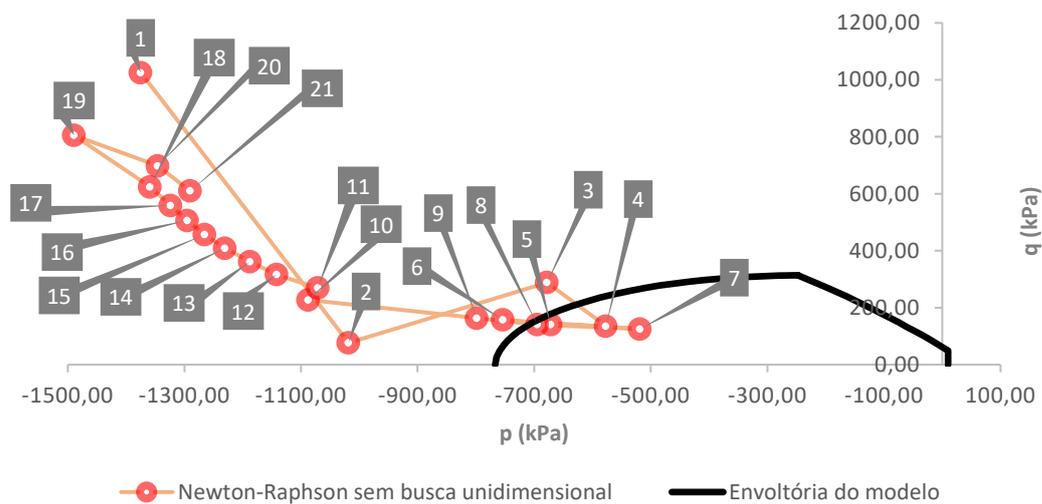


Gráfico 6.13 – Histórico de correções das tensões para EE-3 com o método de Newton-Raphson sem busca unidimensional

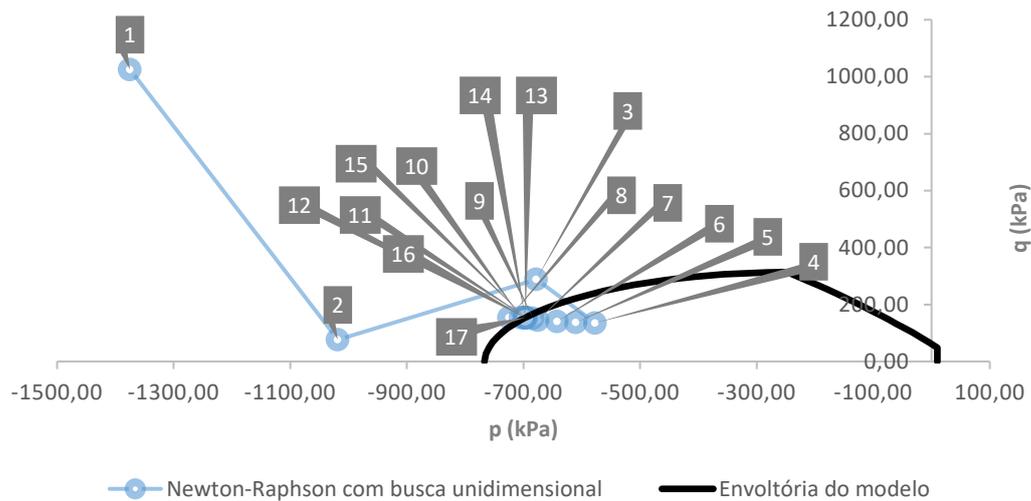


Gráfico 6.14 – Histórico de correções das tensões para EE-3 com o método de Newton-Raphson com busca unidimensional

A partir desses exemplos, observa-se o quão importante é a busca unidimensional para melhorar as características de convergência do algoritmo de projeção ao ponto mais próximo. O fenômeno do endurecimento tornou o problema de plasticidade mais complexo e a busca unidimensional ajudou a estabilizar o algoritmo.

6.3. Ensaio de Compressão Hidrostática

Outra aplicação desenvolvida para validar o algoritmo de mapeamento de retorno, aplicado ao *Cap Model* considerando o endurecimento, é um ensaio numérico de compressão hidrostática, apresentado por Hofstetter, Simo e Taylor (1993). Os parâmetros adotados para esta análise são apresentados na Tabela 6.7, e adaptados corretamente de acordo com a referência original.

Sendo assim, compara-se o resultado apresentado por Hofstetter, Simo e Taylor (1993) com as soluções obtidas pela implementação desenvolvida neste trabalho. Por ser um problema relativamente simples, não se necessita de uma análise de elementos finitos. É essencial apenas que sejam informadas as deformações totais e configurações referentes ao estado anterior. Os resultados são sintetizados no Gráfico 6.15, que relaciona o primeiro invariante do tensor de tensões e a deformação volumétrica, dada por $\varepsilon^v = tr(\boldsymbol{\varepsilon})$. Observa-se uma ótima concordância entre as curvas, sobretudo nos trechos inicial e final da curva.

Tabela 6.7 – Parâmetros adotados para o ensaio de compressão hidrostática

Parâmetro	Valor	Unidade
E	689,4757	MPa
ν	0,25	-
α	1,8616	MPa
β	0,0972	MPa^{-1}
λ	1,1721	MPa
θ	0,02	-
R	2,5	-
D	0,0972	MPa^{-1}
W	0,064	-
κ_0	0,3859	MPa
T	2,0684	MPa

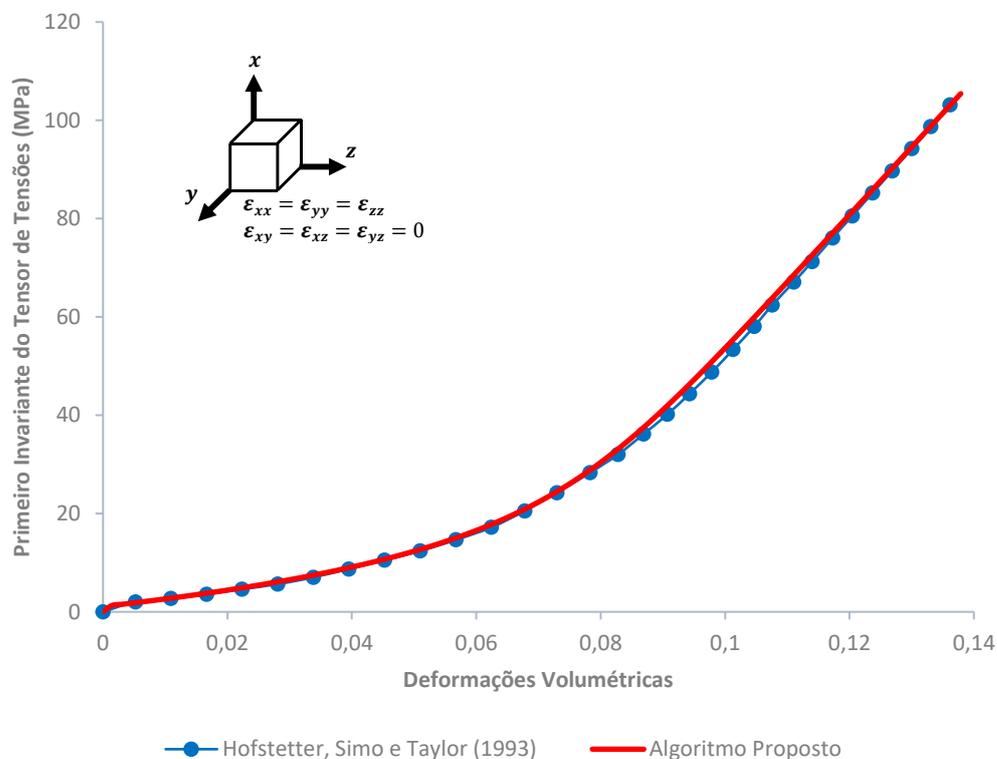


Gráfico 6.15 – Evolução do primeiro invariante do tensor de tensões em relação às deformações volumétrica para o ensaio numérico de compressão hidrostática

Um estudo interessante que pode ser desenvolvido a partir desse exemplo é comparar o esforço computacional entre as metodologias de busca unidimensional apresentadas neste trabalho. Para isso, desenvolveu-se, no GeMA, um modelo de um bloco tridimensional, com dimensões unitárias, em que todas as faces estão submetidas a um deslocamento prescrito de igual intensidade, deixando-o

autoequilibrado. Dessa forma, empregou-se o elemento isoparamétrico quadrático hexaédrico de 20 nós com integração reduzida. Então, realizou-se 5 análises por metodologia e calculou-se o tempo médio. Os tempos relativos de cada análise são apresentados na Tabela 6.8.

Tabela 6.8 – Tempos relativos para cada metodologia de busca unidimensional para o ensaio de compressão hidrostática

Método	Tempo Relativo
Método da Seção Áurea	1
Interpolação Quadrática	1,254604
Interpolação Cúbica	1,316967

Observou-se que todos os métodos produziram o mesmo resultado, todavia o método da seção áurea foi o mais eficiente dentre todos. Uma explicação para isso é que a interpolação quadrática e cúbica são processos aproximados, que não necessariamente retorna à solução exata para o problema de minimização unidimensional. Dessa forma, a qualidade da direção (3.85) gerada por uma solução aproximada não se iguala à direção produzida para a solução exata do problema unidimensional. Sendo assim, o emprego de tal aproximação necessita uma quantidade de iterações maior para atingir a convergência.

6.4. Ensaio Edométrico

Sandler e Rubin (1979) apresentam uma implementação específica para uma versão do *Cap Model*. Os autores aplicam o algoritmo em um ensaio edométrico de uma areia (*McCormick Ranch Sand*) ensaiada por DiMaggio e Sandler (1971, *apud* Sandler e Rubin, 1979). Os parâmetros desse material foram calibrados para o *Cap Model* a partir de um processo de otimização pelos mesmo autores. Apresenta-se a seguir os parâmetros do modelo, adaptados para o modelo descrito na Seção 2.2.

Tabela 6.9 – Parâmetros adotados para o ensaio de edométrico

Parâmetro	Valor	Unidade
E	689,4757	MPa
ν	0,25	-
α	2,4377	MPa
β	0,0972	MPa ⁻¹
λ	1,7551	MPa
θ	0	-
R	1,77	-
D	0,0972	MPa ⁻¹
W	0,064	-
κ_0	0,9173	MPa
T	2,0684	MPa

Utilizam-se as deformações apresentadas por Sandler e Rubin (1979) como dados de entrada para o algoritmo. Assim, o Gráfico 6.16 expõe o resultado obtido pela implementação desenvolvida comparada à solução original.

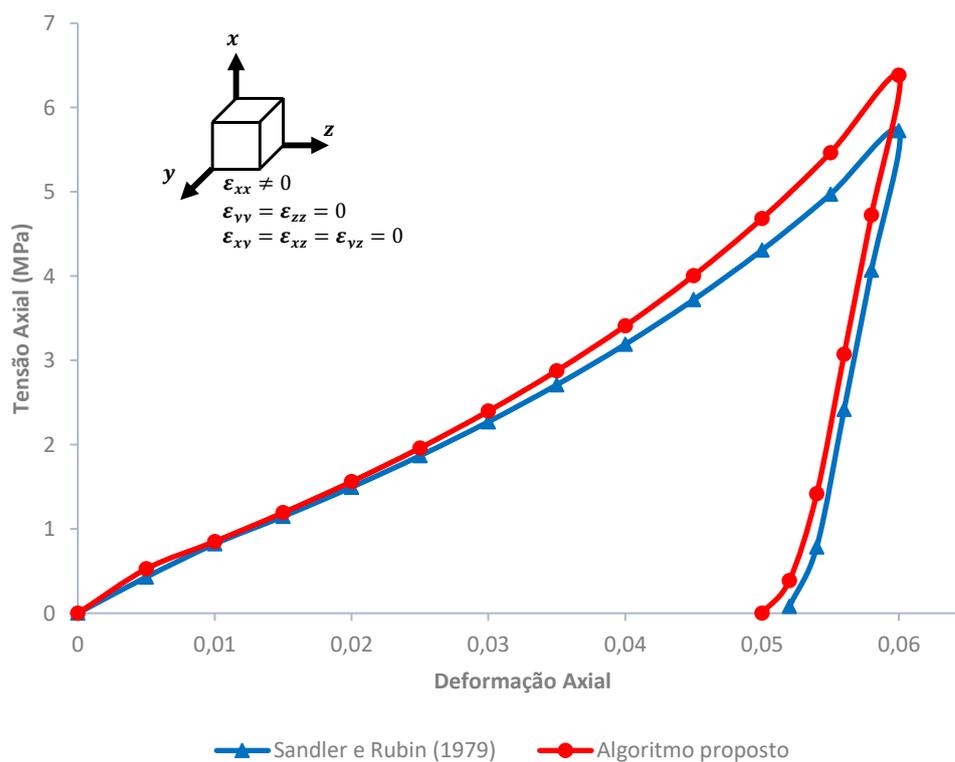


Gráfico 6.16 – Evolução da tensão axial em relação às deformações axiais para o ensaio numérico edométrico

Observa-se que o comportamento geral da solução está correto, contudo esta sofreu um afastamento da referência original. Uma possível justificativa para esta diferença é a formulação dos algoritmos de mapeamento de retorno, que são totalmente distintos. Além disso, Sandler e Rubin (1979) empregam uma lei de endurecimento similar a empregada neste trabalho, mas com algumas considerações adicionais para evitar, por exemplo, comportamento de amolecimento.

Observa-se que o comportamento final das curvas apresentadas no Gráfico 6.16 não é linear, que a princípio não seria esperado, pois imagina-se que o descarregamento ocorreria totalmente no domínio elástico. Todavia, construindo a trajetória de tensões no plano p-q, observa-se o porquê disto. O gráfico 6.17 apresenta a configuração inicial e final do *Cap Model*, devido ao fenômeno do endurecimento, representado no plano p-q.

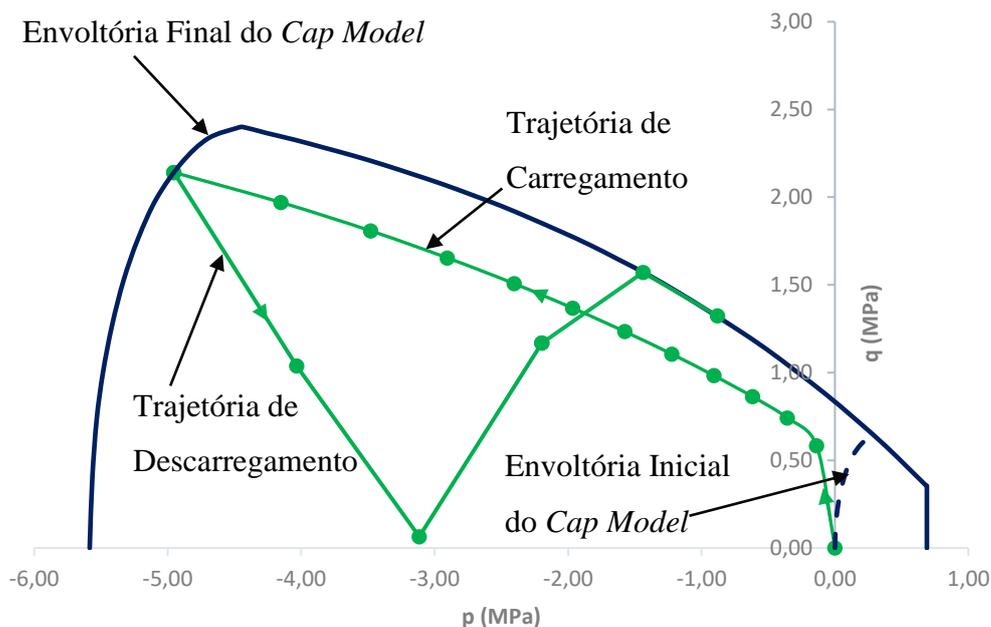


Gráfico 6.17 – Trajetória de tensões para o ensaio numérico edométrico

Enquanto ocorre acréscimo de deformações totais nesse ensaio, o *cap* elíptico expande devido ao endurecimento. Durante o processo de descarregamento, o estado de tensões retorna para dentro da envoltória do modelo. É importante frisar que o eixo p representa um eixo de simetria da envoltória e por isso, a trajetória de

tensões é refletida nesse e volta a tocar novamente na envoltória, gerando plastificação e conseqüente comportamento não linear.

Para esse modelo, durante o processo de busca unidimensional, a primeira verificação da regra de Armijo foi atendida em todas as iterações. Dessa forma, tem-se que $\alpha_{n+1}^{(k)} = 1$, não sendo possível comparar as metodologias para cálculo desse parâmetro.

6.5. Modelagem em Elementos Finitos de uma Sapata Flexível

Para avaliar o algoritmo apresentado por Hofstetter, Simo e Taylor (1993) para o *Cap Model* com lei de endurecimento associada e não associada, os autores analisaram o problema de uma sapata flexível. Considera-se o estado plano de deformações e uma malha regular com 72 elementos finitos, conforme apresentado na Figura 6.1. O mesmo modelo foi idealizado no GeMA, considerando o elemento finito isoparamétrico, bilinear, quadrilateral e de quatro nós. Os autores também empregam um elemento quadrático de nove nós, todavia a diferença entre as duas soluções é muito pequena.

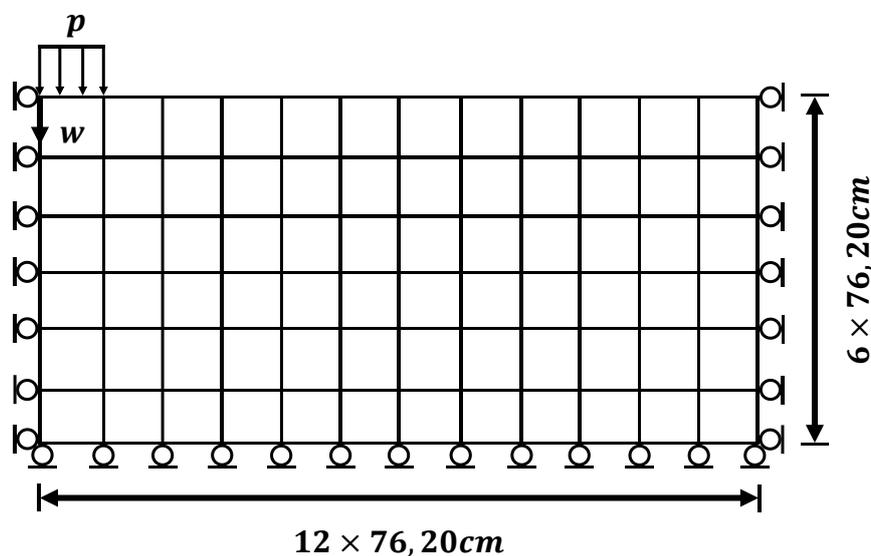


Figura 6.1 – Sapata flexível

Os parâmetros empregados são apresentados na Tabela 6.9. Assim sendo, realizou-se a análise desse problema via elementos finitos isoparamétricos

bilineares de quatro nós (Q4) e quadráticos de oito nós (Q8), considerando respectivamente, integração completa e reduzida.

Apresenta-se, no Gráfico 6.18, as curvas que relacionam as pressões p e os deslocamentos w , conforme indicado na Figura 6.1. Observa-se uma excelente concordância entre as soluções para o elemento Q4 obtidas pelo GeMA e pelos autores originais. Ainda se nota que a solução obtida para o elemento Q4 é mais rígida que a solução obtida pelo elemento Q8, como esperado. Para esse problema, a primeira verificação da regra de Armijo foi atendida em todas as iterações.

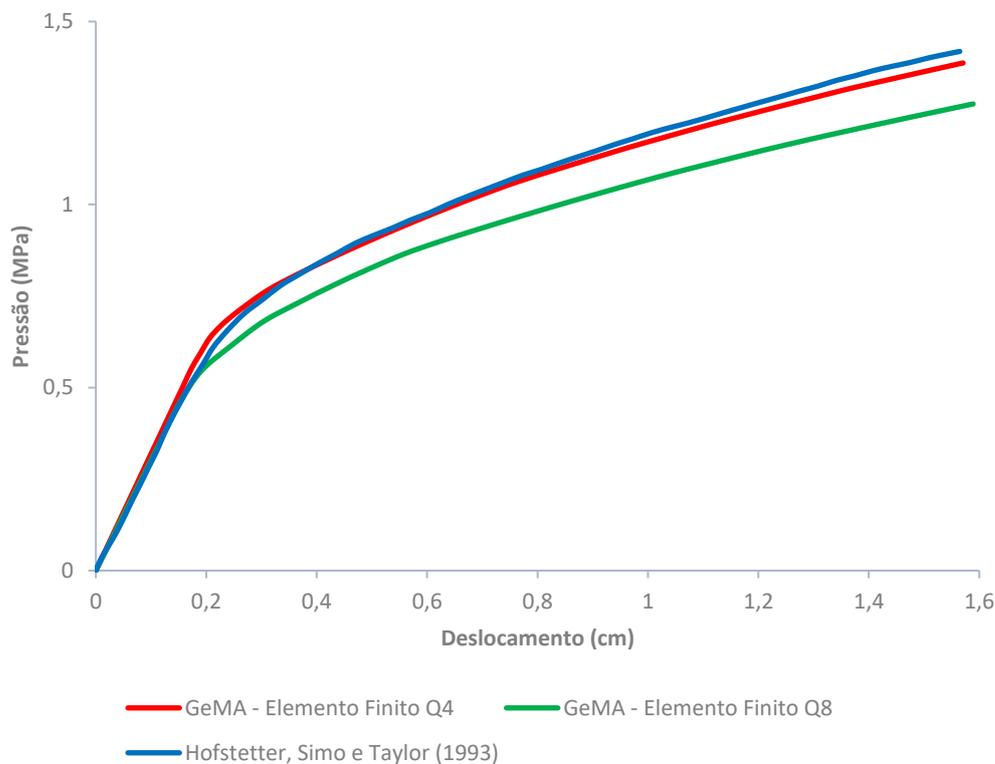


Gráfico 6.18 – Curva pressão-deslocamento para a sapata flexível

6.6. Calibração de Parâmetros do *Cap Model* a partir de Dados Experimentais

Para aplicar a metodologia exposta nos Capítulos 4 e 5, considerou-se um conjunto de dados experimentais apresentados por Salami (1986). Os ensaios foram desenvolvidos na Universidade do Arizona para a proposição de um modelo constitutivo adequado para concreto e rochas. Assim sendo, ensaios de compressão foram realizados em corpos de prova de concreto e pedra-sabão.

Neste trabalho, foram selecionados alguns desses ensaios em corpos de prova de concreto para calibrar os parâmetros do *Cap Model*. Nesses ensaios, carregamentos multiaxiais são aplicados, e como resultados, as deformações principais e máximas tensões principais ou tensões octaédricas de cisalhamento foram registradas graficamente. A convenção adotada é que a compressão é representada por valores positivos.

Para realizar a calibração, os dados observados (máximas tensões principais ou tensões octaédricas de cisalhamento apresentados pelo autor) são comparados, por meio das equações (4.3), (4.7) e (4.8), com os dados previstos pelo modelo constitutivo. Por simplificação, os pesos w_i definidos na equação (4.3) são iguais à unidade e a matriz W_i em (4.7) é igual à identidade. Dessa forma, emprega-se as deformações principais apresentadas pelo autor para definir as deformações totais, necessárias para o algoritmo de mapeamento de retorno. Portanto, considerando as deformações principais ε_1 , ε_2 e ε_3 , constrói-se o tensor de deformações a partir de: $\varepsilon_{xx} = \varepsilon_1$, $\varepsilon_{yy} = \varepsilon_2$, $\varepsilon_{zz} = \varepsilon_3$, $\varepsilon_{xy} = 0$, $\varepsilon_{xz} = 0$ e $\varepsilon_{yz} = 0$.

Isto posto, empregou-se o programa Minerva para realizar tal calibração, considerando os dados experimentais mencionados. Foram selecionados 12 resultados de ensaios, listados na Tabela 6.10. Maiores detalhes sobre as características do ensaio são descritos por Salami (1986).

Tabela 6.10 – Ensaios de Salami (1986) empregados na calibração

Ensaio	Quantidade
Compressão Triaxial Convencional (<i>Conventional Triaxial Compression</i>)	4
Extensão Triaxial Reduzida (<i>Reduced Triaxial Extension</i>)	1
Compressão Triaxial (<i>Triaxial Compression</i>)	2
Cisalhamento Simples (<i>Simple Shear</i>)	1
Extensão Triaxial (<i>Triaxial Extension</i>)	1
Carregamento Proporcional (<i>Proportional Loading</i>)	1
Trajatória Circular de Tensões (<i>Circular Stress Path</i>)	1
Trajatória Arbitrária de Tensões (<i>Arbitrary Stress Path</i>)	1

Por meio desses ensaios, o autor determinou as propriedades elásticas do material, que estão descritas na Tabela 6.11 e empregada no processo de calibração para reduzir o número de incógnitas do problema. O algoritmo genético necessita de intervalos para cada parâmetro. Dessa forma, adotou-se a ordem de grandeza dos resultados de calibração do *Cap Model* realizada por Hofstetter, Simo e Taylor

(1993) para um concreto de características similares, para definir o domínio de pesquisa do algoritmo. Este domínio é descrito na Tabela 6.12.

Tabela 6.11 – Parâmetros elásticos para o concreto ensaiado por Salami (1986)

Parâmetro	Valor	Unidade
E	6983,4	MPa
ν	0,154	-

Tabela 6.12 – Domínio dos parâmetros do *Cap Model* para o concreto ensaiado por Salami (1986)

Parâmetro	Limite Inferior	Limite Superior	Unidade
α	0	70	MPa
β	0	0,145	MPa^{-1}
λ	0	70	MPa
θ	0	1	-
R	1	5	-
D	0	0,0145	MPa^{-1}
W	0	1	-
κ_0	-130	0	MPa
T	0	7	MPa

Todas as análises foram realizadas com o número fixo de 100 gerações. Para evitar que o programa entrasse em um ciclo infinito de iterações, caso o algoritmo de mapeamento de retorno não fosse capaz de atualizar as tensões para uma determinada deformação total, limitou-se o número de iterações deste em 1000. Se esse número é superado, o programa retorna um erro exageradamente grande para esse conjunto de parâmetros. Esta abordagem se assemelha ao método de pena de morte, descrito por Michalewicz e Schoenauer (1996), para problemas de otimização com restrições, em que se atribui um valor muito alto para o valor da função objetivo sempre que uma restrição é violada.

Por fim, os resultados produzidos pelo programa Minerva são descritos a seguir na Tabela 6.13. Para gerar tais resultados, adotam-se os parâmetros descritos na Tabela 5.4, o cruzamento aritmético e a mutação não uniforme. Portanto, três probabilidades de mutação p_m são avaliadas. Para cada uma destas probabilidades, cinco análises foram realizadas. Além dos parâmetros ótimos obtidos pelo programa, apresenta-se o valor de E^t , calculado pela função erro (4.3).

Tabela 6.13 – Parâmetros obtidos após a calibração

p_m	Solução	α	β	λ	θ	R	D	W	κ_0	T	E^t
0,05	PM05-1	36,09	0,00014	22,08	0,219	1,00	0,00024	0,200	-0,09	2,13	0,16104
	PM05-2	14,15	0,13788	0,01	0,217	1,00	0,00033	0,146	0,00	2,08	0,16097
	PM05-3	22,09	0,03472	7,56	0,126	1,07	0,00322	0,017	-0,04	2,51	0,16264
	PM05-4	14,79	0,00211	0,13	0,177	1,00	0,00610	0,010	0,00	3,11	0,15919
	PM05-5	20,06	0,05590	5,17	0,121	1,02	0,00961	0,007	-0,16	3,78	0,16043
0,10	PM10-1	50,90	0,00051	36,62	0,197	1,00	0,00113	0,044	-0,03	2,34	0,16063
	PM10-2	14,59	0,13870	0,00	0,214	1,00	0,00026	0,189	-0,05	4,98	0,16140
	PM10-3	52,53	0,01166	38,46	0,001	1,83	0,00484	0,012	-0,07	0,97	0,16619
	PM10-4	30,49	0,00444	16,02	0,215	1,63	0,00162	0,033	-0,14	0,29	0,16481
	PM10-5	14,13	0,11000	0,03	0,218	1,00	0,00149	0,034	0,00	2,47	0,16044
0,15	PM15-1	32,40	0,13752	23,46	0,152	1,89	0,00318	0,021	-1,87	0,06	0,16931
	PM15-2	17,14	0,00118	3,21	0,219	1,00	0,00018	0,261	0,00	2,16	0,16103
	PM15-3	24,54	0,13683	10,50	0,113	1,30	0,01012	0,008	-0,23	3,07	0,16337
	PM15-4	41,18	0,00011	26,86	0,205	1,00	0,00228	0,023	-0,13	3,21	0,16027
	PM15-5	24,18	0,01652	9,49	0,201	1,26	0,00015	0,347	-0,06	5,37	0,16460

Observa-se que as soluções obtidas são totalmente distintas entre si. Isso indica que o problema tem múltiplas soluções. As melhores soluções, para cada probabilidade de mutação, são apresentadas na tabela a seguir, levando em consideração os menores valores de E^t para tal definição.

Tabela 6.14 – Melhores parâmetros obtidos após a calibração

p_m	Solução	α	β	λ	θ	R	D	W	κ_0	T	E^t
0,05	PM05-4	14,79	0,00211	0,13	0,177	1,00	0,00610	0,010	0,00	3,11	0,15919
0,1	PM10-5	14,13	0,11000	0,03	0,218	1,00	0,00149	0,034	0,00	2,47	0,16044
0,15	PM15-4	41,18	0,00011	26,86	0,205	1,00	0,00228	0,023	-0,13	3,21	0,16027

Os erros calculados para os parâmetros descritos na Tabela 6.14 são muito próximos entre si, podendo-se concluir que não houve grande impacto da probabilidade de mutação na solução final. Para compreender como se dá a evolução do algoritmo ao longo das gerações, avalia-se a evolução do erro ao longo de todas as gerações, para cada análise a partir dos Gráficos 6.19 a 6.21. Esta evolução se refere ao melhor indivíduo gerado em cada geração.

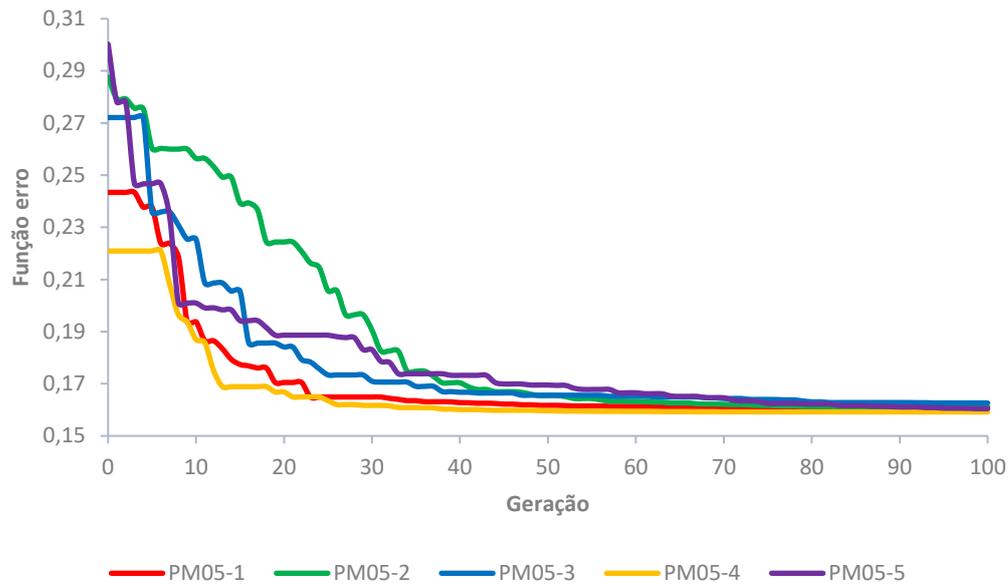


Gráfico 6.19 – Evolução do erro ao longo das análises para as soluções obtidas para $p_m = 0,05$

Observa-se no Gráfico 6.19 que nas primeiras gerações, a diminuição do erro se dá de uma forma mais brusca. A partir de um certo ponto, nota-se a estabilização do erro, com a queda na taxa de redução do erro. Este fato é observado nas demais probabilidades de mutação. Dessa forma, em torno de 50 gerações seriam necessárias para obter uma solução adequada para o problema, reduzindo o tempo de análise pela metade. Assim, se fosse considerado o critério de parada, conforme descrito na Seção 5.5, o algoritmo retornaria à solução final mais rapidamente.

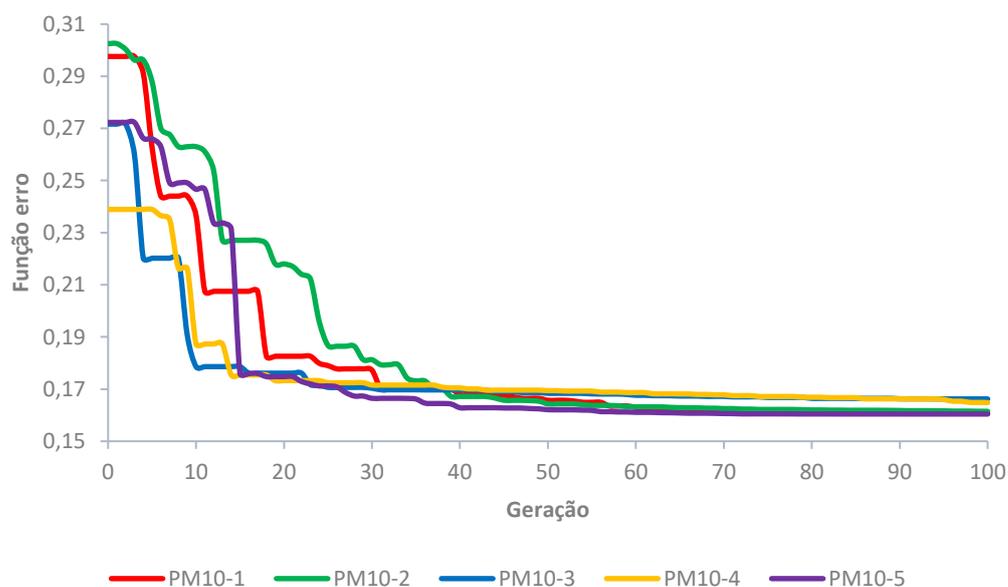


Gráfico 6.20 – Evolução do erro ao longo das análises para as soluções obtidas para $p_m = 0,1$

Diferentemente do Gráfico 6.19, cuja melhor solução, a PM05-4, atingiu mais rápido o ponto que estabiliza a solução, para o problema com $p_m = 0,1$, isso não ocorre. No Gráfico 6.20, observa-se que a solução PM10-3 atinge a estabilização da medida de erro antes das demais. Todavia, a melhor solução obtida foi a PM10-5, que levou algumas gerações a mais para tal. Portanto, conclui-se que a queda rápida da medida de erro não indica necessariamente que a solução final será a mais adequada.

Por fim, observa-se os resultados para $p_m = 0,15$, expostos no Gráfico 6.21. Nota-se novamente que a solução que mais rapidamente chega à estabilização, nesse caso a PM15-5, não necessariamente é a solução mais adequada. Para esta probabilidade de mutação, a PM15-4 produziu a melhor solução. Ademais, ainda para a solução PM15-5, o melhor indivíduo gerado inicialmente foi muito melhor que os das outras soluções, o que não foi capaz de tornar essa a melhor solução dentre as outras.

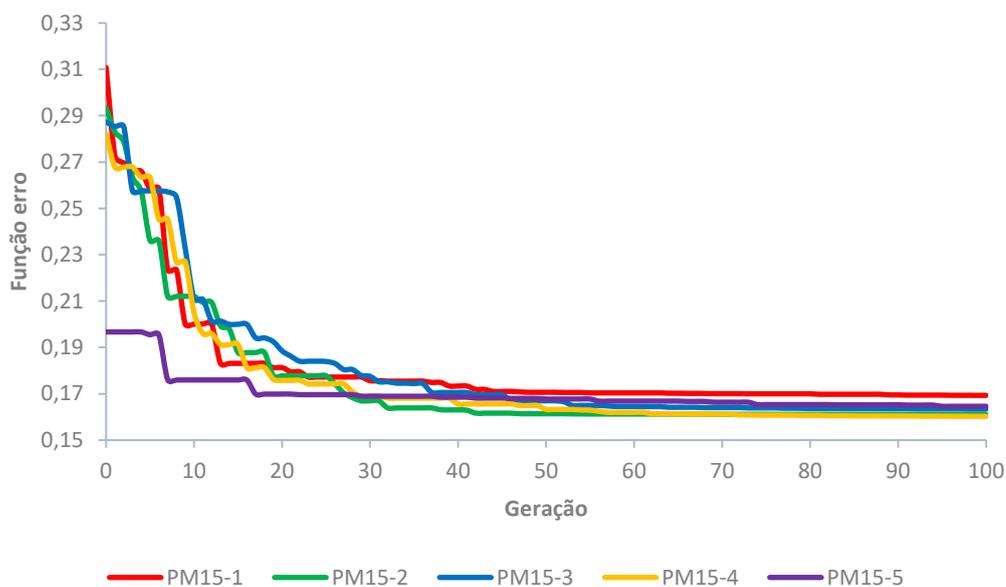


Gráfico 6.21 – Evolução do erro ao longo das análises para as soluções obtidas para $p_m = 0,15$

Por fim, apresenta-se o Gráfico 6.22 que compara os erros obtidos ao longo das análises, considerando apenas as melhores soluções para cada probabilidade de mutação: PM05-4, PM10-5 e PM15-4. Observa-se graficamente que a solução PM05-4 produziu o melhor erro, e conseqüentemente o melhor ajuste para o problema de calibração em questão.

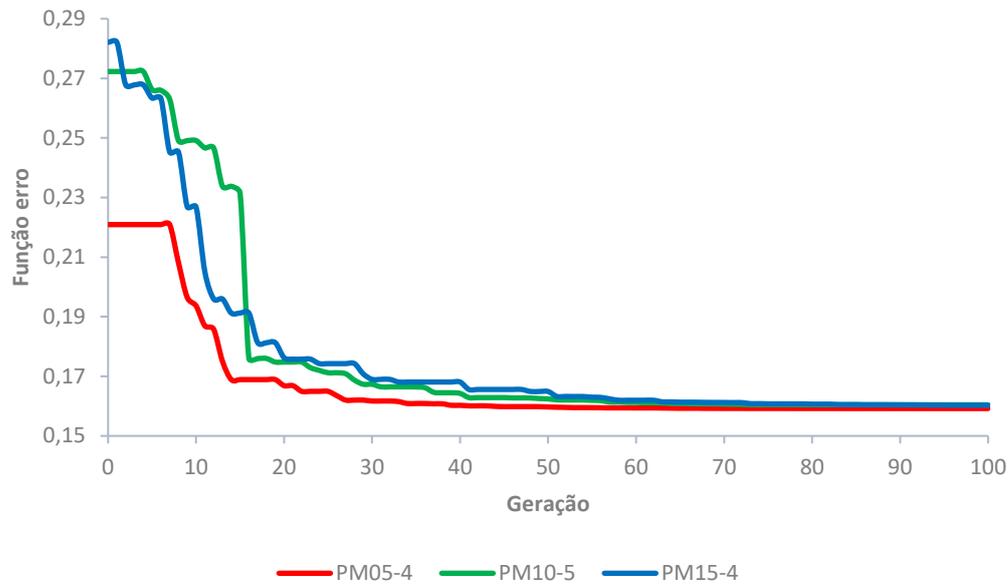


Gráfico 6.22 – Evolução do erro ao longo das análises para as melhores soluções

Considerando a solução PM05-4, considerada aqui como a melhor solução, o Apêndice C apresenta as comparações entre as curvas tensão-deformação experimentais e as obtidas no processo de calibração. Destaca-se que não se pode concluir que $p_m = 0,05$ é o melhor parâmetro para solucionar este problema, já que a quantidade de análises (cinco) é insuficiente.

É importante frisar que os dados fornecidos por Salami (1986) são preponderantemente compressivos, dessa forma os dados existentes podem não ser suficientes para definir a *cut-off* de tração. Em um problema real, é importante que os dados utilizados na calibração sejam significativos. Por isso, para calibrar os parâmetros de um modelo constitutivo de forma mais eficaz, é importante uma grande quantidade de informação que abranja um domínio de tensões bem variado.

7

Conclusões e Sugestões para Trabalhos Futuros

Apresentam-se aqui as conclusões obtidas durante o desenvolvimento deste trabalho e as sugestões para trabalhos futuros acerca dos problemas de plasticidade computacional para múltiplas superfícies de plastificação e calibração de parâmetros.

7.1. Conclusões

Neste trabalho, abordou-se a implementação de algoritmos aplicados à modelos constitutivos com múltiplas superfícies de plastificação para integração das equações de evolução elastoplástica e calibração de parâmetros. Para o algoritmo de integração, adotou-se o método de Newton-Raphson com busca unidimensional. Já para o problema de calibração de parâmetros, soluciona-se o problema inverso a partir de um algoritmo genético. Baseado nos conceitos básicos e formulações apresentados, tais implementações foram desenvolvidas para aplicá-las em determinados problemas.

Antes de aplicar em um problema de calibração de parâmetros, desenvolveu-se um estudo paramétrico dos parâmetros específicos do algoritmo genético implementado neste trabalho. Observou-se que o parâmetro mais impactante na solução obtida é a probabilidade de mutação. Não foi possível indicar um valor único para esta probabilidade, pois essa é muito dependente do tipo de problema.

Para avaliar o algoritmo de mapeamento de retorno, foram desenvolvidas algumas aplicações locais (ponto de Gauss) e três problemas com solução conhecida. Para os problemas locais, analisou-se o modelo sem e com o endurecimento. Para o primeiro caso, comparou-se o algoritmo proposto com a versão tradicional do algoritmo de projeção ao ponto mais próximo e, também, com a programação quadrática sequencial. Todos os métodos retornaram o mesmo resultado, quando a convergência foi alcançada. Quando se compara a

implementação proposta com a tradicional, verifica-se que a busca unidimensional melhora as características de convergência do algoritmo.

Os problemas com solução conhecida foram: um ensaio de compressão hidrostática, um ensaio edométrico e uma sapata flexível. As três análises tiveram boa concordância com a solução apresentada na literatura, sobretudo o ensaio de compressão hidrostática e a sapata flexível.

Uma análise realizada para o ensaio de compressão hidrostática foi a comparação entre as abordagens de busca unidimensional apresentadas neste trabalho. Portanto, o método da seção áurea, interpolação quadrática e interpolação cúbica foram confrontadas diretamente em termos de eficiência computacional, no *framework* GeMA. O método da seção áurea foi a abordagem mais eficiente, já que esta é um método de busca exato, produzindo soluções de alta qualidade a cada iteração.

Para os problemas de calibração de parâmetros, o exemplo proposto foi calibrar o *Cap Model*, descrito na Seção 2.2, a partir de dados experimentais fornecidos por Salami (1986). Tais dados consistem em uma série de ensaios de compressão em amostras de concreto. Dessa forma, sendo o módulo de elasticidade e coeficiente de Poisson definidos pelo autor, nove parâmetros foram determinados por meio da metodologia de calibração de parâmetros.

A partir dos parâmetros indicados na Tabela 5.4, analisou-se três grupos de calibrações com cinco resultados cada. A melhor solução foi obtida quando a probabilidade de mutação foi igual a 0,05. Todavia não se pode concluir que este é o melhor parâmetro para este problema, já que a quantidade de análises realizadas é relativamente pequena.

Destaca-se que os dados apresentados por Salami (1986) podem não ser suficientes para definir o limitante à tração da envoltória, já que as tensões envolvidas nos ensaios são preponderantemente compressivas. É essencial que os dados utilizados na calibração originem informações importantes para a calibração.

7.2. Sugestões para Trabalhos Futuros

A respeito dos estudos da integração das equações de evolução elastoplásticas, sugere-se estudar outros algoritmos que possam resolver esse tipo

de problema. Algumas sugestões são: Programação Quadrática Sequencial, Método dos Pontos Interiores, Método de Região de Confiança e Método Quase-Newton BFGS.

Devido às diversas formas de representar matematicamente um critério de plastificação, é possível tratar as superfícies clássicas de Tresca e Mohr-Coulomb como critérios com múltiplas superfícies de plastificação, já que esses são formados por diversos planos no espaço das tensões principais. Portanto, é interessante avaliar o algoritmo desenvolvido para tais problemas.

Com respeito às aplicações da plasticidade computacional desenvolvidas neste trabalho, aponta-se que a comparação entre os métodos de busca unidimensional apresentados neste trabalho se resumiu apenas ao ensaio numérico de compressão hidrostática. Uma sugestão é desenvolver outros problemas para que tal comparação possa ser feita de forma mais aprofundada.

Acerca dos estudos da análise inversa, sugere-se, modificar os parâmetros empregados nos operadores de cruzamento e mutação do algoritmo genético implementado. É interessante também considerar modificações no algoritmo genético apresentado. Sugere-se acrescentar uma metodologia de busca local no algoritmo genético, implementada logo após incluir as mutações na população, conforme descrito por Sawyerr, Adewumi e Ali (2014) e Jin *et al.* (2017). Outra modificação sugerida é a implementação híbrida, descrita por Jin *et al.* (2016) e Jin *et al.* (2017), que consiste em utilizar um segundo operador de cruzamento, mediante uma segunda probabilidade de cruzamento, o que torna o algoritmo mais eficaz, segundo estes autores.

Outros algoritmos de otimização podem ser estudados para solucionar os problemas inversos, como a otimização por enxames de partícula, arrefecimento simulado (conhecido em inglês como *simulated annealing*), algoritmo de evolução diferencial e colônia artificial de abelhas, descritos de forma sucinta por Yin (2017). Uma abordagem diferente de implementação de um algoritmo genético é o algoritmo microgenético, avaliado por Rokouzzaman e Sakai (2010) para calibração dos parâmetros para um modelo elastoplástico.

Tratar o problema de calibração de múltiplos dados por intermédio da otimização multiobjetivo considerando a fronteira de Pareto, como descrito por Papon *et al.* (2012) e Jiang *et al.* (2018), também uma abordagem interessante para que seja comparada à otimização mono-objetivo.

Em relação às análises desenvolvidas neste trabalho, sugere-se avaliar valores adequados para os pesos w_i e a matriz \mathbf{W}_i , empregadas, respectivamente, em (4.3) e (4.7). Devido à quantidade diferente de pontos para cada gráfico apresentado no Apêndice C, pesos iguais podem privilegiar certos resultados em detrimento de outros. Dessa forma, é válido um estudo acerca destes parâmetros.

ADHIKARY, D. P.; JAYASUNDARA, C. T.; PODGORNEY, R. K.; WILKINS, A. H. **A robust return-map algorithm for general multisurface plasticity.** International Journal for Numerical Methods in Engineering, v. 109, n. 2, p. 218–234, 2017.

ALI, M. M.; KHOMPATRAPORN, C.; ZABINSKY, Z. B. **A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems.** Journal of Global Optimization, v. 31, n. 4, p. 635–672, 2005.

ARAI, K.; OHTA, H.; KOJIMA, K.; WAKASUGI, M. **Application of back-analysis to several test embankments on soft clay deposits.** Soils and Foundations, v. 26, n. 2, p. 60–72, 1986.

ARAI, K.; OHTA, H.; YASUI, T. **Simple optimization techniques for evaluating deformation moduli from field observations.** Soils and Foundations, v. 23, n. 1, p. 107–113, 1983.

ARMIJO, L. **Minimization of functions having Lipschitz continuous first partial derivatives.** Pacific Journal of Mathematics, v. 16, n. 1, p. 1–3, 1966.

ARORA, J. S. **Introduction to optimum design.** 3. ed. Waltham: Elsevier, 2012.

BÄCK, T. **Selective pressure in evolutionary algorithms: A Characterization of Selection Mechanisms.** In: IEEE World Congress on Computational Intelligence, 1994, Orlando. Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE, 1994, p. 57-62.

BAKSALARY, J. K.; STYAN, G. P. H. **Generalized inverses of partitioned matrices in Banachiewicz–Schur form.** Linear Algebra and its Applications, v. 354, n. 1–3, p. 41–47, 2002.

BRUNO, H.; BARROS, G.; MARTHA, L. F.; MENEZES, I. F. M de. **Formulação e solução de problemas de retorno à superfície de escoamento via programação cônica.** In: XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering, 2017, Florianópolis. Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering. Florianópolis: ABMEC, 2017.

BULJAK, V. **Inverse analyses with model reduction: Proper orthogonal decomposition in structural mechanics.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

CALVELLO, M.; FINNO, R. J. **Selecting parameters to optimize in model calibration by inverse analysis.** Computers and Geotechnics, v. 31, n. 5, p. 411–425, 2004.

CHEN, W.-F. **Plasticity in reinforced concrete.** Fort Lauderdale: J. Ross Publishing, 2007.

CHEN, W.-F.; HAN, D. J. **Plasticity for structural engineers.** New York: Springer-Verlag, 1988.

CHUANG, Y.-C.; CHEN, C.-T.; HWANG, C. **A real-coded genetic algorithm with a direction-based crossover operator.** Information Sciences, v. 305, n. 1, p. 320–348, 2015.

CIVIDINI, A.; JURINA, L.; GIODA, G. **Some aspects of ‘characterization’ problems in geomechanics.** International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, v. 18, n. 6, p. 487–503, 1981.

CLAUSEN, J.; DAMKILDE, L.; ANDERSEN, L. **Efficient return algorithms for associated plasticity with multiple yield planes.** International Journal for Numerical Methods in Engineering, v. 66, n. 6, p. 1036–1059, 2006.

DEB, K. **Optimization for engineering design: Algorithm and Examples.** 2. ed. New Delhi: PHI Learning Private Limited, 2012.

DEB, K.; AGRAWAL, R. B. **Simulated binary crossover for continuous search space.** Complex Systems, v. 9, n. 2, p. 115–148, 1995.

DEEP, K.; THAKUR, M. **A new crossover operator for real coded genetic algorithms.** Applied Mathematics and Computation, v. 188, n. 1, p. 895–911, 2007.

DUTKO, M.; PERIĆ, D.; OWEN, D. R. J. **Universal anisotropic yield criterion based on superquadric functional representation: Part 1. Algorithmic issues and accuracy analysis.** Computer Methods in Applied Mechanics and Engineering, v. 109, n. 1–2, p. 73–93, 1993.

EBOLI, C. R.; VAZ, L. E.; VARGAS JÚNIOR, E. do A. **Considerações sobre algoritmos de atualizações de tensões em análise elastoplásticas associada e não associada.** In: Congreso Argentino de Mecánica Computacional, 1996, Tucumán. Mecánica Computacional Vol. XVII. Tucumán: MECON, 1996, p. 291-300.

ELSAYED, S. M.; SARKER, R. A.; ESSAM, D. L. **A comparative study of different variants of genetic algorithms for constrained optimization.** In: Asia-Pacific Conference on Simulated Evolution and Learning, 2010, Kanpur. Lecture Notes in Computer Science, Vol. 6457. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, p. 177-186.

FINNO, R. J.; CALVELLO, M. **Supported excavations: Observational method and inverse modeling.** Journal of Geotechnical and Geoenvironmental Engineering, v. 131, n. 7, p. 826–836, 2005.

FOSTER, C. D.; REGUEIRO, R. A.; FOSSUM, A. F.; BORJA, R. I. **Implicit numerical integration of a three-invariant, isotropic/kinematic hardening cap plasticity model for geomaterials.** *Computer Methods in Applied Mechanics and Engineering*, v. 194, n. 50–52, p. 5109–5138, 2005.

GENS, A.; LEDESMA, A.; ALONSO, E. E. **Estimation of parameters in geotechnical backanalysis - II.** Application to a tunnel excavation problem. *Computers and Geotechnics*, v. 18, n. 1, p. 29–46, 1996.

GIODA, G.; MAIER, G. **Direct search solution of an inverse problem in elastoplasticity:** Identification of cohesion, friction angle and in situ stress by pressure tunnel tests. *International Journal for Numerical Methods in Engineering*, v. 15, n. 12, p. 1823–1848, 1980.

GODIO, M.; STEFANOU, I.; SAB, K.; SULEM, J. **Multisurface plasticity for Cosserat materials:** Plate element implementation and validation. *International Journal for Numerical Methods in Engineering*, v. 108, n. 5, p. 456–484, 2016.

GOLDBERG, D. E. **Real-coded genetic algorithms, virtual alphabets, and blocking.** *Complex Systems*, v. 5, n. 2, p. 139–167, 1991.

HASHASH, Y. M. A.; LEVASSEUR, S.; OSOULI, A.; FINNO, R.; MALECOT, Y. **Comparison of two inverse analysis techniques for learning deep excavation response.** *Computers and Geotechnics*, v. 37, n. 3, p. 323–333, 2010.

HOFSTETTER, G.; SIMO, J. C.; TAYLOR, R. L. **A modified cap model:** Closest point solution algorithms. *Computers & Structures*, v. 46, n. 2, p. 203–214, 1993.

HUANG, J.; GRIFFITHS, D. V. **Observations on return mapping algorithms for piecewise linear yield criteria.** *International Journal of Geomechanics*, v. 8, n. 4, p. 253–265, 2008.

HUANG, J.; GRIFFITHS, D. V. **Return mapping algorithms and stress predictors for failure analysis in geomechanics.** *Journal of Engineering Mechanics*, v. 135, n. 4, p. 276–284, 2009.

JAKOBSEN, K. P.; LADE, P. V. **Implementation algorithm for a single hardening constitutive model for frictional materials.** *International Journal for Numerical and Analytical Methods in Geomechanics*, v. 26, n. 7, p. 661–681, 2002.

JIA, Y.; CHI, S. **Back-analysis of soil parameters of the Malutang II concrete face rockfill dam using parallel mutation particle swarm optimization.** *Computers and Geotechnics*, v. 65, p. 87–96, 2015.

JIANG, Q.; SUN, Y.; YI, B.; LI, T.; XIONG, F. **Inverse analysis for geomaterial parameter identification using Pareto multiobjective optimization.** *International Journal for Numerical and Analytical Methods in Geomechanics*, v. 42, n. 14, p. 1698–1718, 2018.

JIN, Y.-F.; YIN, Z.-Y.; SHEN, S.-L.; HICHER, P.-Y. **Selection of sand models and identification of parameters using an enhanced genetic algorithm.** International Journal for Numerical and Analytical Methods in Geomechanics, v. 40, n. 8, p. 1219–1240, 2016.

JIN, Y.-F.; YIN, Z.-Y.; SHEN, S.-L.; ZHANG, D.-M. **A new hybrid real-coded genetic algorithm and its application to parameters identification of soils.** Inverse Problems in Science and Engineering, v. 25, n. 9, p. 1343–1366, 2017.

JU, J.-W. W.; SIMO, J. C.; PISTER, K. S.; TAYLOR, R. L. **A parameter estimation algorithm and extensive numerical simulations for the cap model.** 2016. 48f. Relatório Técnico – Department of Civil Engineering, University of California, Berkeley, 1985.

KAELO, P.; ALI, M. M. **Integrated crossover rules in real coded genetic algorithms.** European Journal of Operational Research, v. 176, n. 1, p. 60–76, 2007.

KARAOULANIS, F. E. **Implicit numerical integration of nonsmooth multisurface yield criteria in the principal stress space.** Archives of Computational Methods in Engineering, v. 20, n. 3, p. 263–308, 2013.

KENNEDY, J.; EBERHART, R. **Particle swarm optimization.** In: International Conference on Neural Networks, 1995, Perth. Proceedings of ICNN'95 - International Conference on Neural Networks. IEEE, 1995, p. 1942–1948

KNABE, T.; DATCHEVA, M.; LAHMER, T.; COTECCHIA, F.; SCHANZ, T. **Identification of constitutive parameters of soil using an optimization strategy and statistical analysis.** Computers and Geotechnics, v. 49, p. 143–157, 2013.

KOITER, W. T. **Stress-strain relations, uniqueness and variational theorems for elastic-plastic materials with a singular yield surface.** Quarterly of Applied Mathematics, v. 11, n. 3, p. 350–354, 1953.

KOJIĆ, M.; BATHE, K.-J. **Inelastic analysis of solids and structures.** Heidelberg: Springer-Verlag, 2005.

KRABBENHOFT, K.; LYAMIN, A. V.; SLOAN, S. W.; WRIGGERS, P. **An interior-point algorithm for elastoplasticity.** International Journal for Numerical Methods in Engineering, v. 69, n. 3, p. 592–626, 2007.

LE, T. M.; FATAHI, B.; KHABBAZ, H. **Numerical optimisation to obtain elastic viscoplastic model parameters for soft clay.** International Journal of Plasticity, v. 65, p. 1–21, 2015.

LEDESMA, A.; GENS, A.; ALONSO, E. E. **Estimation of parameters in geotechnical backanalysis - I. Maximum likelihood approach.** Computers and Geotechnics, v. 18, n. 1, p. 1–27, 1996.

LESTER, B. T.; SCHERZINGER, W. M. **Trust-region based return mapping algorithm for implicit integration of elastic-plastic constitutive models.** International Journal for Numerical Methods in Engineering, v. 112, n. 3, p. 257–282, 2017.

LEVASSEUR, S.; MALÉCOT, Y.; BOULON, M.; FLAVIGNY, E. **Soil parameter identification using a genetic algorithm.** International Journal for Numerical and Analytical Methods in Geomechanics, v. 32, n. 2, p. 189–213, 2008.

LI, T.; CROUCH, R. **A C_2 plasticity model for structural concrete.** Computers & Structures, v. 88, n. 23–24, p. 1322–1332, 2010.

LIMA, M. J.; LEMOS, M.; PEREIRA, F.; COUTO, R.; ROEHL, D. **Functional requirements for developing ERAS - A portal to support collaborative geomechanical simulations.** In: BreSci - XII Brazilian e-Science Workshop - XXXVIII Congresso da Sociedade Brasileira de Computação, 2018, Natal. Proceedings of the BreSci - XII Brazilian e-Science Workshop. Natal, 2018.

LOURENÇO, P. B.; BORST, R. de; ROTS, J. G. **A plane stress softening plasticity model for orthotropic materials.** International Journal for Numerical Methods in Engineering, v. 40, n. 21, p. 4033–4057, 1997.

LOURENÇO, P. B.; ROTS, J. G. **Multisurface interface model for analysis of masonry structures.** Journal of Engineering Mechanics, v. 123, n. 7, p. 660–668, 1997.

MACKENZIE-HELNWEIN, P.; EBERHARDSTEINER, J.; MANG, H. A. **A multi-surface plasticity model for clear wood and its application to the finite element analysis of structural details.** Computational Mechanics, v. 31, n. 1–2, p. 204–218, 2003.

MATLAB R2018b. Natick, Massachusetts: MathWorks, 2018.

MELO, L. T. B. de. **Utilização de um modelo elasto-plástico para a análise de deformações em solos.** 1995. 98f. Tese (Doutorado) – Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1995.

MENDES, C. A. T. **GeMA, um novo framework para a prototipação, desenvolvimento e integração de simulações multifísicas e multiescalas em grupos multidisciplinares.** 2016. 168f. Tese (Doutorado) – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2016.

MICHALEWICZ, Z. **Genetic algorithms + data structures = evolution Programs.** 3. ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996.

MICHALEWICZ, Z.; SCHOENAUER, M. **Evolutionary algorithms for constrained parameter optimization problems.** Evolutionary Computation, v. 4, n. 1, p. 1–32, 1996.

MILLER, B. L.; GOLDBERG, D. E. **Genetic algorithms, tournament selection, and the effects of noise.** *Complex Systems*, v. 9, n. 3, p. 193–212, 1995.

MIRANDA, T.; DIAS, D.; ECLAIRCY-CAUDRON, S.; GOMES CORREIA, A.; COSTA, L. **Back analysis of geomechanical parameters by optimisation of a 3D model of an underground structure.** *Tunnelling and Underground Space Technology*, v. 26, n. 6, p. 659–673, 2011.

MOKHADE, A. S.; KAKDE, O. G. **Overview of selection schemes in real-coded genetic algorithms and their applications.** *Journal of Industrial and Intelligent Information*, v. 2, n. 1, p. 71–77, 2014.

MOREIRA, N.; MIRANDA, T.; PINHEIRO, M.; FERNANDES, P.; DIAS, D.; COSTA, L.; SENA-CRUZ, J. **Back analysis of geomechanical parameters in underground works using an Evolution Strategy algorithm.** *Tunnelling and Underground Space Technology*, v. 33, p. 143–158, 2013.

NOCEDAL, J.; WRIGHT, S. J. **Numerical optimization.** New York: Springer New York, 2006.

ORTIZ, M.; POPOV, E. P. **Accuracy and stability of integration algorithms for elastoplastic constitutive relations.** *International Journal for Numerical Methods in Engineering*, v. 21, n. 9, p. 1561–1576, 1985.

ORTIZ, M.; SIMO, J. C. **An analysis of a new class of integration algorithms for elastoplastic constitutive relations.** *International Journal for Numerical Methods in Engineering*, v. 23, n. 3, p. 353–366, 1986.

PAL, S.; WATHUGALA, G. W.; KUNDU, S. **Calibration of a constitutive model using genetic algorithms.** *Computers and Geotechnics*, v. 19, n. 4, p. 325–348, 1996.

PANKAJ; BIĆANIĆ, N. **Detection of multiple active yield conditions for Mohr-Coulomb elasto-plasticity.** *Computers & Structures*, v. 62, n. 1, p. 51–61, 1997.

PAPON, A.; RIOU, Y.; DANO, C.; HICHER, P.-Y. **Single-and multi-objective genetic algorithm optimization for identifying soil parameters.** *International Journal for Numerical and Analytical Methods in Geomechanics*, v. 36, n. 5, p. 597–618, 2012.

PÉREZ-FOGUET, A.; ARMERO, F. **On the formulation of closest-point projection algorithms in elastoplasticity—part II: Globally convergent schemes.** *International Journal for Numerical Methods in Engineering*, v. 53, n. 2, p. 331–374, 2002.

PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. **Numerical recipes.** 3. ed. Cambridge: Cambridge University Press, 2007.

PYTHON 3.6.7. Delaware: Python Software Foundation, 2018.

RAO, S. S. **Engineering optimization**. 4. ed. Hoboken: John Wiley & Sons, Inc., 2009.

RECHEA, C.; LEVASSEUR, S.; FINNO, R. **Inverse analysis techniques for parameter identification in simulation of excavation support systems**. Computers and Geotechnics, v. 35, n. 3, p. 331–345, 2008.

ROKONUZZAMAN, M.; SAKAI, T. **Calibration of the parameters for a hardening-softening constitutive model using genetic algorithms**. Computers and Geotechnics, v. 37, n. 4, p. 573–579, 2010.

SAKSALA, T. **Geometric return algorithm for non-associated plasticity with multiple yield planes extended to linear softening/hardening models**. Rakenteiden Mekaniikka (Journal of Structural Mechanics), v. 42, n. 2, p. 83–98, 2009.

SALAMI, M. R. **Constitutive modelling of concrete and rocks under multiaxial compressive loadings**. 1986. 406f. Tese (Doutorado) - University of Arizona, Tucson, 1986.

SAMARAJIVA, P.; MACARI, E. J.; WATHUGALA, W. **Genetic algorithms for the calibration of constitutive models for soils**. International Journal of Geomechanics, v. 5, n. 3, p. 206, 2005.

SANDLER, I. S.; RUBIN, D. **An algorithm and a modular subroutine for the CAP model**. International Journal for Numerical and Analytical Methods in Geomechanics, v. 3, n. 2, p. 173–186, 1979.

SAWYERR, B. A.; ADEWUMI, A. O.; ALI, M. M. **Real-coded genetic algorithm with uniform random local search**. Applied Mathematics and Computation, v. 228, p. 589–597, 2014.

SAWYERR, B. A.; ADEWUMI, A. O.; ALI, M. M. **Benchmarking RCGAu on the noiseless BBOB testbed**. The Scientific World Journal, v. 2015, 2015.

SAWYERR, B. A.; ALI, M. M.; ADEWUMI, A. O. **A comparative study of some real-coded genetic algorithms for unconstrained global optimization**. Optimization Methods and Software, v. 26, n. 6, p. 945–970, 2011.

SCALET, G.; AURICCHIO, F. **Computational methods for elastoplasticity: An overview of conventional and less-conventional approaches**. Archives of Computational Methods in Engineering, v. 25, n. 3, p. 545–589, 2018.

SCHERZINGER, W. M. **A return mapping algorithm for isotropic and anisotropic plasticity models using a line search method**. Computer Methods in Applied Mechanics and Engineering, v. 317, p. 526–553, 2017.

SCHWER, L. E.; MURRAY, Y. D. **A three-invariant smooth cap model with mixed hardening**. International Journal for Numerical and Analytical Methods in Geomechanics, v. 18, n. 10, p. 657–688, 1994.

SEIFERT, T.; SCHMIDT, I. **Line-search methods in general return mapping algorithms with application to porous plasticity.** International Journal for Numerical Methods in Engineering, v. 73, n. 10, p. 1468–1495, 2008.

SIMO, J. C.; HUGHES, T. J. R. **Computational inelasticity.** New York: Springer-Verlag, 1998.

SIMO, J. C.; JU, J.; PISTER, K. S.; TAYLOR, R. L. **Assessment of cap model: Consistent return algorithms and rate-dependent extension.** Journal of Engineering Mechanics, v. 114, n. 2, p. 191–218, 1988.

SIMO, J. C.; JU, J. W. **Strain- and stress-based continuum damage models—I. Formulation.** International Journal of Solids and Structures, v. 23, n. 7, p. 821–840, 1987a.

SIMO, J. C.; JU, J. W. **Strain- and stress-based continuum damage models—II. Computational aspects.** International Journal of Solids and Structures, v. 23, n. 7, p. 841–869, 1987b.

SIMO, J. C.; KENNEDY, J. G.; GOVINDJEE, S. **Non-smooth multisurface plasticity and viscoplasticity.** Loading/unloading conditions and numerical algorithms. International Journal for Numerical Methods in Engineering, v. 26, n. 10, p. 2161–2185, 1988.

SIMO, J. C.; TAYLOR, R. L. **Consistent tangent operators for rate-independent elastoplasticity.** Computer Methods in Applied Mechanics and Engineering, v. 48, n. 1, p. 101–118, 1985.

SIMO, J. C.; TAYLOR, R. L. **A return mapping algorithm for plane stress elastoplasticity.** International Journal for Numerical Methods in Engineering, v. 22, n. 3, p. 649–670, 1986.

SLOAN, S. W. **Substepping schemes for the numerical integration of elastoplastic stress-strain relations.** International Journal for Numerical Methods in Engineering, v. 24, n. 5, p. 893–911, 1987.

SONG, Z.-P.; JIANG, A.-N.; JIANG, Z.-B. **Back analysis of geomechanical parameters using hybrid algorithm based on difference evolution and extreme learning machine.** Mathematical Problems in Engineering, v. 2015, p. 1–11, 2015.

SOUZA NETO, E. A. de; PERIĆ, D.; OWEN, D. R. J. **Computational methods for plasticity.** Chichester: John Wiley & Sons, Ltd, 2008.

SWOBODA, G.; ICHIKAWA, Y.; DONG, Q.; ZAKI, M. **Back analysis of large geotechnical models.** International Journal for Numerical and Analytical Methods in Geomechanics, v. 23, n. 13, p. 1455–1472, 1999.

TANG, Y.-G.; KUNG, G. T.-C. **Application of nonlinear optimization technique to back analyses of deep excavation.** Computers and Geotechnics, v. 36, n. 1–2, p. 276–290, 2009.

TIKHONOV, A. N.; ARSENIN, V. Y. **Solutions of Ill-Posed Problems**. Washington: V. H. Winston & Sons, 1977.

VAIRAKTARIS, E. **Inverse problems in geomechanics: Review and examples**. European Journal of Environmental and Civil Engineering, v. 14, n. 8–9, p. 1155–1166, 2010.

VARDAKOS, S.; GUTIERREZ, M.; XIA, C. **Parameter identification in numerical modeling of tunneling using the Differential Evolution Genetic Algorithm (DEGA)**. Tunnelling and Underground Space Technology, v. 28, n. 1, p. 109–123, 2012.

YAZDI, J. S.; KALANTARY, F.; YAZDI, H. S. **Calibration of soil model parameters using particle swarm optimization**. International Journal of Geomechanics, v. 12, n. 3, p. 229–238, 2012.

YE, L.; JIN, Y.-F.; SHEN, S.-L.; SUN, P.-P.; ZHOU, C. **An efficient parameter identification procedure for soft sensitive clays**. Journal of Zhejiang University Science A, v. 17, n. 1, p. 76–88, 2016.

YIN, Z.-Y.; JIN, Y.-F.; SHEN, S.-L.; HUANG, H.-W. **An efficient optimization method for identifying parameters of soft structured clay by an enhanced genetic algorithm and elastic–viscoplastic model**. Acta Geotechnica, v. 12, n. 4, p. 849–867, 2016.

YIN, Z.-Y.; JIN, Y.-F.; SHEN, J. S.; HICHER, P.-Y. **Optimization techniques for identifying soil parameters in geotechnical engineering: Comparative study and enhancement**. International Journal for Numerical and Analytical Methods in Geomechanics, n. November 2016, p. 70–94, 2017.

ZENTAR, R.; HICHER, P. Y.; MOULIN, G. **Identification of soil parameters by inverse analysis**. Computers and Geotechnics, v. 28, n. 2, p. 129–144, 2001.

ZHANG, Y.; GALLIPOLI, D.; AUGARDE, C. E. **Simulation-based calibration of geotechnical parameters using parallel hybrid moving boundary particle swarm optimization**. Computers and Geotechnics, v. 36, n. 4, p. 604–615, 2009.

ZHANG, Y.; GALLIPOLI, D.; AUGARDE, C. E. **Parameter identification for elasto-plastic modelling of unsaturated soils from pressuremeter tests by parallel modified particle swarm optimization**. Computers and Geotechnics, v. 48, p. 293–303, 2013.

ZHAO, B. D.; ZHANG, L. L.; JENG, D. S.; WANG, J. H.; CHEN, J. J. **Inverse analysis of deep excavation using differential evolution algorithm**. International Journal for Numerical and Analytical Methods in Geomechanics, v. 39, n. 2, p. 115–134, 2015.

ZHAO, H.-B; YIN, S. **Geomechanical parameters identification by particle swarm optimization and support vector machine**. Applied Mathematical Modelling, v. 33, n. 10, p. 3997–4012, 2009.

Apêndice A

Resultados do Estudo Paramétrico do Algoritmo Genético

Os gráficos a seguir se referem ao estudo paramétrico desenvolvido para os problemas de dimensão variável, indicados na Tabela 5.2. Esses resultados foram importantes para entender o impacto dos parâmetros dos operadores na solução final. Considerou-se as dimensões: 5, 10, 20, 30 e 40.

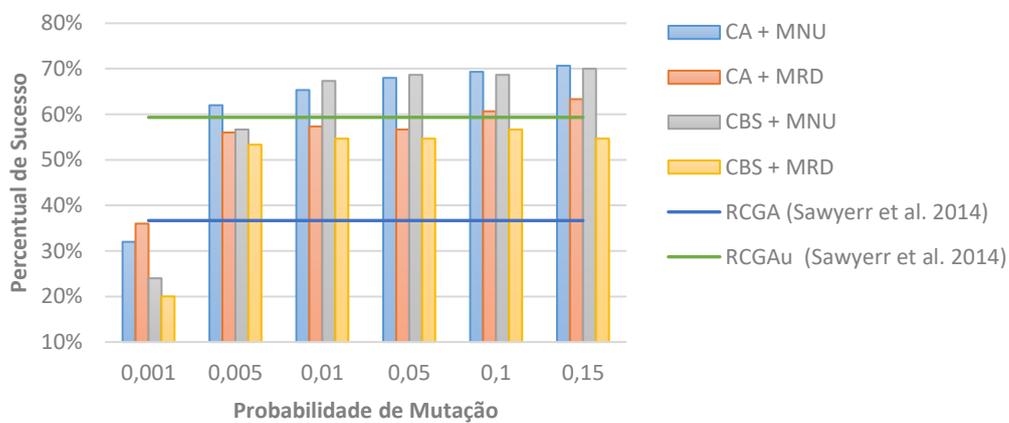


Gráfico A.1 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 5)

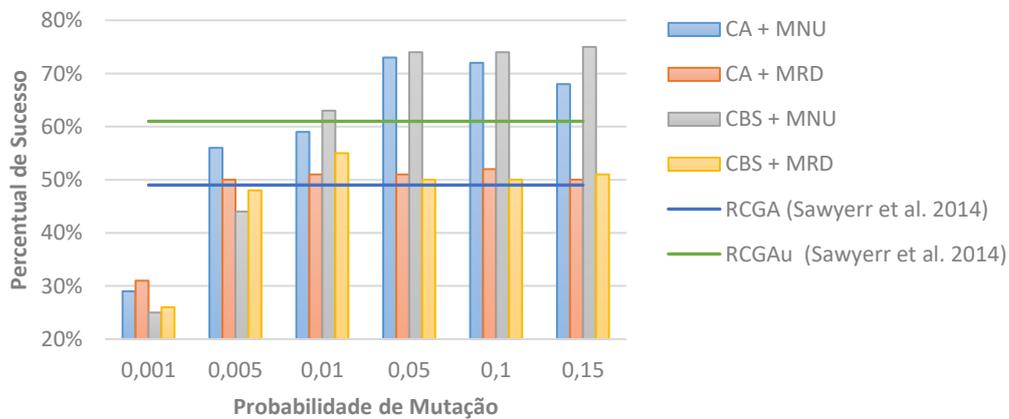


Gráfico A.2 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 10)

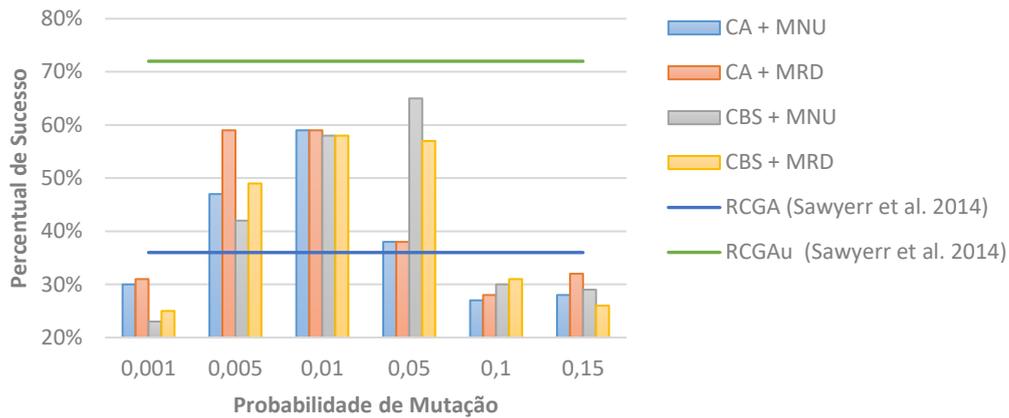


Gráfico A.3 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 20)

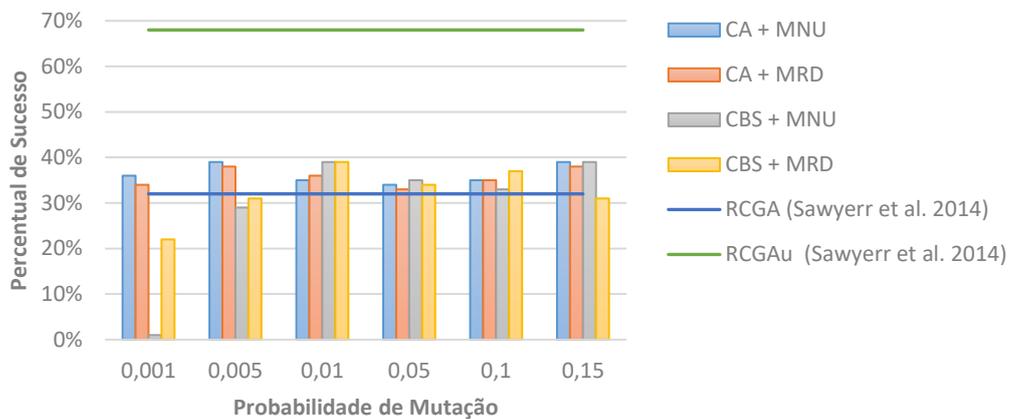


Gráfico A.4 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 30)

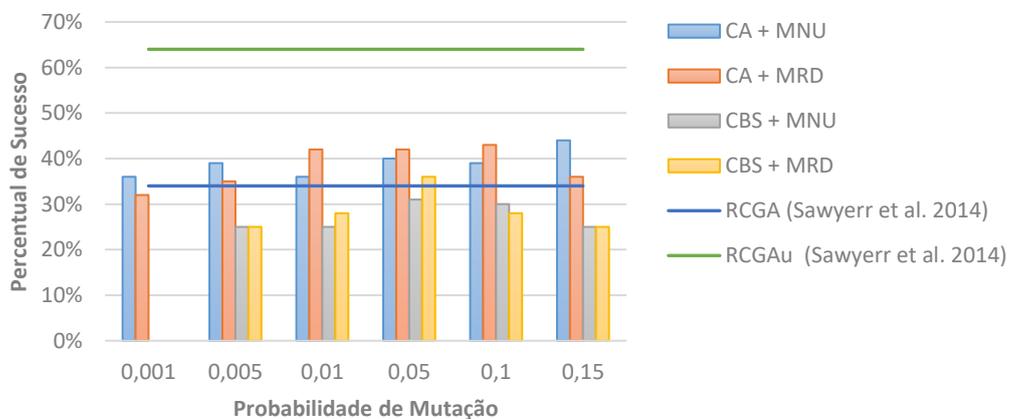


Gráfico A.5 – Resultados do estudo paramétrico da probabilidade de mutação para as funções de dimensão variável (dimensão 40)

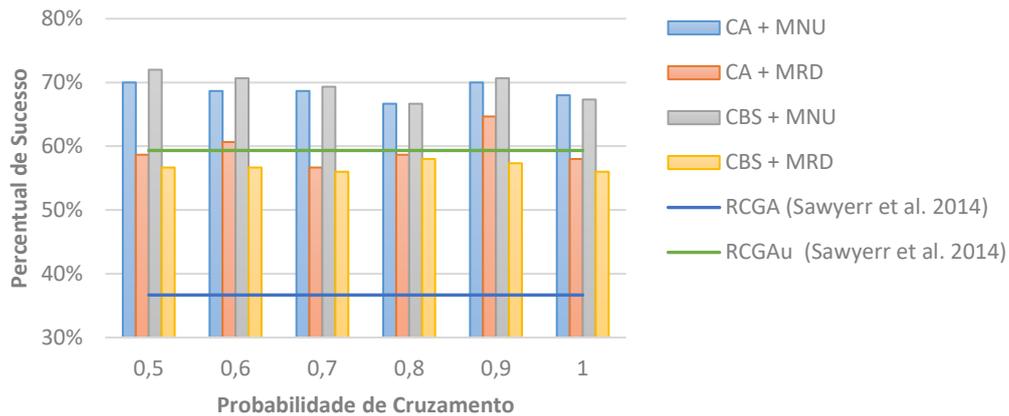


Gráfico A.6 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 5)

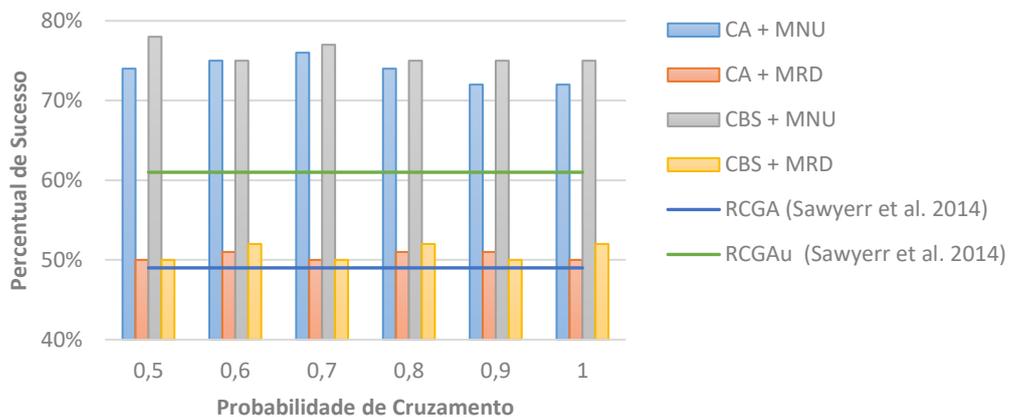


Gráfico A.7 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 10)

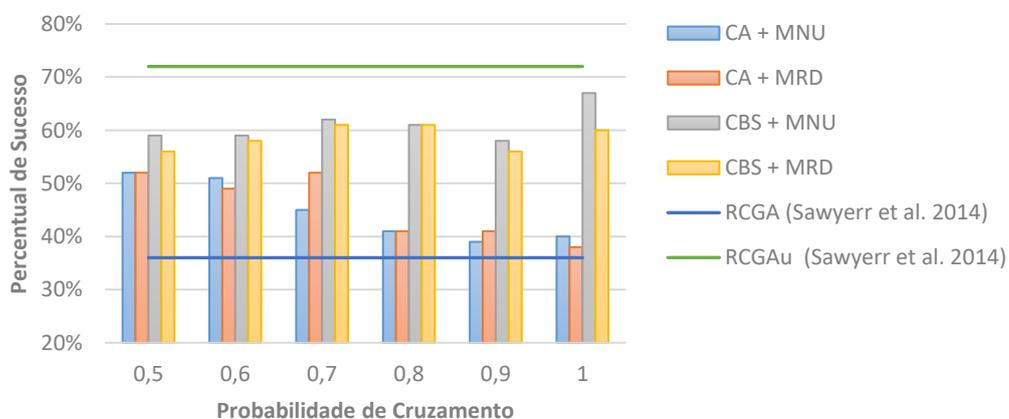


Gráfico A.8 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 20)

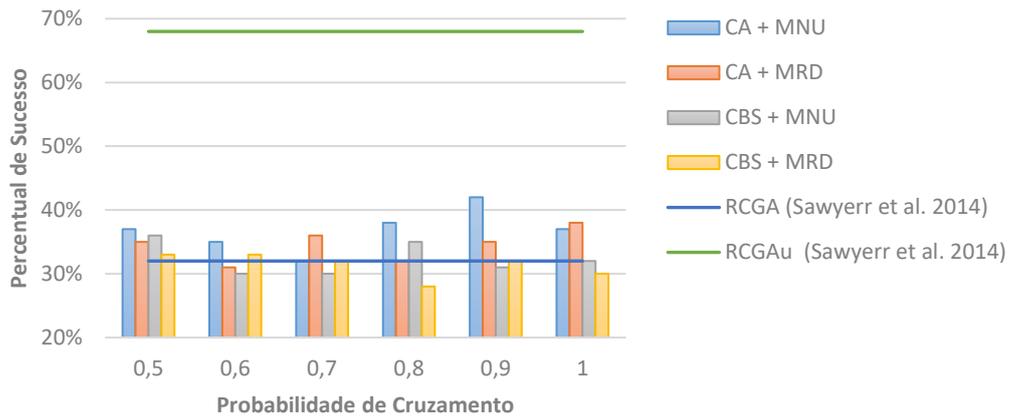


Gráfico A.9 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 30)

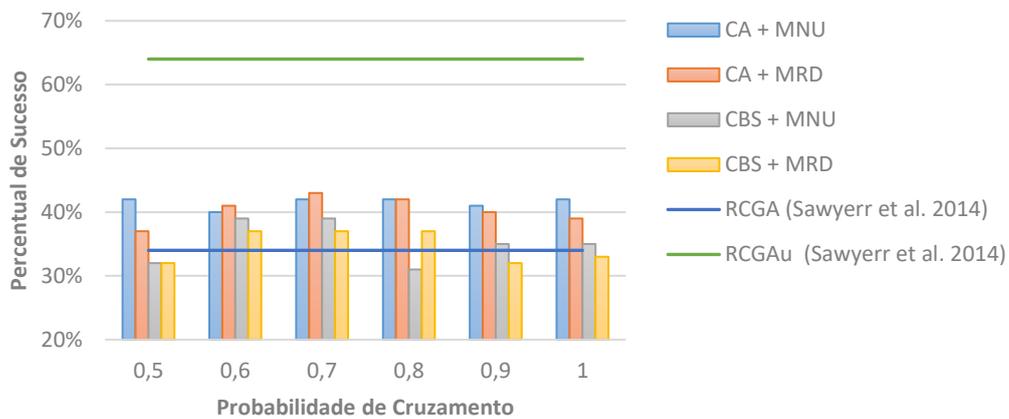


Gráfico A.10 – Resultados do estudo paramétrico da probabilidade de cruzamento para as funções de dimensão variável (dimensão 40)

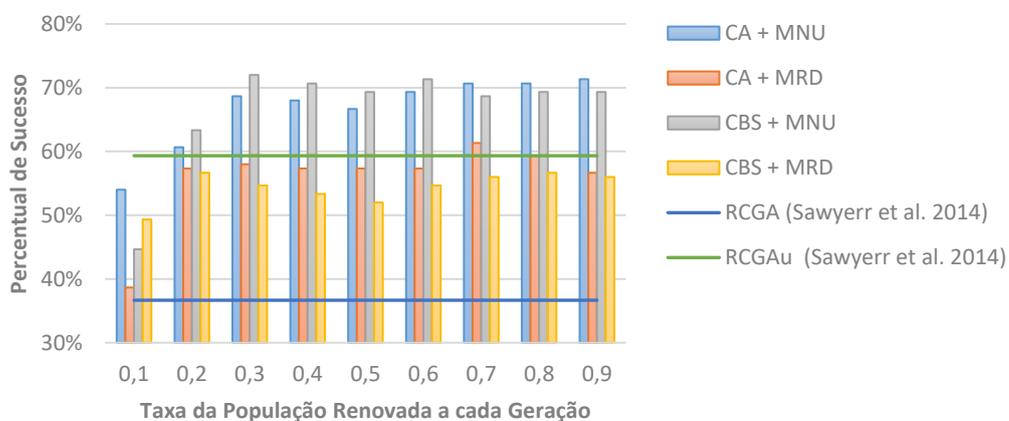


Gráfico A.11 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 5)

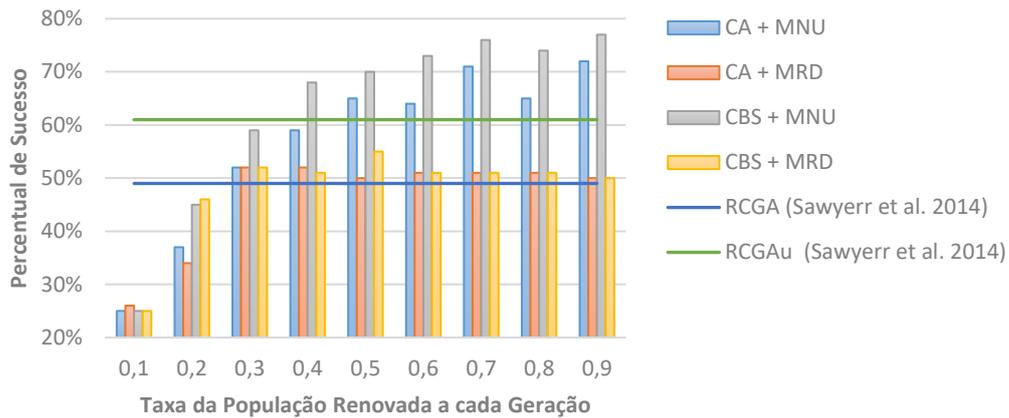


Gráfico A.12 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 10)

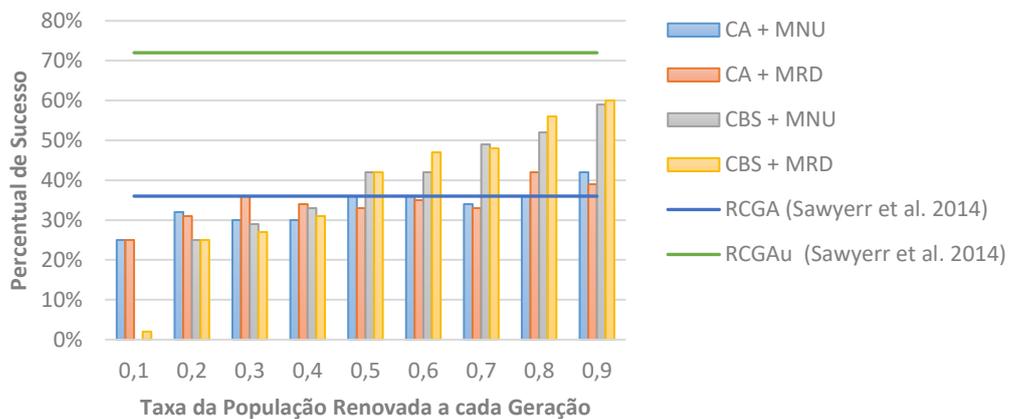


Gráfico A.13 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 20)

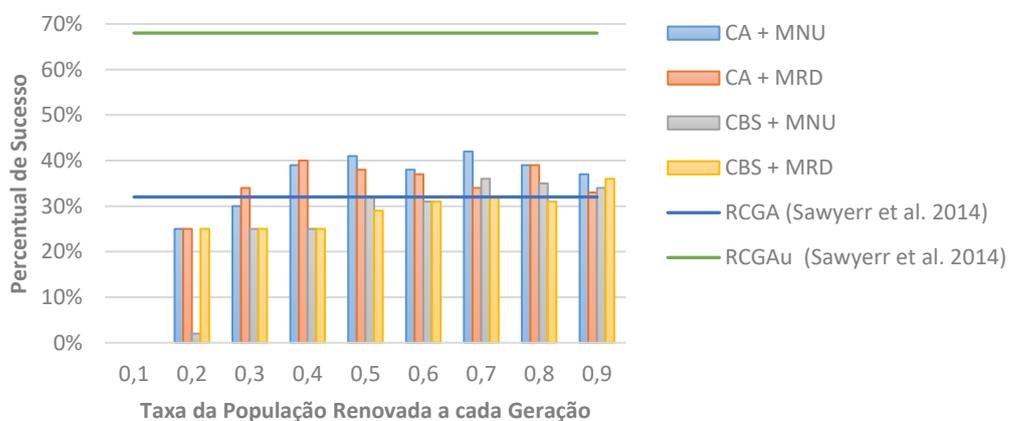


Gráfico A.14 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 30)

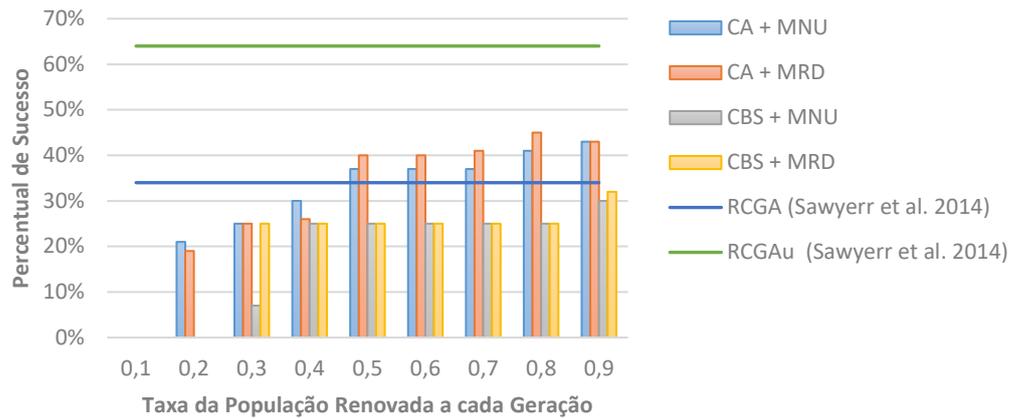


Gráfico A.15 – Resultados do estudo paramétrico da taxa de população renovada a cada geração para as funções de dimensão variável (dimensão 40)

Ao fim deste estudo, definiu-se os parâmetros apresentados na Tabela 5.4. Para analisar a qualidade das soluções obtidas pelo algoritmo para tais parâmetros, avaliou-se a média e desvio padrão do erro definido por (5.12), considerando 25 inicializações do algoritmo por problema de otimização. Assim, os resultados apresentados nas tabelas a seguir sintetizam os resultados obtidos.

Nas tabelas são apresentados resultados para todos os problemas de otimização analisados neste trabalho. Apresenta-se os resultados para três probabilidades de mutação: 0,01; 0,05 e 0,15. Compara-se os resultados com os apresentados por Sawyer, Adewumi e Ali (2014). Apesar de diferenças entre os parâmetros empregados pelos autores e os adotados por este trabalho, considera-se um bom estudo comparativo

Tabela A.1 – Quantidade e taxas de sucessos para os problemas de otimização com dimensão fixada

Nº	D	Quantidade de Sucessos					Taxas de Sucessos				
		Probabilidades de Mutação			Sawyer et al. (2014)		Probabilidades de Mutação			Sawyer et al. (2014)	
		0,01	0,05	0,15	RCGA	RCGAu	0,01	0,05	0,15	RCGA	RCGAu
1	2	25	25	25	25	25	100%	100%	100%	100%	100%
2	2	25	25	25	25	25	100%	100%	100%	100%	100%
3	2	25	25	25	25	25	100%	100%	100%	100%	100%
4	2	25	25	25	24	25	100%	100%	100%	96%	100%
5	2	25	25	25	25	25	100%	100%	100%	100%	100%
6	4	25	25	25	25	25	100%	100%	100%	100%	100%
7	2	25	25	25	25	25	100%	100%	100%	100%	100%
8	2	25	25	25	25	25	100%	100%	100%	100%	100%
9	3	25	25	25	24	25	100%	100%	100%	96%	100%
10	3	25	25	25	25	25	100%	100%	100%	100%	100%
11	6	16	17	12	19	22	64%	68%	48%	76%	88%
12	3	25	25	25	24	25	100%	100%	100%	96%	100%
13	2	25	25	25	25	25	100%	100%	100%	100%	100%
14	4	25	25	25	25	25	100%	100%	100%	100%	100%
15	2	22	18	25	10	20	88%	72%	100%	40%	80%
16	2	14	15	20	19	25	56%	60%	80%	76%	100%
17	4	25	25	25	9	25	100%	100%	100%	36%	100%
18	2	25	25	25	22	16	100%	100%	100%	88%	64%
19	4	12	16	14	10	10	48%	64%	56%	40%	40%
20	4	17	15	17	15	23	68%	60%	68%	60%	92%
21	4	19	17	18	13	20	76%	68%	72%	52%	80%
22	20	15	21	23	6	25	60%	84%	92%	24%	100%

Tabela A.2 – Médias das funções erro para os problemas de otimização com dimensão fixada

N°	D	Médias das Funções Erro				
		Probabilidades de Mutação			Sawyers et al. (2014)	
		0,01	0,05	0,15	RCGA	RCGAu
1	2	8,61E-05	8,61E-05	8,61E-05	4,12E-05	5,18E-05
2	2	0,00E+00	0,00E+00	0,00E+00	5,12E-05	4,38E-05
3	2	0,00E+00	0,00E+00	0,00E+00	4,34E-05	5,04E-05
4	2	0,00E+00	0,00E+00	0,00E+00	3,54E-04	4,40E-05
5	2	2,22E-16	2,22E-16	2,22E-16	4,45E-04	9,21E-05
6	4	0,00E+00	0,00E+00	0,00E+00	1,80E-04	1,66E-04
7	2	4,23E-05	4,23E-05	4,23E-05	1,86E-09	1,50E-09
8	2	7,45E-14	7,61E-14	7,70E-14	1,68E-05	1,52E-05
9	3	3,19E-04	1,40E-04	4,99E-06	1,31E-03	1,05E-03
10	3	1,48E-07	1,48E-07	1,48E-07	1,66E-05	1,62E-05
11	6	1,14E-08	1,14E-08	1,14E-08	2,26E-05	2,11E-05
12	3	1,80E-310	2,71E-295	8,62E-93	6,40E-05	5,50E-05
13	2	1,16E-05	1,16E-05	1,16E-05	2,53E-05	1,69E-05
14	4	2,80E-07	1,90E-08	5,99E-09	2,35E-01	2,01E-01
15	2	4,05E-06	4,05E-06	4,05E-06	6,43E-04	3,33E-06
16	2	0,00E+00	0,00E+00	0,00E+00	9,51E-04	5,10E-05
17	4	4,62E-06	1,75E-06	2,91E-07	1,91E-03	6,57E-05
18	2	8,83E-06	8,83E-06	8,83E-06	3,57E-07	1,18E-07
19	4	3,21E-07	3,21E-07	3,21E-07	7,16E-06	6,88E-06
20	4	4,06E-05	4,06E-05	4,06E-05	7,29E-06	5,02E-06
21	4	9,82E-06	9,82E-06	9,82E-06	3,32E-05	5,30E-06
22	20	0,00E+00	3,68E-12	7,91E-11	6,78E-04	2,57E-05

Tabela A.3 – Desvios padrões das funções erro para os problemas de otimização com dimensão fixada

N°	D	Desvios Padrões das Funções Erro				
		Probabilidades de Mutação			Sawyer et al. (2014)	
		0,01	0,05	0,01	0,05	0,01
1	2	0,00E+00	0,00E+00	0,00E+00	5,47E-05	4,50E-05
2	2	0,00E+00	0,00E+00	0,00E+00	3,42E-05	2,97E-05
3	2	0,00E+00	0,00E+00	0,00E+00	2,65E-05	3,17E-05
4	2	0,00E+00	0,00E+00	0,00E+00	4,23E-02	2,97E-05
5	2	0,00E+00	0,00E+00	0,00E+00	5,58E-04	2,66E-05
6	4	0,00E+00	0,00E+00	0,00E+00	2,37E-05	2,26E-05
7	2	7,28E-13	7,28E-13	0,00E+00	3,99E-04	4,88E-04
8	2	3,04E-15	1,81E-15	1,85E-15	3,06E-05	3,30E-05
9	3	5,45E-04	1,92E-04	7,38E-06	3,94E-02	9,56E-04
10	3	8,88E-17	8,88E-17	8,88E-17	2,14E-05	2,37E-05
11	6	1,79E-16	1,75E-16	2,29E-16	5,48E-02	3,95E-02
12	3	0,00E+00	0,00E+00	4,31E-92	1,97E-02	2,32E-05
13	2	3,10E-15	0,00E+00	0,00E+00	3,02E-05	3,36E-05
14	4	5,05E-07	1,22E-08	5,78E-12	2,02E-05	2,36E-05
15	2	8,77E-17	5,23E-17	9,06E-17	3,89E-02	2,66E-02
16	2	0,00E+00	0,00E+00	0,00E+00	3,85E-02	2,57E-05
17	4	3,23E-06	1,60E-06	2,82E-07	4,03E+00	2,81E-05
18	2	1,67E-14	2,21E-14	2,37E-14	4,91E-01	1,99E-01
19	4	0,00E+00	6,07E-16	6,45E-16	3,64E+00	3,12E+00
20	4	8,34E-16	7,35E-16	6,28E-16	3,64E+00	1,52E+00
21	4	9,01E-16	1,07E-15	1,10E-15	3,85E+00	2,72E+00
22	20	0,00E+00	2,82E-12	3,47E-11	1,88E-01	1,16E-05

Tabela A.4 – Quantidade e taxas de sucessos para os problemas de otimização com dimensão variável

Nº	D	Quantidade de Sucessos					Taxas de Sucessos				
		Probabilidades de Mutação			Sawyer et al. (2014)		Probabilidades de Mutação			Sawyer et al. (2014)	
		0,01	0,05	0,15	RCGA	RCGAu	0,01	0,05	0,15	RCGA	RCGAu
23	5	25	25	25	25	25	100%	100%	100%	100%	100%
24	5	25	25	25	25	25	100%	100%	100%	100%	100%
25	5	4	2	4	1	0	16%	8%	16%	4%	0%
26	5	25	25	25	2	1	100%	100%	100%	8%	4%
27	5	1	0	0	1	16	4%	0%	0%	4%	64%
28	5	23	25	25	1	22	92%	100%	100%	4%	88%
23	10	25	25	25	22	25	100%	100%	100%	88%	100%
24	10	25	25	25	25	25	100%	100%	100%	100%	100%
25	10	0	0	2	2	1	0%	0%	8%	8%	4%
26	10	25	25	24	-	-	100%	100%	96%	-	-
27	10	1	0	1	-	-	4%	0%	4%	-	-
28	10	9	24	18	0	10	36%	96%	72%	0%	40%
23	20	25	6	0	9	25	100%	24%	0%	36%	100%
24	20	25	25	25	25	25	100%	100%	100%	100%	100%
25	20	8	8	3	2	15	32%	32%	12%	8%	60%
26	20	10	4	0	-	-	40%	16%	0%	-	-
27	20	0	0	0	-	-	0%	0%	0%	-	-
28	20	0	0	0	0	7	0%	0%	0%	0%	28%
23	30	0	0	0	1	24	0%	0%	0%	4%	96%
24	30	25	25	25	25	25	100%	100%	100%	100%	100%
25	30	12	13	15	6	15	48%	52%	60%	24%	60%
26	30	0	0	0	-	-	0%	0%	0%	-	-
27	30	0	0	0	-	-	0%	0%	0%	-	-
28	30	0	0	0	0	4	0%	0%	0%	0%	16%
23	40	0	0	0	0	18	0%	0%	0%	0%	72%
24	40	25	25	25	25	25	100%	100%	100%	100%	100%
25	40	21	15	13	9	16	84%	60%	52%	36%	64%
26	40	0	0	0	-	-	0%	0%	0%	-	-
27	40	0	0	0	-	-	0%	0%	0%	-	-
28	40	0	0	0	0	5	0%	0%	0%	0%	20%

Tabela A.5 – Médias das funções erro para os problemas de otimização com dimensão variável

N°	D	Médias das Funções Erro				
		Probabilidades de Mutação			Sawyer et al. (2014)	
		0,01	0,05	0,15	RCGA	RCGAu
23	5	1,33E-11	4,44E-16	5,72E-11	9,90E-05	8,33E-09
24	5	2,42E-136	5,98E-108	1,09E-20	7,00E-09	6,84E-09
25	5	3,70E-03	3,70E-03	3,70E-03	8,76E-09	-
26	5	0,00E+00	0,00E+00	0,00E+00	8,77E-09	6,52E-09
27	5	7,12E-03	-	-	1,26E-03	4,34E-03
28	5	1,36E-05	1,36E-05	1,36E-05	6,51E-09	6,51E-09
23	10	3,43E-15	2,37E-12	3,74E-08	4,25E-04	9,25E-09
24	10	3,48E-62	5,34E-23	2,48E-15	7,34E-09	8,06E-09
25	10	-	-	3,70E-03	3,70E-03	7,40E-03
26	10	0,00E+00	0,00E+00	6,38E-13	-	-
27	10	2,60E-03	-	7,52E-03	-	-
28	10	2,73E-05	2,73E-05	2,73E-05	1,00E+00	6,51E-09
23	20	5,09E-09	1,27E-06	-	5,33E-04	1,56E-08
24	20	1,10E-21	2,08E-12	6,21E-11	9,36E-08	9,03E-09
25	20	0,00E+00	1,85E-03	2,47E-03	5,14E-06	1,97E-03
26	20	1,78E-16	5,47E-10	-	-	-
27	20	-	-	-	-	-
28	20	-	-	-	1,00E+00	6,51E-09
23	30	-	-	-	8,59E-03	7,78E-05
24	30	4,83E-13	2,07E-09	2,02E-08	6,81E-07	1,54E-08
25	30	6,16E-04	5,71E-04	5,12E-04	2,53E-03	9,87E-04
26	30	-	-	-	-	-
27	30	-	-	-	-	-
28	30	-	-	-	1,00E+00	8,37E-09
23	40	-	-	-	0,00E+00	8,19E-04
24	40	1,16E-09	1,63E-07	8,73E-07	5,33E-06	1,99E-07
25	40	1,76E-03	1,08E-03	2,79E-03	3,18E-03	9,33E-04
26	40	-	-	-	-	-
27	40	-	-	-	-	-
28	40	-	-	-	1,00E+00	1,02E-07

Tabela A.6 – Desvios padrões das funções erro para os problemas de otimização com dimensão variável

N°	D	Desvios Padrões das Funções Erro				
		Probabilidades de Mutação			Sawyer et al. (2014)	
		0,01	0,05	0,01	0,05	0,01
23	5	6,66E-11	0,00E+00	6,92E-11	3,57E-04	1,49E-09
24	5	8,05E-136	2,48E-107	2,73E-20	2,73E-09	2,33E-09
25	5	4,27E-03	5,23E-03	4,27E-03	9,25E-02	6,39E-02
26	5	0,00E+00	0,00E+00	0,00E+00	1,48E+00	1,79E+00
27	5	-	-	-	6,97E+01	4,29E+00
28	5	3,07E-14	2,48E-14	3,08E-14	1,69E+02	9,16E+01
23	10	1,33E-15	5,49E-12	1,46E-08	1,20E+00	7,43E-10
24	10	1,18E-61	2,66E-22	1,64E-15	3,13E-09	1,47E-09
25	10	-	-	5,23E-03	2,11E-01	2,67E-01
26	10	0,00E+00	0,00E+00	4,87E-13	-	-
27	10	-	-	-	-	-
28	10	2,84E-14	4,58E-14	1,37E-12	5,22E+02	1,52E+02
23	20	6,23E-09	3,24E-07	-	1,76E+00	3,13E-08
24	20	4,70E-21	1,17E-12	4,58E-11	2,13E-07	1,22E-09
25	20	0,00E+00	3,42E-03	4,27E-03	1,29E-01	2,08E-01
26	20	5,62E-16	3,11E-10	-	-	-
27	20	-	-	-	-	-
28	20	-	-	-	1,15E+03	4,93E+02
23	30	-	-	-	1,77E+00	3,00E-01
24	30	4,43E-13	6,86E-10	1,14E-08	1,38E-06	3,01E-08
25	30	2,14E-03	2,05E-03	1,91E-03	3,72E-01	7,07E-03
26	30	-	-	-	-	-
27	30	-	-	-	-	-
28	30	-	-	-	1,90E+03	7,15E+02
23	40	-	-	-	2,46E+00	6,62E-01
24	40	9,96E-10	8,00E-08	4,56E-07	9,98E-06	6,54E-07
25	40	3,23E-03	2,62E-03	3,59E-03	6,57E-02	2,51E-01
26	40	-	-	-	-	-
27	40	-	-	-	-	-
28	40	-	-	-	2,54E+03	1,17E+03

Apêndice B

Equações Auxiliares para Implementação do *Cap Model*

Para aplicar o algoritmo descrito no Capítulo 3, é necessário calcular as derivadas referentes às funções de plastificação do modelo constitutivo à ser adotado. Neste trabalho, adotou-se o *Cap Model*. Dessa forma, a seguir são apresentadas as expressões necessárias para esse modelo.

As funções de plastificação são normalmente expressadas a partir dos invariantes do tensor de tensões. Como são necessários cálculos de derivadas em relação ao tensor de tensões, deve-se aplicar a regra da cadeia para obter tal derivada. Assim, define-se as derivadas das funções de plastificação em relação aos seus invariantes do *Cap Model*. Aqui serão apresentadas as expressões em notação matricial. Sendo assim, o tensor de tensões é definido da forma que se segue:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} \quad (\text{B.1})$$

B.1. Derivadas dos Invariantes do Tensor de Tensões

Calculando-se a primeira e segunda derivada de I_1 em relação ao tensor $\boldsymbol{\sigma}$, obtém-se:

$$\frac{\partial I_1}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{B.2})$$

$$\frac{\partial^2 I_1}{\partial \boldsymbol{\sigma}^2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{B.3})$$

Calculando-se a primeira e segunda derivada de J_2 em relação ao tensor $\boldsymbol{\sigma}$, obtém-se:

$$\frac{\partial J_2}{\partial \boldsymbol{\sigma}} = \begin{bmatrix} \frac{1}{3}(2\sigma_{xx} - \sigma_{yy} - \sigma_{zz}) \\ -\frac{1}{3}(\sigma_{xx} - 2\sigma_{yy} + \sigma_{zz}) \\ -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} - 2\sigma_{zz}) \\ 2\sigma_{xy} \\ 2\sigma_{xz} \\ 2\sigma_{yz} \end{bmatrix} \quad (\text{B.4})$$

$$\frac{\partial^2 J_2}{\partial \boldsymbol{\sigma}^2} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (\text{B.5})$$

B.2. Derivadas das Funções de Plastificação do *Cap Model*

Sendo as superfícies de plastificação do *Cap Model* definidas pelos invariantes I_1 e J_2 , aplicando a regra da cadeia obtém-se:

$$\frac{\partial f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial \boldsymbol{\sigma}} = \frac{\partial f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial I_1} \frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial J_2} \frac{\partial J_2}{\partial \boldsymbol{\sigma}} \quad (\text{B.6})$$

$$\begin{aligned} \frac{\partial^2 f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial \boldsymbol{\sigma}^2} &= \frac{\partial^2 f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial \boldsymbol{\sigma} \partial I_1} \frac{\partial I_1}{\partial \boldsymbol{\sigma}} + \frac{\partial f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial I_1} \frac{\partial^2 I_1}{\partial \boldsymbol{\sigma}^2} + \frac{\partial^2 f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial \boldsymbol{\sigma} \partial J_2} \frac{\partial J_2}{\partial \boldsymbol{\sigma}} \\ &+ \frac{\partial f_i(\boldsymbol{\sigma}, \boldsymbol{q})}{\partial J_2} \frac{\partial^2 J_2}{\partial \boldsymbol{\sigma}^2} \end{aligned} \quad (\text{B.7})$$

Nesse modelo, tem-se apenas uma variável interna que define o endurecimento, representado por $\mathbf{q} = [\kappa]$, que influencia apenas a superfície do *cap* elíptico. Dessa forma, são definidas tais derivadas a seguir para a superfície cônica do modelo:

$$\frac{\partial f_1(\boldsymbol{\sigma})}{\partial I_1} = \lambda \beta \exp(\beta I_1(\boldsymbol{\sigma})) + \theta \quad (\text{B.8})$$

$$\frac{\partial f_1(\boldsymbol{\sigma})}{\partial J_2} = \frac{1}{\sqrt{2J_2(\boldsymbol{\sigma})}} \quad (\text{B.9})$$

$$\frac{\partial^2 f_1(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma} \partial I_1} = \begin{bmatrix} \lambda \beta^2 \exp(\beta I_1(\boldsymbol{\sigma})) \\ \lambda \beta^2 \exp(\beta I_1(\boldsymbol{\sigma})) \\ \lambda \beta^2 \exp(\beta I_1(\boldsymbol{\sigma})) \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{B.10})$$

$$\frac{\partial^2 f_1(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma} \partial J_2} = -\frac{1}{\sqrt{2(J_2(\boldsymbol{\sigma}))^3}} \begin{bmatrix} \frac{1}{6}(3\sigma_{xx} - I_1(\boldsymbol{\sigma})) \\ \frac{1}{6}(3\sigma_{yy} - I_1(\boldsymbol{\sigma})) \\ \frac{1}{6}(3\sigma_{zz} - I_1(\boldsymbol{\sigma})) \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} \quad (\text{B.11})$$

Para o *cap* elíptico, define-se a seguinte expressão auxiliar:

$$\boldsymbol{\zeta}(\boldsymbol{\sigma}, \kappa) = \begin{bmatrix} (I_1(\boldsymbol{\sigma}) - L(\kappa))(\sigma_{xx} - p(\boldsymbol{\sigma})) - 2J_2(\boldsymbol{\sigma}) \\ (I_1(\boldsymbol{\sigma}) - L(\kappa))(\sigma_{yy} - p(\boldsymbol{\sigma})) - 2J_2(\boldsymbol{\sigma}) \\ (I_1(\boldsymbol{\sigma}) - L(\kappa))(\sigma_{zz} - p(\boldsymbol{\sigma})) - 2J_2(\boldsymbol{\sigma}) \\ 2(I_1(\boldsymbol{\sigma}) - L(\kappa))\sigma_{xy} \\ 2(I_1(\boldsymbol{\sigma}) - L(\kappa))\sigma_{xz} \\ 2(I_1(\boldsymbol{\sigma}) - L(\kappa))\sigma_{yz} \end{bmatrix} \quad (\text{B.12})$$

Dessa maneira, para o *cap* elíptico do modelo constitutivo, tem-se as seguintes derivadas:

$$\frac{\partial f_2(\boldsymbol{\sigma}, \kappa)}{\partial I_1} = \frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{\sqrt{2J_2(\boldsymbol{\sigma})R^4 + \left(R(I_1(\boldsymbol{\sigma}) - L(\kappa))\right)^2}} \quad (\text{B.13})$$

$$\frac{\partial f_2(\boldsymbol{\sigma}, \kappa)}{\partial J_2} = \frac{1}{\sqrt{2J_2(\boldsymbol{\sigma}) + \left(\frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R}\right)^2}} \quad (\text{B.14})$$

$$\frac{\partial^2 f_2(\boldsymbol{\sigma}, \kappa)}{\partial \boldsymbol{\sigma} \partial I_1} = -\frac{1}{R^2 \left(2J_2(\boldsymbol{\sigma}) + \left(\frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R}\right)^2\right)^{\frac{3}{2}}} \boldsymbol{\zeta}(\boldsymbol{\sigma}, \kappa) \quad (\text{B.15})$$

$$\frac{\partial^2 f_2(\boldsymbol{\sigma}, \kappa)}{\partial \boldsymbol{\sigma} \partial J_2} = -\frac{1}{\left(2J_2(\boldsymbol{\sigma}) + \left(\frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R}\right)^2\right)^{\frac{3}{2}}} \begin{bmatrix} \sigma_{xx} - p(\boldsymbol{\sigma}) + \frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R} \\ \sigma_{yy} - p(\boldsymbol{\sigma}) + \frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R} \\ \sigma_{zz} - p(\boldsymbol{\sigma}) + \frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R} \\ 2\sigma_{xy} \\ 2\sigma_{xz} \\ 2\sigma_{yz} \end{bmatrix} \quad (\text{B.16})$$

Por fim, para o *cut-off* de tração, tem-se:

$$\frac{\partial f_1(\boldsymbol{\sigma})}{\partial I_1} = 1 \quad (\text{B.17})$$

$$\frac{\partial f_1(\boldsymbol{\sigma})}{\partial J_2} = 0 \quad (\text{B.18})$$

$$\frac{\partial^2 f_1(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma} \partial I_1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{B.19})$$

$$\frac{\partial^2 f_1(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma} \partial J_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (\text{B.20})$$

Ainda é necessário definir as derivadas em relação à variável de endurecimento. Assim, tem-se que:

$$\frac{\partial f_2(\sigma, \kappa)}{\partial q} = \lambda \beta \exp(\beta I_1(\sigma)) + \theta - \frac{I_1(\sigma) - L(\kappa)}{\sqrt{2J_2(\sigma)R^4 + (R(I_1(\sigma) - L(\kappa)))^2}} \quad (\text{B.21})$$

$$\frac{\partial^2 f_2(\sigma, \kappa)}{\partial \sigma \partial q} = \frac{1}{R^2 \left(2J_2(\sigma) + \left(\frac{I_1(\sigma) - L(\kappa)}{R} \right)^2 \right)^{\frac{3}{2}}} \zeta(\sigma, \kappa) \quad (\text{B.22})$$

Para as outras superfícies, tem-se que estas derivadas são nulas. O *Cap Model* possui a regra de fluxo associada, dessa forma:

$$\mathbf{n}_i(\sigma, q) = \frac{\partial f_i(\sigma, q)}{\partial \sigma} \quad (\text{B.23})$$

$$\partial_\sigma \mathbf{n}_i(\sigma, q) = \frac{\partial^2 f_i(\sigma, q)}{\partial \sigma^2} \quad (\text{B.24})$$

$$\partial_q \mathbf{n}_i(\sigma, q) = \frac{\partial^2 f_i(\sigma, q)}{\partial \sigma \partial q} \quad (\text{B.25})$$

Para definir o endurecimento do material, deve-se definir $\mathbf{h}_i(\sigma, q)$ para cada uma das superfícies de plastificação. De acordo com Hofstetter, Simo e Taylor (1993) e Schwer e Murray (1994), tem-se a seguinte expressão:

$$\mathbf{h}_2(\sigma, q) = 3 \frac{\frac{\partial f_2(\sigma, \kappa)}{\partial J_2}}{\frac{\partial \varepsilon_v^p}{\partial X} \frac{\partial X(\kappa)}{\partial \kappa}} \quad (\text{B.26})$$

Desta maneira, a expressão (B.25) se torna:

$$\mathbf{h}_2(\sigma, q) = c(\sigma, \kappa)(I_1(\sigma) - L(\kappa)) \quad (\text{B.27})$$

considerando que:

$$c(\boldsymbol{\sigma}, \kappa) = \frac{3 \exp(-D(X(\kappa) - X_0))}{R^2 W D \left(1 + R(\lambda \beta \exp(\beta L(\kappa)) + \theta)\right) \sqrt{2J_2(\boldsymbol{\sigma}) + \left(\frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R}\right)^2}} \quad (\text{B.28})$$

Para as outras superfícies de plastificação, que não estão submetidas ao fenômeno de endurecimento, tem-se que $\mathbf{h}_1(\boldsymbol{\sigma}, \mathbf{q}) = \mathbf{h}_1(\boldsymbol{\sigma}, \mathbf{q}) = 0$. Por fim, tem-se que:

$$\partial_{\boldsymbol{\sigma}} \mathbf{h}_2(\boldsymbol{\sigma}, \mathbf{q}) = - \frac{c(\boldsymbol{\sigma}, \kappa)}{2J_2(\boldsymbol{\sigma}) + \left(\frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R}\right)^2} \boldsymbol{\zeta}(\boldsymbol{\sigma}, \kappa) \quad (\text{B.29})$$

$$\begin{aligned} \partial_{\mathbf{q}} \mathbf{h}_2(\boldsymbol{\sigma}, \mathbf{q}) = \mathbf{h}_2(\boldsymbol{\sigma}, \mathbf{q}) & \left(\frac{(I_1(\boldsymbol{\sigma}) - L(\kappa))}{R^2 \left(2J_2(\boldsymbol{\sigma}) + \left(\frac{I_1(\boldsymbol{\sigma}) - L(\kappa)}{R}\right)^2\right)} \right. \\ & - \frac{1}{I_1(\boldsymbol{\sigma}) - L(\kappa)} - D \left(1 + R(\lambda \beta \exp(\beta L(\kappa)) + \theta)\right) \\ & \left. - \frac{R \lambda \beta^2 \exp(\beta L(\kappa))}{1 + R(\lambda \beta \exp(\beta L(\kappa)) + \theta)} \right) \end{aligned} \quad (\text{B.30})$$

Apêndice C

Curvas Tensão-Deformação Calibradas para o *Cap Model*

Apresentam-se aqui os gráficos que comparam as curvas tensão-deformação obtidas experimentalmente e calibradas pelo programa Minerva para o *Cap Model*. Apresenta-se somente o resultado para a melhor calibração, conforme descrito na Seção 6.6. Os gráficos apresentam as três deformações principais (ε_1 , ε_2 e ε_3) e a máxima tensão principal ou tensão octaédrica de cisalhamento, definida pela seguinte expressão:

$$\tau_{oct} = \frac{1}{3} \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2} \quad (C.1)$$

em que σ_1 , σ_2 e σ_3 são as tensões principais.

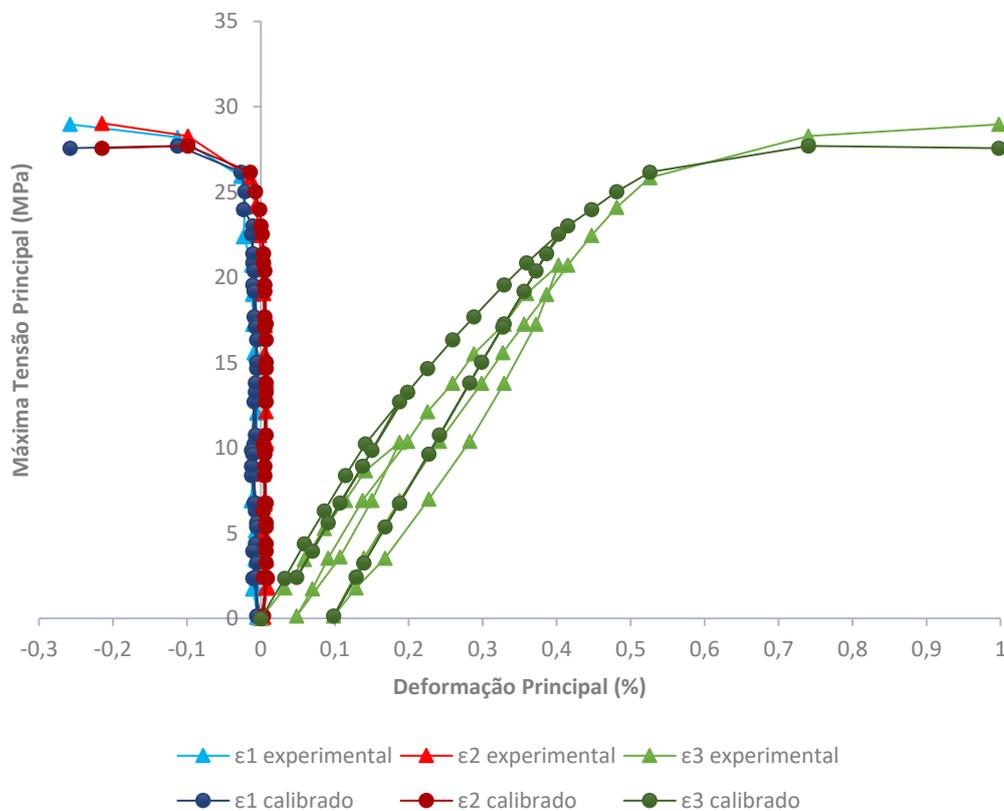


Gráfico C.1 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional

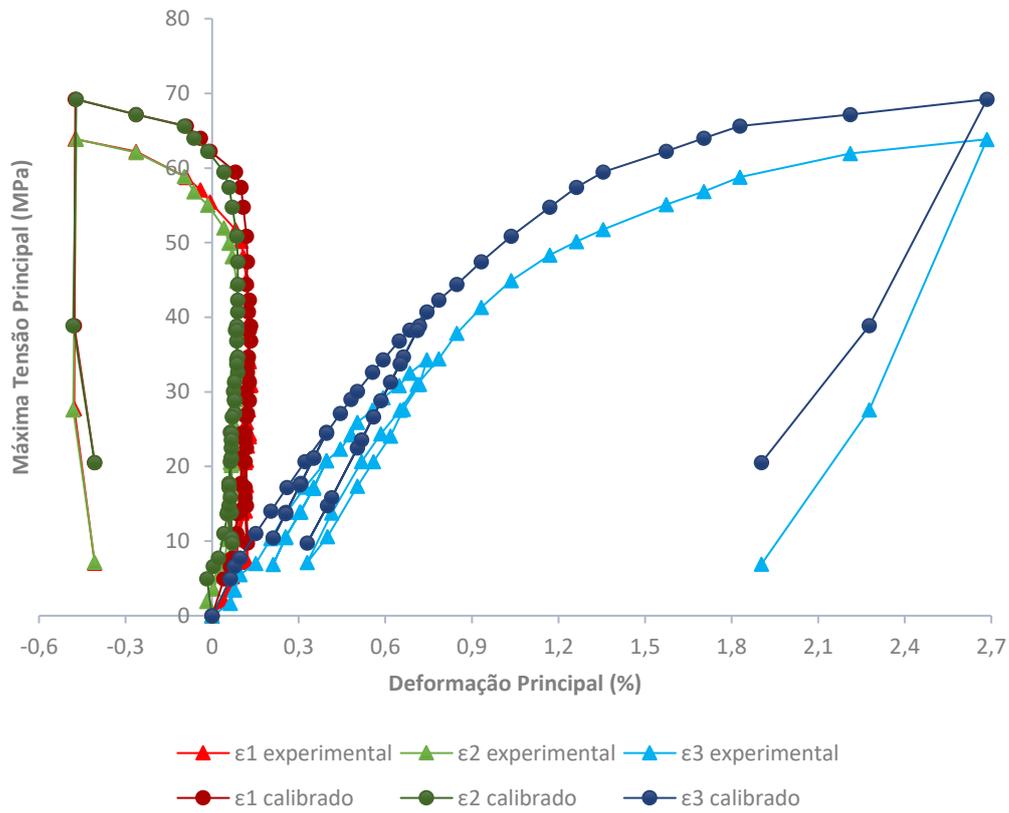


Gráfico C.2 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional

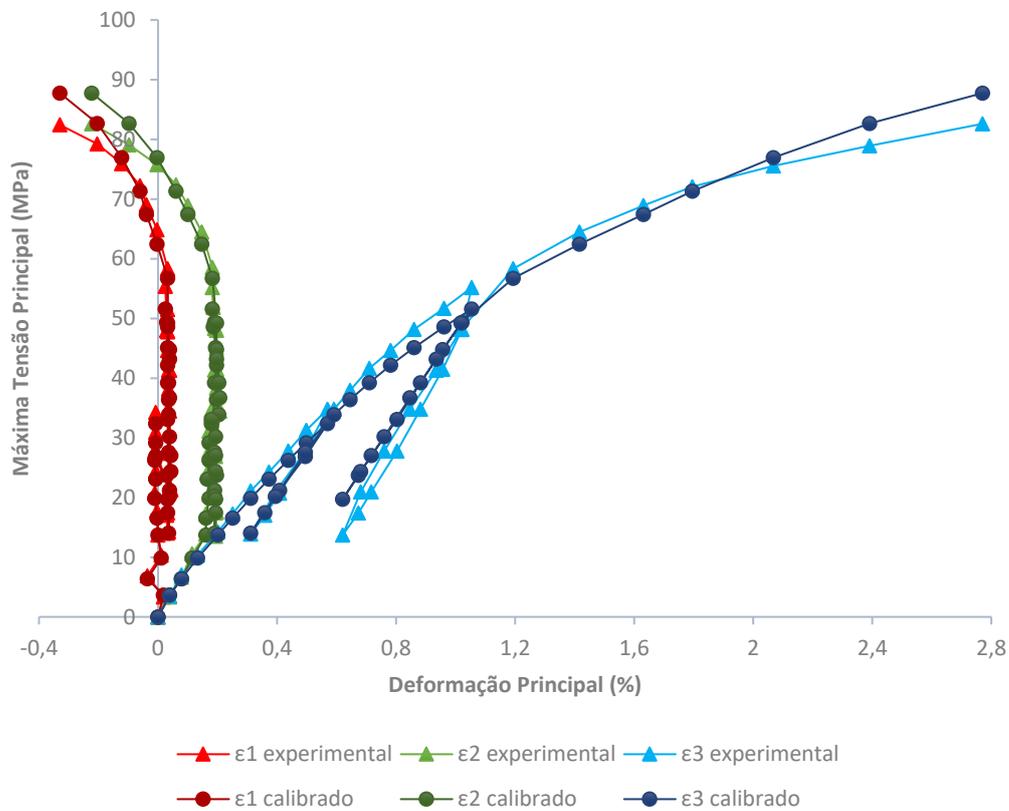


Gráfico C.3 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional

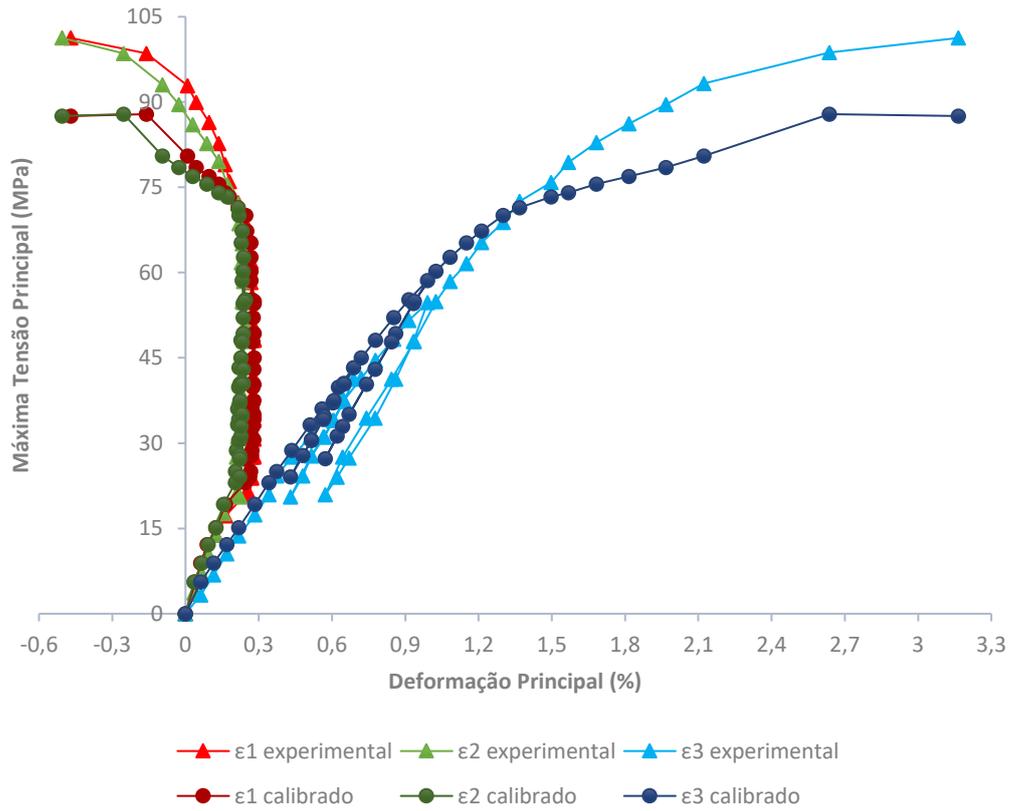


Gráfico C.4 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial Convencional

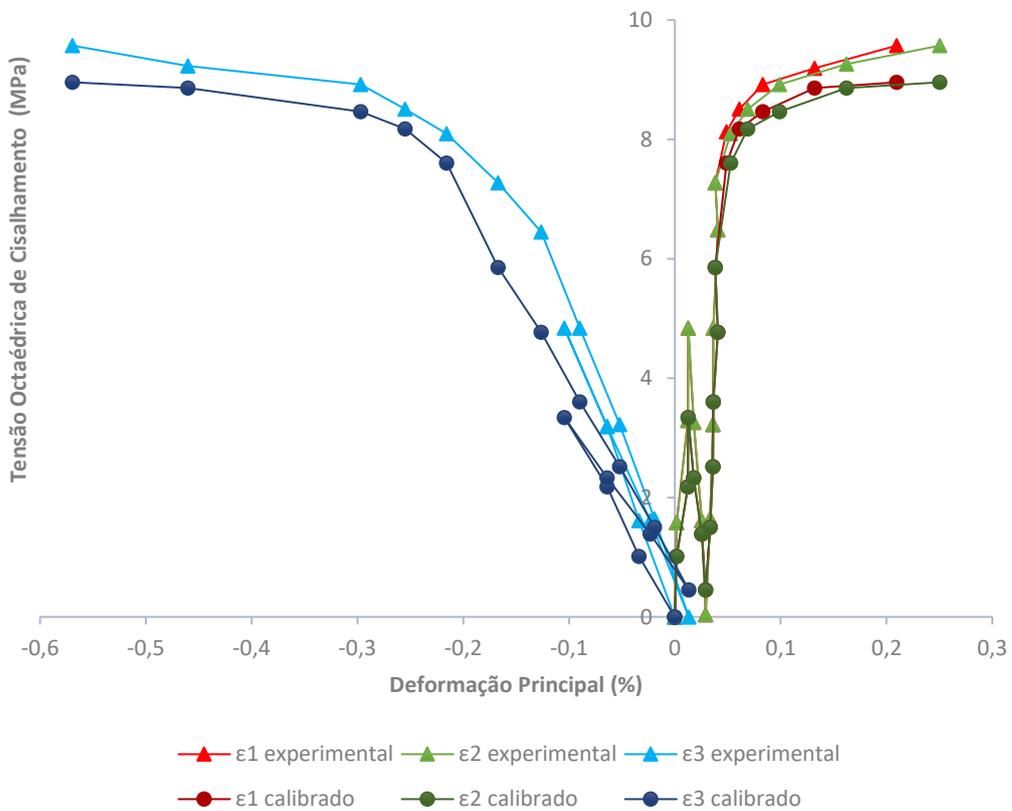


Gráfico C.5 – Curvas Tensão-Deformação para o Ensaio de Extensão Triaxial Reduzida

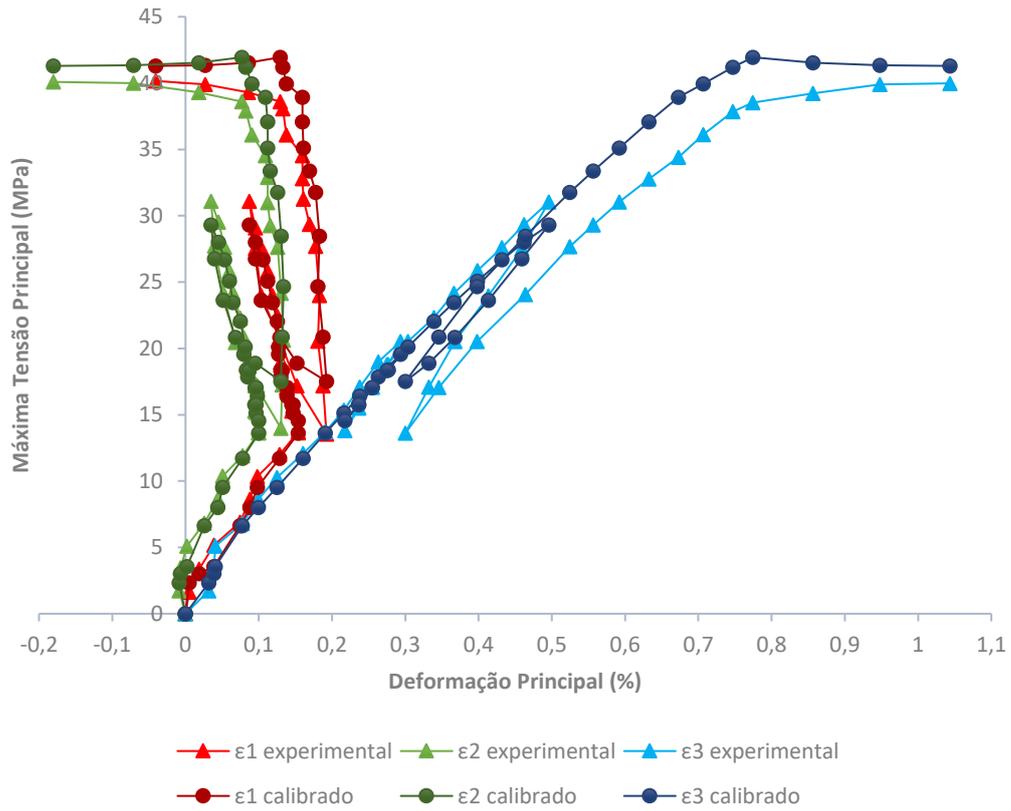


Gráfico C.6 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial

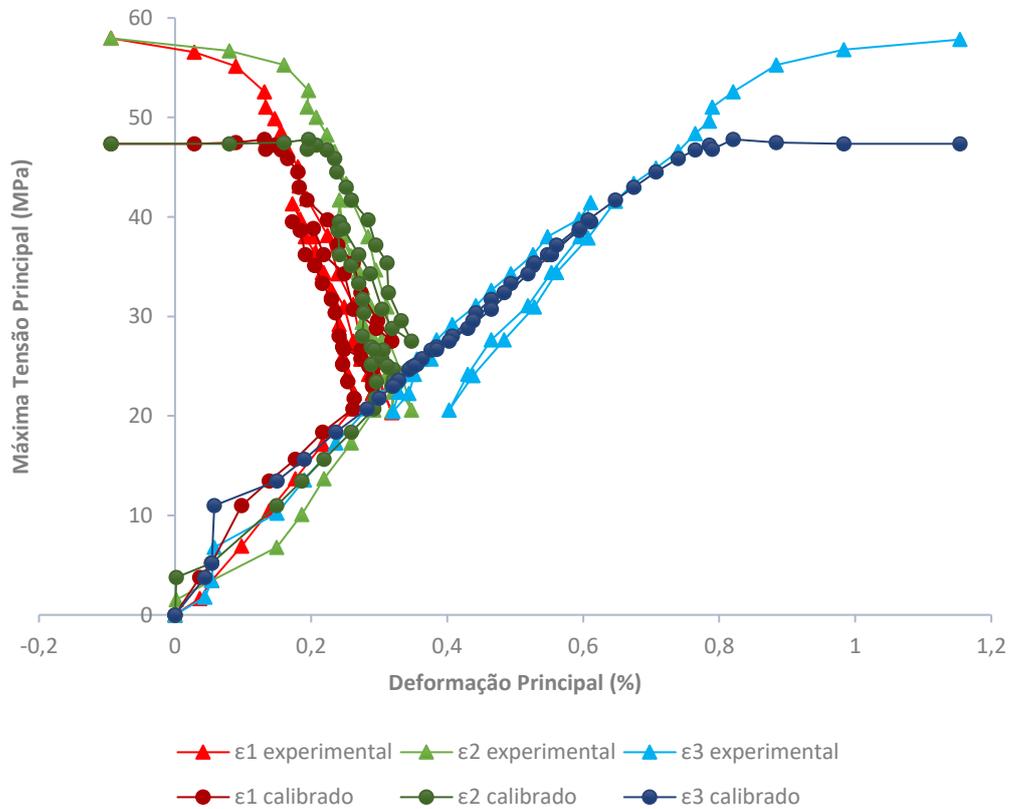


Gráfico C.7 – Curvas Tensão-Deformação para o Ensaio de Compressão Triaxial

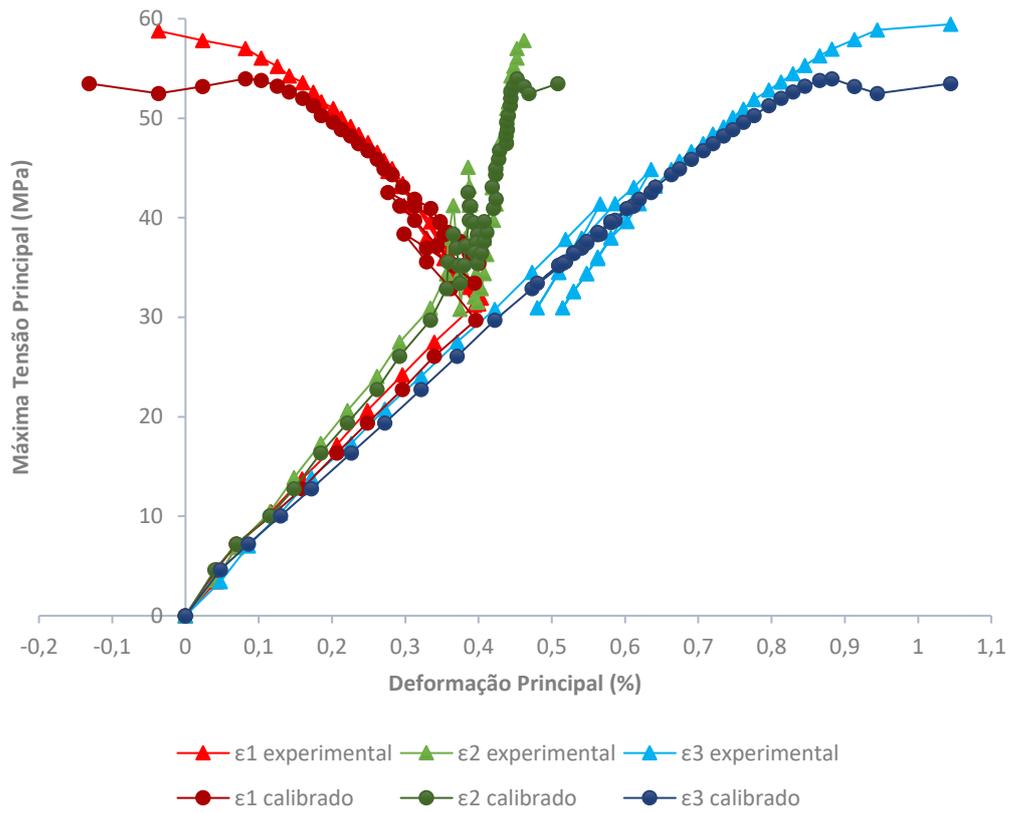


Gráfico C.8 – Curvas Tensão-Deformação para o Ensaio de Cisalhamento Simples

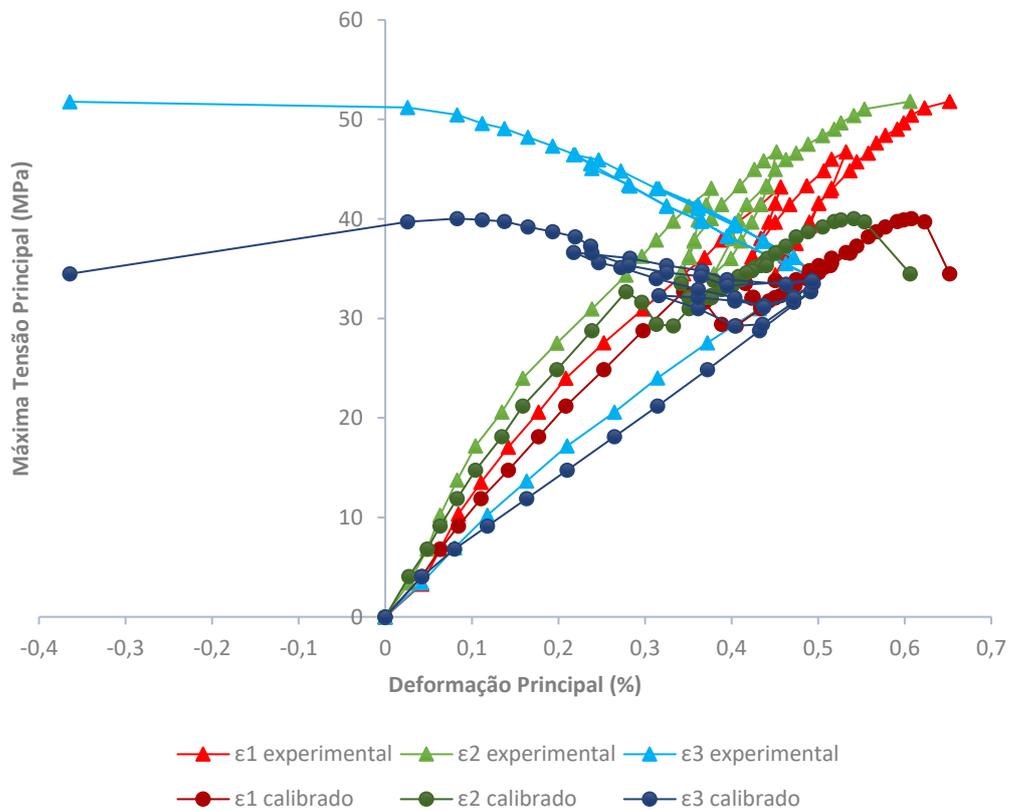


Gráfico C.9 – Curvas Tensão-Deformação para o Ensaio de Extensão Triaxial

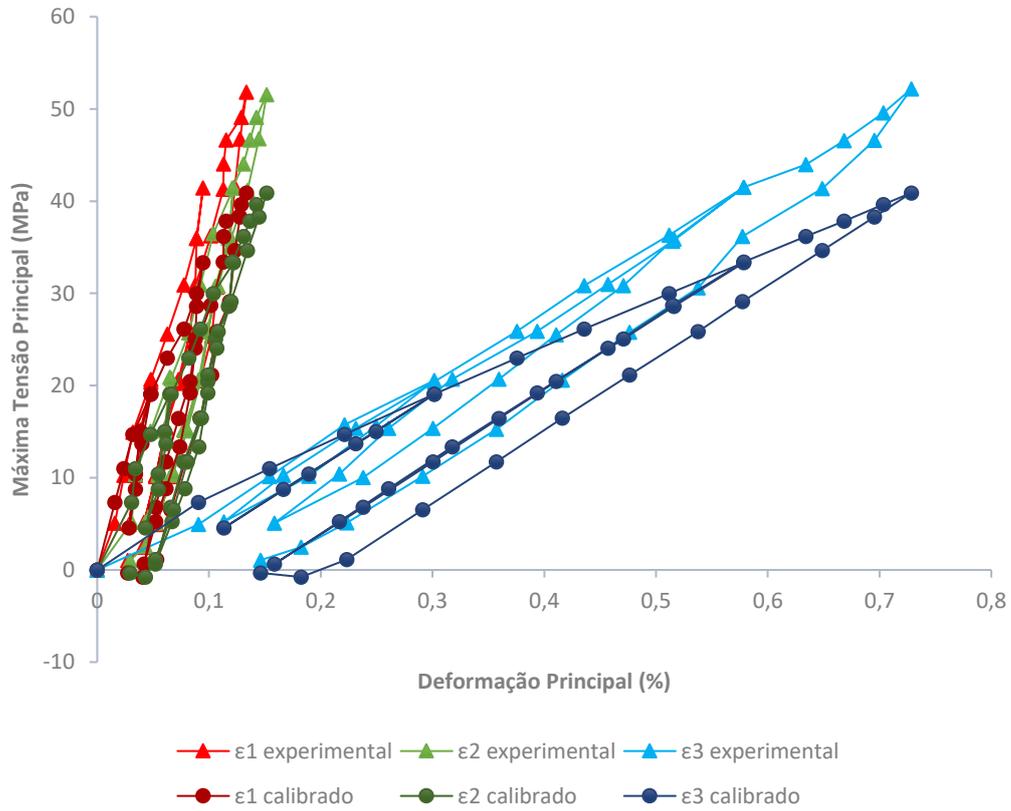


Gráfico C.10 – Curvas Tensão-Deformação para o Ensaio de Carregamento Proporcional

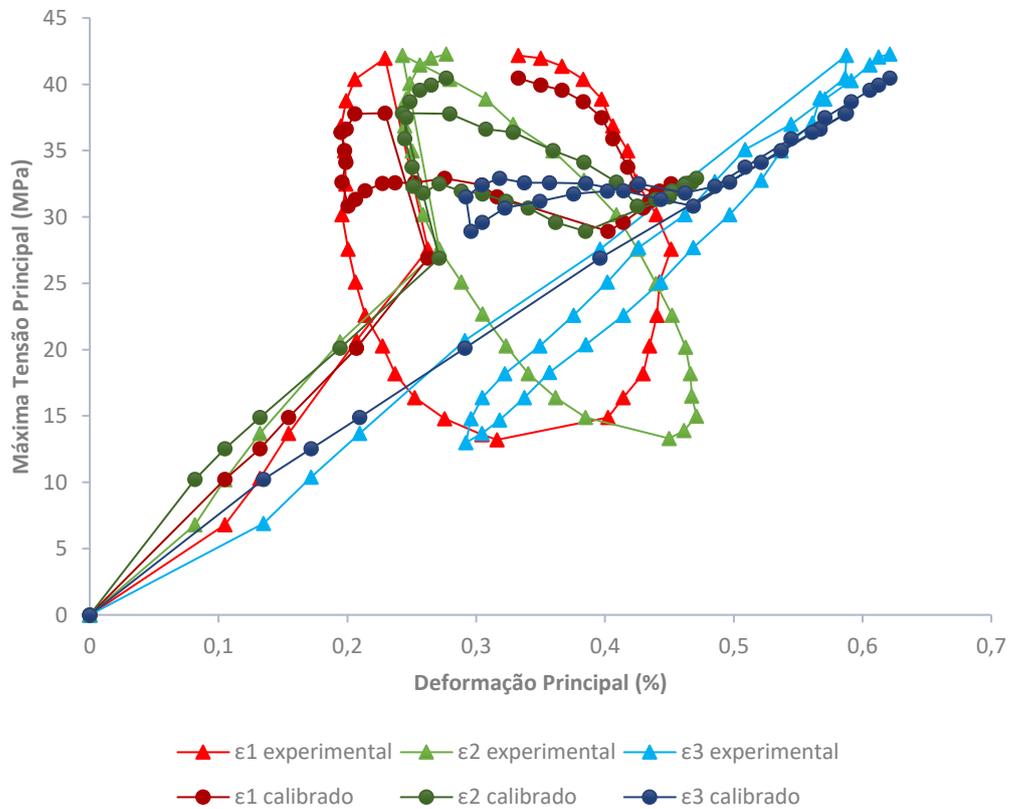


Gráfico C.11 – Curvas Tensão-Deformação para o Ensaio de Trajetória Circular de Tensões

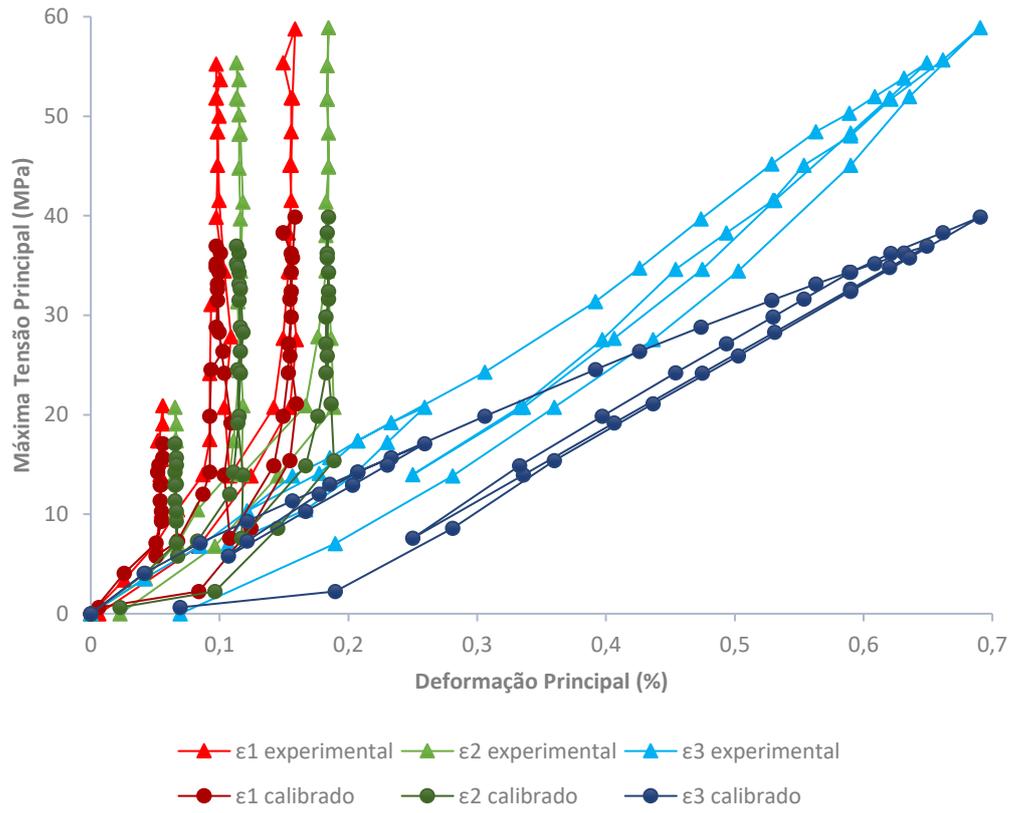


Gráfico C.12 – Curvas Tensão-Deformação para o Ensaio de Trajetória Arbitrária de Tensões