



Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Engenharia Mecânica

Pós-graduação em Engenharia Mecânica

**Desenvolvimento de um Simulador para
Sistemas com Acoplamento Elastoplástico
e Difusivo**

Felipe Augusto Cruz

Dissertação de Mestrado

Recife
2012

Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Engenharia Mecânica

Felipe Augusto Cruz

Desenvolvimento de um Simulador para Sistemas com Acoplamento Elastoplástico e Difusivo

*Trabalho apresentado ao Programa de Pós-graduação em
Engenharia Mecânica do Departamento de Engenharia
Mecânica da Universidade Federal de Pernambuco como
requisito parcial para obtenção do grau de Mestre em En-
genharia Mecânica.*

Orientador: *José Maria Andrade Barbosa*
Co-orientador: *Félix Christian Guimarães Santos*

Recife
2012

Catálogo na fonte
Bibliotecário Marcos Aurélio Soares da Silva, CRB-4 / 1175

C957d

Cruz, Felipe Augusto.

Desenvolvimento de um simulador para sistemas com acoplamento elastoplástico e difusivo / Felipe Augusto Cruz. - Recife: O Autor, 2012.

xii, 109 folhas, il., gráfs., tabs.

Orientador: Prof^o Dr^o José Maria Andrade Barbosa.

Dissertação (Mestrado) – Universidade Federal de Pernambuco.

CTG. Programa de Pós-Graduação em Engenharia Mecânica, 2012.

Inclui Referências e Apêndices.

1. Engenharia Mecânica. 2.Elemento Finito. 3.Plasticidade. 4.Simulação. 5.Multi-Físicas. I. Barbosa, José Maria Andrade (Orientador). II. Título.

621 CDD (22. ed.)

UFPE

BCTG/2012- 297

“DESENVOLVIMENTO DE UM SIMULADOR PARA SISTEMAS COM
ACOPLAMENTO ELASTOPLÁSTICO E DIFUSIVO”

FELIPE AUGUSTO CRUZ

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DO
TÍTULO DE MESTRE EM ENGENHARIA MECÂNICA

ÁREA DE CONCENTRAÇÃO: MECÂNICA COMPUTACIONAL
E PROJETO MECÂNICO
APROVADA EM SUA FORMA FINAL PELO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
MECÂNICA/CTG/EEP/UFPE

Prof. Dr. JOSÉ MARIA ANDRADE BARBOSA
ORIENTADOR/PRESIDENTE

Prof. Dr. FÉLIX CHRISTIAN GUIMARÃES SANTOS
CO-ORIENTADOR

Prof. Dr. JORGE RECARTE HENRÍQUEZ GUERRERO
COORDENADOR DO PROGRAMA

BANCA EXAMINADORA:

Prof. Dr. JOSÉ MARIA ANDRADE BARBOSA (UFPE)

Prof. Dr. FÉLIX CHRISTIAN GUIMARÃES SANTOS (UFPE)

Prof. Dr. RAMIRO BRITO WILLMERSDORF (UFPE)

Prof. Dr. LEONARDO JOSÉ DO NASCIMENTO GUIMARÃES (UFPE)

Dedico esta dissertação de mestrado aos meus pais, Sérgio Luiz Cruz e Maria Lúcia Cruz que me deram todo o apoio e suporte para eu alcançar este feito. Dedico também aos meus irmãos, namorada e amigos que tanto me ajudaram.

Agradecimentos

Primeiramente agradeço a Deus por todas as alegrias, pela saúde e pela força que me concedeu, para que eu conseguisse chegar até aqui.

Ao meu orientador, José Maria Andrade Barbosa, pela oportunidade que me foi oferecida e por todo apoio necessário à finalização desta dissertação.

Ao meu co-orientador, Félix Christian Guimarães Santos, com quem aprendi muito sobre implementação e programação, necessárias à este trabalho.

Aos professores do Departamento de Engenharia Mecânica da Universidade Federal de Pernambuco, com os quais consegui obter meu grau de graduado em engenharia mecânica.

Agradeço aos meus pais, Sérgio Luiz Cruz e Maria Lúcia Cruz por tudo que me ensinaram e por sempre me darem o apoio necessário em todas as decisões tomadas.

Aos meus irmãos, Oswaldo Cruz Neto e Fabrício Cruz, pelos momentos de descontrações e alegria. À minha irmã, Ana Aparecida Gabriel de Lima, por ter me dado dois sobrinhos maravilhosos.

Aos meus avós e avôs, tios e tias, primos e primas, por fazerem parte da minha vida.

Agradeço muito também à minha namorada, Juliana Dias Barreto, pela paciência e compreensão quando eu estive ausente em algumas ocasiões. Pelas confecções de figuras utilizadas nesta dissertação.

Aos meus amigos de infância, da escola, da graduação, da Marinha do Brasil e todos os meus amigos do LABCOM que me ajudaram a ser a pessoa que sou hoje.

Agradeço à CAPES pelo suporte financeiro oferecido como apoio para o desenvolvimento desta dissertação.

E finalmente, agradeço a todos que me ajudaram direta e indiretamente no desenvolvimento deste trabalho e que não foram citados.

*"A arte de ser louco é jamais cometer a loucura de ser um
sujeito normal"
(Raul Seixas)*

Resumo

Os simuladores voltados para mecânica computacional são um recurso tecnológico e foram construídos para preverem uma situação da realidade, ou pelo menos tentar se aproximar o máximo possível da mesma. O simulador é uma ferramenta criada que juntamente com simulações, ou seja, configurações variadas para um mesmo simulador, permite avaliar o comportamento de um processo físico que foi anteriormente implementado.

O Método do Elemento Finito é uma forma de se obter uma aproximação de um modelo matemático que descreve um comportamento físico. Este método é frequentemente utilizado na área da mecânica computacional, pois é uma importante ferramenta para solução de problemas com equações diferenciais e integrais que são frequentes nos ramos das engenharias.

O MPhyScas (Multi-Physics Multi-Scale Solver Environment) é um ambiente computacional dedicado a simular problemas de fenômenos acoplados, ou seja, um conjunto de fenômenos interagindo uns com os outros, levando em consideração o tempo e o espaço. Estes fenômenos podem ser interpretados como sendo as deformações de sólidos, transferência de calor, campos eletromagnéticos.

Algumas soluções de problemas elásticos, plásticos, transferência de calor, são conhecidas, entretanto quando estes fenômenos interagem mutuamente cria-se um novo problema. Este novo problema que envolve múltiplos fenômenos possui uma solução não mais conhecida, tornando sua solução mais complexa. Com base neste ambiente é necessário adquirir programas que possuam a capacidade de trabalhar com vários fenômenos atuantes ao mesmo tempo. O MPhyScas, por possuir uma estrutura em camadas e fornecer um ambiente dedicado a interação entre vários fenômenos, foi utilizado no desenvolvimento deste trabalho.

Esta dissertação tem o objetivo de desenvolver simuladores e simulações (utilizando o MPhyScas) para problemas multi-físicas. Inicialmente foram criados simuladores e simulações para problemas envolvendo apenas um fenômeno. Após testes realizados e comparações com as soluções analíticas criadas, foram construídos outros simuladores e configurados adequadamente para os problemas envolvendo, simultaneamente, os fenômenos de elasticidade, plasticidade e difusão.

Palavras-chave: Método do elemento finito, Multi-físicas, Plasticidade, Simulação

Abstract

Simulations focused on computational mechanics are technological resource and were built to forecast a real situation, or at least try to bring as much of it. The simulator is a tool created by different configurations that allow the evaluation of one physical process previously modeled.

The Finite Element Methods is one way to obtain an approximation of a mathematical model that describes a physical behavior of one particular process. This method is frequently used in computational mechanics methods, because is an important tool for solving problems with differential and integral equations which are very common in engineering areas.

MPhyScas (Multi-Physics Multi-Scale Solver Environment) is a computational environment dedicated to simulations of coupled phenomena problems, in other words, a set of phenomena interacting one to another, and considering time and space effects. This phenomena can be interpreted, for example, as deformations in solids, heat transfer and electromagnetic fields.

Solutions for elastic, plastic and heat transfer problems, for example, are well know, but a new problem arise when all these phenomena are interacting one to another. The solution for problems with high levels of complexity that involves interactions among multiple phenomena, is not clear and represents one of the major challenges in computational mechanics . Based on this environment , there is a need for software that has the ability to work with several phenomena acting simultaneously. MPhyScas was used to the analysis in this project because it presents layer structure and is dedicated to the set of phenomena interactions.

The aim of this dissertation is to develop simulators and simulations (using MPhyScas environment) for multi-physics problems. Simulators and simulations were initially created for problems involving only one phenomenon. After the tests and comparisons with the analytical solutions, other simulators were built and accordingly configured to the problems involving elasticity, plasticity and diffusion phenomena.

Keywords: Finite element method, Multi-physics, Plasticity, Simulation

Lista de Figuras

2.1	Representação geométrica da elipse no campo de tensões	5
2.2	Aplicação da Tensão e incremento	7
2.3	Descarregamento após plastificação	8
2.4	Condição de Plastificação I	8
2.5	Condição de Plastificação II	8
2.6	Endurecimento Isotrópico	9
2.7	Tensão x Deformação	10
2.8	Endurecimento Cinemático	11
3.1	Esquema de solução	16
3.2	Problema Acoplado	21
4.1	Representação do Simulador e da Simulação	25
4.2	Simulador - Construção do Corpo	26
4.3	Construção do <i>Kernel</i>	26
4.4	Construção do Block	27
4.5	Construção do <i>Group</i>	28
4.6	Construção do Phenomenon	28
4.7	Simulação - Construção do Corpo	29
4.8	Construção da geometria	30
5.1	Estrutura Tridimensional	33
5.2	Estrutura Bidimensional	34
5.3	Estrutura Unidimensional	34
5.4	Estrutura de Dimensão Zero	34
5.5	Malha Bidimensional	35
5.6	Grafo de uma Malha Geométrica Bidimensional	36
6.1	Placa com Refinamento de 0.25	38
6.2	Placa com Refinamento de 0.125	39
6.3	Placa com Refinamento de 0.05	39
6.4	Placa com Furo e Refinamento de 0.25	40
6.5	Placa com Furo e Refinamento de 0.125	40
6.6	Placa com Furo e Refinamento de 0.05	41
6.7	Problema para o Simulador Elástico	42
6.8	Problema para o Simulador Difusivo	44

6.9	Problema Elástico Difusivo - Configuração	45
6.10	Problema Elástico Difusivo - Concentração do soluto	46
6.11	Problema Elástico Difusivo - Tensão de Von Mises	46
6.12	Problema Elástico Difusivo - Tensão Hidrostática	46
6.13	Problema Elástico Difusivo - Estado Final	47
6.14	Problema Elastoplástico com Difusão - Concentração do soluto	47
6.15	Problema Elastoplástico com Difusão - Tensão de Von Mises	48
6.16	Problema Elastoplástico com Difusão - Tensão Hidrostática	48
6.17	Problema Elastoplástico com Difusão - Deformação Plástica Acumulada	48

Lista de Tabelas

3.1	Legenda da Figura 3.2	21
6.1	Valores para a geometria da placa	39
6.2	Valores para a geometria da placa com furo interno	41
6.3	Parâmetros para Condição de Contorno	42
6.4	Parâmetros para os Resolvedores	43
6.5	Erros para o problema do Simulador Elástico	43
6.6	Resolvedor para Problema Difusivo - Resultados	44
6.7	Características para o Problema Difusivo	45
6.8	Erros para o problema do Simulador Difusivo	45
6.9	Valores Máximos e Mínimos para o problema Elástico Difusivo	46
6.10	Valores Máximos e Mínimos para o problema Elastoplástico com Difusão	48

Sumário

1	Introdução	1
1.1	Objetivos	2
1.2	Organização do Trabalho	2
2	Modelo Elastoplástico e Difusivo	3
2.1	Modelo Elástico	3
2.2	Tensão de Von Mises	4
2.2.1	Decomposição do tensor de tensões	5
2.3	Deformação Plástica Acumulada	6
2.4	Análise da Função de Escoamento	7
2.5	Condição de Consistência	9
2.6	Endurecimento Isotrópico e Cinemático	9
2.6.1	Endurecimento Isotrópico	9
2.6.2	Endurecimento Cinemático	10
2.7	Modelo Difusivo	11
2.8	Equações para o modelo Elastoplástico com Difusão	13
3	Implementação utilizando o MPhyScas	15
3.1	Problema Elastoplástico com Difusão	15
3.2	Algoritmos	16
3.2.1	Algoritmo de nível I	16
3.2.2	Algoritmo de nível II	19
3.2.3	Algoritmo de nível III	20
3.3	Implementação no MPhyScas	20
3.3.1	<i>Kernel</i>	22
3.3.2	<i>Block</i>	22
3.3.3	<i>Group</i>	22
3.3.4	<i>Phenomenon</i>	23
4	Construção do Simulador e Simulação	25
4.1	Construção do Simulador - Corpo	25
4.1.1	Construção do Simulador - Corpo - <i>Kernel</i>	26
4.1.2	Construção do Simulador - Corpo - <i>Block</i>	27
4.1.3	Construção do Simulador - Corpo - <i>Group</i>	27
4.1.4	Construção do Simulador - Corpo - <i>Phenomenon</i>	28
4.2	Construção da Simulação	29

4.2.1	Construção da Simulação - Corpo - Geometria	30
4.2.2	Construção da Simulação - Corpo - <i>Kernel</i>	30
4.2.3	Construção da Simulação - Corpo - <i>Block</i>	30
4.2.4	Construção da Simulação - Corpo - <i>Group</i>	31
4.2.5	Construção da Simulação - Corpo - <i>Phenomenon</i>	31
5	Malha Geométrica	32
5.1	Dimensão da Estrutura e Espaço	32
5.1.1	Espaço de Dimensão Três	33
5.1.2	Espaço de Dimensão Dois	33
5.1.3	Espaço Dimensão Um	34
5.1.4	Espaço de Dimensão Zero	34
5.2	Construção da Malha Bidimensional	34
5.2.1	Malha de Triângulos (<i>GeomMesh_Tri</i>)	36
5.2.2	Malha Linha (<i>GeomMesh_Line</i>)	37
5.2.3	Malha Ponto (<i>GeomMesh_Pointer</i>)	37
6	Resultados e Análises	38
6.1	Malhas Geométricas	38
6.1.1	Malha da Placa	38
6.1.2	Malha da Placa com Furo Interno	40
6.2	Simulador para o Problema Elástico	41
6.3	Simulador para o Problema Difusivo	43
6.4	Problema Elástico e Difusivo	45
6.5	Simulador para Problema Elastoplástico com Difusão	47
7	Conclusão e Trabalhos Futuros	49
7.1	Conclusões	49
7.2	Trabalhos Futuros	50
A	Formulação Numérica do Problema ElastoPlástico com Difusão	53
B	Solução exata para o Problema Elástico	61
C	Solução exata para o Problema Difusivo	62
D	Desenvolvimento da Variação da Plasticidade	63
E	Geometria para uma Placa Bidimensional	65
F	Geometria para uma Placa Bidimensional com Furo Interno	67
G	Desenvolvimento do Simulador Elastoplástico com Difusão	70
H	Desenvolvimento do Simulador Elastoplástico com Difusão	96

CAPÍTULO 1

Introdução

Sempre houve a existência de problemas que atuam mutuamente e com o avanço da engenharia esta divisão de problemas e suas interações têm sido estudada com frequência, alguns destes problemas foram estudados em: [4], [25], [24], [5], [19] entre outros. Devido a esta evolução foi capaz de construir simuladores e criar diversas simulações envolvendo estes fenômenos. Juntando-se a isto, esquemas numéricos estão disponibilizados em diversas literaturas, pois para um mesmo problema há diversas combinações existentes.

Os simuladores [9], [7] voltados para mecânica computacional são um recurso tecnológico e foram construídos para preverem uma situação da realidade, ou pelo menos tentar se aproximar o máximo possível da mesma. O simulador é uma ferramenta criada que juntamente com simulações, ou seja, configurações variadas para um mesmo simulador, permite avaliar o comportamento de um processo físico que foi anteriormente modelado.

O Método do Elemento Finito (MEF) [14], [20] é uma forma de se obter uma aproximação de um modelo matemático que descreve um comportamento físico. Este método está frequentemente sendo utilizado na área da mecânica computacional, pois é uma importante ferramenta para solução de problemas com equações diferenciais e integrais que são frequentes nos ramos das engenharias.

O MPhyScas (Multi-Physics Multi-Scale Solver Environment) é um ambiente computacional dedicado a simular problemas de fenômenos acoplados, ou seja, um conjunto de fenômenos interagindo uns com os outros, levando em consideração o tempo e o espaço. Estes fenômenos podem ser interpretados como sendo as deformações de sólidos [3], transferência de calor [13], campos eletromagnéticos, entre outros.

Ele fornece um ambiente que é suficientemente flexível para acomodar todos os níveis computacionais que podem ser encontrados em simuladores baseados no método dos elementos finitos. Eles são estudados e representados em uma arquitetura de camadas.

A fragilização pelo hidrogênio [11], [8] é um problema que afeta o comportamento mecânico dos metais, pois o hidrogênio altera as propriedades dos metais. O hidrogênio se desloca na rede cristalina gerando um enriquecimento da concentração localizada, tornando o material mais sensível à fragilização. Quando um metal está sendo fabricado devem ser levados em conta os carregamentos mecânicos (fenômeno elastoplástico) sofridos pelo mesmo, assim, facilitando o movimento do hidrogênio (fenômeno difusivo) no interior da rede cristalina.

1.1 Objetivos

Algumas soluções de problemas como: elasticidade [15], plasticidade [6], [23], transferência de calor, entre outros, é conhecida, porém quando estes fenômenos interagem uns com os outros é criado um novo problema. Este novo problema que envolve múltiplos fenômenos possui uma solução não mais conhecida, tornando sua solução mais complexa. Com base neste ambiente é necessário adquirir programas que possuam a capacidade de trabalhar com vários fenômenos atuantes ao mesmo tempo. O MPhyScas [17], [18], [16] por possuir uma estrutura em camadas e fornecer um ambiente dedicado a interação entre vários fenômenos, foi utilizado no desenvolvimento deste trabalho.

Esta dissertação tem o objetivo de desenvolver simuladores e simulações (utilizando o MPhyScas) para problemas multi-físicas. Inicialmente foram criados simuladores e simulações para problemas envolvendo apenas um fenômeno. Após testes realizados e comparações com as soluções analíticas criadas, foram construídos outros simuladores e configurados adequadamente para os problemas envolvendo, simultaneamente, os fenômenos de elasticidade, plasticidade e difusão.

1.2 Organização do Trabalho

Esta dissertação divide-se em sete capítulos. Neste primeiro capítulo foi apresentada uma breve visão fazendo a contextualização geral do trabalho e foram mostradas uma motivação pela qual o trabalho foi desenvolvido.

Capítulo 2 - Modelo Elastoplástico e Difusivo - são apresentados modelos que serão utilizados para o desenvolvimento do termo de acoplamento do problema multi-física elastoplástico com difusão.

Capítulo 3 - Implementação utilizando o MPhyScas - são apresentados os processos e as divisões feitas em um problema acoplado mecânico-difusivo, além de ser explicitado as funções de cada camada do MPhyScas relacionada com o problema em questão.

Capítulo 4 - Construção do Simulador e Simulação - é apresentada a descrição detalhada sobre o arquivo do Simulador e da Simulação. Nestes arquivos são feitas as configurações de cada uma das camadas do MPhyScas, além disto, têm-se o carregamento do arquivo da geometria.

Capítulo 5 - Malha Geométrica - são apresentadas as definições e a organização entre as estruturas do MPhyScas relacionado as malhas geométricas sendo citados alguns tipos de malhas geométricas implementadas no mesmo.

Capítulo 6 - Resultados e Análises - são apresentados alguns resultados obtidos a partir de simulações realizadas. Simulações numéricas do problema de elasticidade, do problema de difusão e do problema acoplado envolvendo elastoplasticidade com difusão.

Capítulo 7 - Conclusões e Trabalhos Futuros - são feitas conclusões e considerações finais sobre o trabalho apresentado. Neste capítulo também são apresentados propostas e desafios para possíveis trabalhos futuros.

Modelo Elastoplástico e Difusivo

Este capítulo apresenta as equações que modelam o comportamento elastoplástico com difusão utilizados nesta dissertação. Para isto é necessário expor cada um dos modelos envolvidos tratando-se de um modelo bidimensional com estado de tensão plana para pequenas deformações.

Inicialmente tem-se o modelo elástico utilizado para o desenvolvimento deste trabalho para o caso de um material isotrópico elástico linear juntamente com o critério da tensão de Von Mises, muito utilizada para materiais metálicos, é possível definir e identificar as regiões elástica e plástica nas quais o material está suscetível a permanecer.

A deformação plástica acumulada é responsável por armazenar as informações adquiridas pelo material durante a sua vida e juntamente com análises da função de escoamento, condições de Kuhn-Tucker e consistência são abordados neste capítulo.

O modelo elastoplástico com difusão envolve endurecimentos isotrópico e cinemático. O endurecimento isotrópico é da forma não linear e diz respeito à um aumento da tensão de escoamento do material. Através do endurecimento cinemático verifica-se que curva de escoamento se translada no espaço das tensões, mantendo o seu formato, tamanho e a orientação da curva inicial.

Após estas abordagens, o modelo difusivo é apresentado e uma finalização das equações para o modelo elastoplástico com difusão é então exposta.

2.1 Modelo Elástico

O comportamento de um material submetido a este modelo pode ser descrito pela lei de Hooke [15], onde as tensões são determinadas pelas deformações.

$$\sigma = C \cdot \varepsilon^e \quad (2.1)$$

onde:

- σ , é o tensor das tensões;
- C , é a matriz de rigidez elástica do material;
- ε^e , são as deformações elásticas.

Tomando-se como base a equação da lei geral de Hooke expressa em 2.1, encontra-se que para o caso de um material isotrópico elástico linear [15], a equação da tensão relacionado com a deformação possui a seguinte relação:

$$\sigma = 2\mu\varepsilon^e + \lambda tr(\varepsilon^e) \mathbf{I} \quad (2.2)$$

onde λ é a constante de Lamé e μ é o módulo de cisalhamento.

Assumindo que a deformação total (ε) do material é composta por parcelas de algumas deformações, faz-se a decomposição aditiva da deformação nas seguintes parcelas: elástica (ε^e), plástica (ε^p) e de dilatação devido ao processo difusivo (ε^c). Logo, tem-se:

$$\varepsilon = \varepsilon^e + \varepsilon^p + \varepsilon^c \longrightarrow \varepsilon^e = \varepsilon - \varepsilon^p - \varepsilon^c \quad (2.3)$$

Por questões de implementação o tensor de tensões está representado por (**S**). Reescrevendo o tensor das tensões e substituindo a deformação elástica encontrada após a decomposição aditiva obtém-se:

$$\mathbf{S} = 2\mu (\varepsilon - \varepsilon^p - \varepsilon^c) + \lambda \text{tr}((\varepsilon - \varepsilon^p - \varepsilon^c)) \mathbf{I} \quad (2.4)$$

2.2 Tensão de Von Mises

O comportamento unidimensional elástico de um material metálico é definido sempre que a tensão aplicada for menor que a tensão de escoamento. Neste tipo de comportamento, verifica-se que após ser retirada a tensão aplicada, o material não apresenta deformações, isto é, possui a mesma configuração inicial.

O comportamento unidimensional plástico de um material metálico é definido sempre que a tensão aplicada for igual ou maior que a tensão de escoamento. Neste tipo de comportamento, verifica-se que após ser retirada a tensão aplicada, o material apresenta deformações residuais, ou seja, não possui a mesma configuração inicial.

A tensão efetiva (Von Mises σ_e) em termos das tensões principais para um problema tridimensional ($\sigma_1, \sigma_2, \sigma_3$) está apresentado abaixo:

$$\sigma_e = \frac{1}{\sqrt{2}} \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2} \quad (2.5)$$

Como o problema em questão foi utilizado tensão plana ($\sigma_3 = 0$), reescreve-se a Equação 2.5, então:

$$\frac{1}{\sqrt{2}} \sqrt{(\sigma_1 - \sigma_2)^2 + \sigma_2^2 + \sigma_1^2} = \sigma_e \quad (2.6)$$

então:

$$\sigma_1^2 - \sigma_1 \sigma_2 + \sigma_2^2 = \sigma_e^2 \quad (2.7)$$

A Equação 2.7 está representada geometricamente pela Figura 2.1, para uma tensão de escoamento fixa diferente de zero. Através desta figura identifica-se uma região no interior da

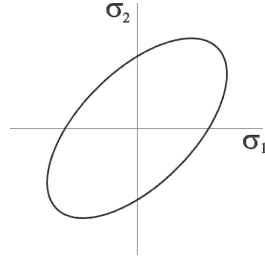


Figura 2.1 Representação geométrica da elipse no campo de tensões

elipse que representa a região elástica do material. Sobre a curva da elipse há a plastificação do material e para esta curva dá-se o nome de superfície de plastificação [22].

$$f = \|\mathbf{S}_{dev}\| - \sigma_y \quad (2.8)$$

onde:

- f , é a superfície de plastificação;
- $\mathbf{S}_{dev}$, é o tensor desviador da tensão;
- σ_y , é a tensão de escoamento do material.

A norma do tensor de tensões abordada acima possui a seguinte característica:

$$\|\mathbf{A}\| = \sqrt{\frac{3}{2} \mathbf{A} : \mathbf{A}} \quad (2.9)$$

onde o produto interno está expresso abaixo:

$$\mathbf{A} : \mathbf{A} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} : \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} = \sum_{i,j=0}^2 a_{ij} \cdot a_{ij} = \quad (2.10)$$

$$= a_{00} \cdot a_{00} + a_{01} \cdot a_{01} + a_{02} \cdot a_{02} + a_{10} \cdot a_{10} + a_{11} \cdot a_{11} + a_{12} \cdot a_{12} + a_{20} \cdot a_{20} + a_{21} \cdot a_{21} + a_{22} \cdot a_{22}$$

As análises da Equação 2.8 estão apresentadas na seção 2.4

2.2.1 Decomposição do tensor de tensões

Um estado qualquer de tensões pode ser dividido linearmente em duas componentes, sendo elas apresentadas abaixo:

- σ_H , representa o tensor de tensão hidrostático;
- σ_{dev} , representa o tensor desviador.

A componente hidrostática (σ_H) é composta unicamente por tensões normais, ou seja, parcelas que somente produzem alterações de volume. A outra componente é conhecida como desviadora (σ_{dev}) e suas componentes cisalhantes são responsáveis pela rotação, uma vez que as tensões normais são decrescidas da parcela do tensor de tensão hidrostática.

A decomposição do tensor tensão está apresentada na Equação 2.11.

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_H + \boldsymbol{\sigma}_{dev} \quad (2.11)$$

Para o caso tridimensional o tensor desviador está expresso na Equação 2.13 e diz respeito ao tensor tensão total retirando-se a parcela da tensão hidrostática (Equação 2.12). A parcela relacionada ao tensor desviador é responsável pela mudança de forma e a parcela do tensor hidrostático é responsável pela variação do volume.

$$\boldsymbol{\sigma}_H = \frac{1}{3} \text{tr}(\boldsymbol{\sigma}) \mathbf{I} \quad (2.12)$$

$$\boldsymbol{\sigma}_{dev} = \boldsymbol{\sigma} - \frac{1}{3} \text{tr}(\boldsymbol{\sigma}) \mathbf{I} \quad (2.13)$$

Para o estado de tensão plana as Equações 2.12 e 2.13 são representadas por:

$$\boldsymbol{\sigma}_H = \frac{1}{2} \text{tr}(\boldsymbol{\sigma}) \mathbf{I} \quad (2.14)$$

$$\boldsymbol{\sigma}_{dev} = \boldsymbol{\sigma} - \frac{1}{2} \text{tr}(\boldsymbol{\sigma}) \mathbf{I} \quad (2.15)$$

Explicitando as tensões para o estado plano de tensões, tem-se:

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\sigma_{11}+\sigma_{22}}{2} & 0 \\ 0 & \frac{\sigma_{11}+\sigma_{22}}{2} \end{bmatrix}}_{\boldsymbol{\sigma}_H} + \underbrace{\begin{bmatrix} \frac{\sigma_{11}-\sigma_{22}}{2} & \sigma_{12} \\ \sigma_{21} & \frac{\sigma_{22}-\sigma_{11}}{2} \end{bmatrix}}_{\boldsymbol{\sigma}_{dev}} \quad (2.16)$$

Esta decomposição do tensor representada na Equação 2.16 é muito útil para o cálculo do desviador do tensor de tensão utilizado no modelo elastoplástico com difusão. Com a apresentação do modelo elástico, a próxima seção trata de conceitos relacionadas à deformação plástica.

2.3 Deformação Plástica Acumulada

A deformação plástica acumulada é a responsável por guardar as informações adquiridas pelo material durante a sua vida e é um escalar positivo e pode ser expresso pela integral representada na Equação 2.17.

$$p = \int \|\dot{\epsilon}^p\| dt \quad (2.17)$$

onde, p representa a deformação plástica acumulada e $\dot{\epsilon}^p$ é a taxa de deformação plástica.

Na plasticidade associativa, tem-se que a direção de plastificação $\mathbf{N}$, segue a direção normal da superfície de plastificação f , no campo de tensões $\boldsymbol{\sigma}$ sendo representada pela equação:

$$\mathbf{N} = \frac{\partial f}{\partial \boldsymbol{\sigma}} \quad (2.18)$$

A lei da normalidade estabelece que o incremento do tensor da deformação plástica é diretamente proporcional ao gradiente da função de plastificação ou de escoamento relativo ao campo de tensões.

$$d\boldsymbol{\varepsilon}^p = d\alpha \cdot \left(\frac{\partial f}{\partial \boldsymbol{\sigma}} \right) = d\alpha \cdot \mathbf{N} \quad (2.19)$$

onde $d\alpha$ é o multiplicador plástico.

2.4 Análise da Função de Escoamento

Através da Equação 2.8, pode-se obter duas configurações diferentes. A primeira diz respeito ao estado elástico ($f < 0$, região interna à elipse) e a segunda trata do estado plástico ($f = 0$, região sobre a curva da elipse).

Para o caso de elasticidade, basta que a função de plastificação seja um número negativo, ou seja, a tensão aplicada ao material não atinja o limite de escoamento atual do mesmo. O limite de escoamento atual é justamente devido ao possível endurecimento sofrido anteriormente pelo material, no caso de materiais não virgens. Quando a função de plastificação for zero identifica-se escoamento do material, que está representado pelo ponto de encontro da superfície de escoamento ($f = 0$) com o destino do vetor $\boldsymbol{\sigma}$ (ver Figura 2.2). A partir desta configuração há alguns casos a serem analisados.

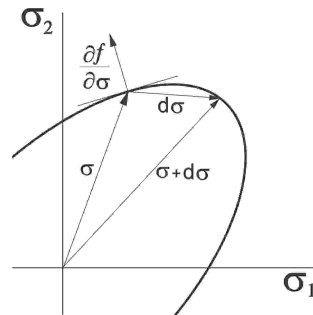


Figura 2.2 Aplicação da Tensão e incremento

Primeiramente considera-se que o material não irá mais escoar, ou seja, sofrerá descarregamento, em outras palavras $\frac{\partial f}{\partial \sigma} \cdot d\sigma < 0$. Isto quer dizer que o ângulo formado entre a normal exterior à superfície e o incremento da tensão é maior que $\frac{\pi}{2}$, estando representado na Figura 2.3 onde o material está sendo descarregado.

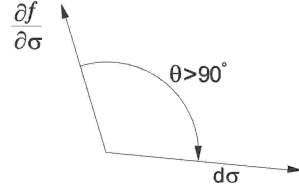


Figura 2.3 Descarregamento após plastificação

Para que o material continue escoando é necessário que no próximo instante de tempo de $f = 0$, o sinal de $\frac{\partial f}{\partial \sigma} \cdot d\sigma > 0$. Isto quer dizer que o gradiente da função de plastificação forma um ângulo agudo com o vetor de incremento de tensão, exposto na Figura 2.4, dando continuidade ao escoamento.

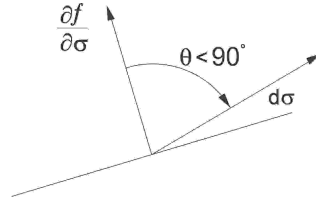


Figura 2.4 Condição de Plastificação I

O próximo caso de plastificação ocorre quando $f = 0$ e $\frac{\partial f}{\partial \sigma} \cdot d\sigma \leq 0$, então o valor do multiplicador plástico ($d\alpha = 0$) deve ser necessariamente zero. Na Figura 2.5 verifica-se o ângulo reto formado entre os dois vetores, só que para a atual configuração o vetor $\frac{\partial f}{\partial \sigma}$ possui módulo nulo, ou seja, permanece no mesmo ponto (na superfície de plastificação).

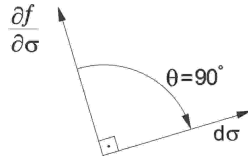


Figura 2.5 Condição de Plastificação II

As condições de Kuhn-Tucker [22] são condições necessárias, mas não suficientes para haver plastificação e estão apresentadas abaixo:

$$\begin{cases} f \leq 0 \\ \dot{\alpha} \geq 0 \\ f\dot{\alpha} = 0 \end{cases} \quad (2.20)$$

Para complementar as condições de Kuhn-Tucker representadas pela Equação 2.20, a próxima seção trata da condição de consistência que juntamente com as condições de Kuhn-Tucker garantem que um estado de plastificação será mantido, ou seja, o ponto onde houve a plastificação se manterá plastificando no próximo instante.

2.5 Condição de Consistência

A condição de consistência [22] é conhecida por exigir que o ponto de carregamento permaneça na superfície de escoamento ($f = 0$) durante a deformação plástica e só assim determinar o multiplicador plástico ($\dot{\alpha}$). Ela pode ser expressa pela seguinte Equação 2.21:

$$f(\mathbf{S}_{\text{dev}} + d\mathbf{S}_{\text{dev}}, \mathbf{X} + d\mathbf{X}, Y + dY) = \frac{\partial f}{\partial \mathbf{S}_{\text{dev}}} : d\mathbf{S}_{\text{dev}} + \frac{\partial f}{\partial \mathbf{X}} : d\mathbf{X} + \frac{\partial f}{\partial Y} \cdot dY = 0 \quad (2.21)$$

Verifica-se então que a condição é escrita para uma mudança incremental da tensão e dos endurecimentos cinemático ($\mathbf{X}$) e isotrópico (Y).

Fazendo as análises da condição de Kuhn-Tucker, tem-se:

$$f\dot{\alpha} = 0 \begin{cases} \dot{\alpha} > 0 & \text{e} & f = 0 \\ \dot{\alpha} = 0 & \text{e} & f < 0 \end{cases} \quad (2.22)$$

2.6 Endurecimento Isotrópico e Cinemático

Para os metais os endurecimentos alteram a curva de plastificação no campo de tensões, como pode ser observado na Equação 2.21. Dividindo-se em endurecimento isotrópico e cinemático, ambos serão abordados nas próximas subseções.

2.6.1 Endurecimento Isotrópico

Este tipo de endurecimento exposto na Equação 2.23 diz respeito à um aumento da tensão de escoamento do material, ou seja, a superfície de plastificação ($f = 0$), assim como apresentado na Figura 2.6, tem um aumento uniforme em todas as direções resultando em um aumento da sua região de elasticidade (região interna da elipse).

$$Y = r - \sigma_y \quad (2.23)$$

onde

Y , é o termo de endurecimento isotrópico;

r , é uma função dependente da deformação plástica acumulada;

σ_y , é a tensão de escoamento.

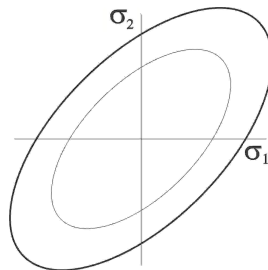


Figura 2.6 Endurecimento Isotrópico

Para este tipo de endurecimento, apresentam-se três casos:

- o endurecimento isotrópico é nulo;
- o endurecimento isotrópico é uma função linear;
- o endurecimento isotrópico é uma função não linear.

Na Figura 2.7 apresenta-se o teste de tensão uniaxial realizado para os três casos citados acima. Sendo 2.7(a), 2.7(b) e 2.7(c) relativos à função de endurecimento nulo, linear e não linear, respectivamente.

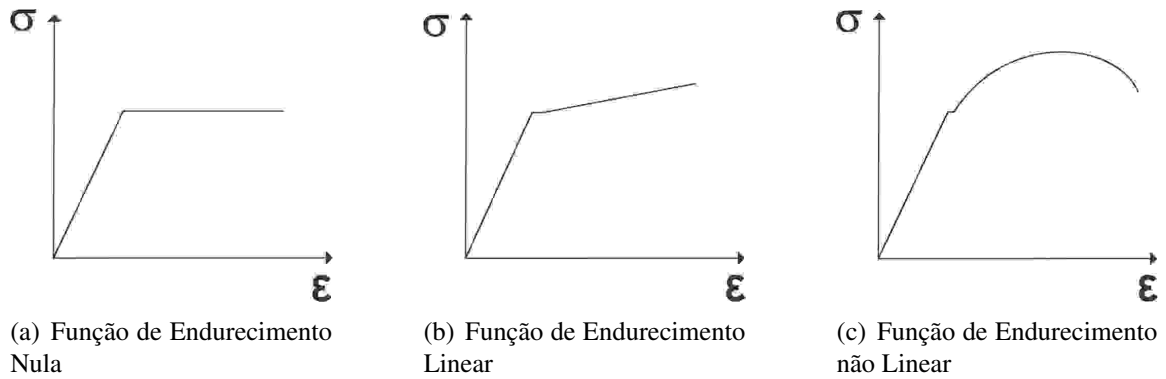


Figura 2.7 Tensão x Deformação

Para o primeiro caso, tem-se a função ($r(p)$) expressa na Equação 2.24 onde não há endurecimento do material, resultando em plasticidade ideal.

$$r(p) = 0 \quad (2.24)$$

onde p , é a deformação plástica acumulada.

Para o segundo caso exposto acima, sabe-se que a função de endurecimento é linear e possui a seguinte característica:

$$r(p) = c \cdot p \quad (2.25)$$

onde c , é uma constante.

Tratando-se agora do último caso, apresenta-se uma função não linear para o endurecimento em função da deformação plástica acumulada [3].

$$r(p) = b \cdot \left(1 - e^{-dp}\right) \quad (2.26)$$

onde b e d , são parâmetros constitutivos.

2.6.2 Endurecimento Cinemático

Neste tipo de endurecimento, a curva de escoamento ($f = 0$) se translada no espaço das tensões, mantém o seu formato, tamanho e a orientação da superfície inicial.

Apresenta-se na Figura 2.8(a) um endurecimento cinemático relativo à direção σ_1 e na Figura 2.8(b) é apresentado endurecimento cinemático relativo à direção σ_2 .

Na Equação 2.27 é apresentado o endurecimento cinemático [22] em função de uma função que sua lei de evolução está apresentada na Equação 2.28.

$$\mathbf{X} = a\beta \quad (2.27)$$

$$\dot{\beta} = \dot{\epsilon}^p - \frac{\varphi}{a}\dot{\alpha}\beta \quad (2.28)$$

onde

$\mathbf{X}$, é o tensor de endurecimento cinemático;
 a e φ , são parâmetros constitutivos;
 β , é uma função.

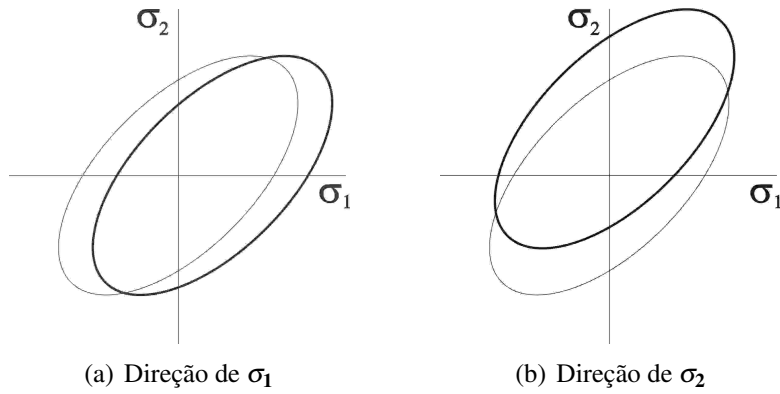


Figura 2.8 Endurecimento Cinemático

Com a definição dos endurecimentos, a Equação 2.8 pode ser reescrita na seguinte forma:

$$f(\mathbf{S}, \mathbf{X}, Y) = |(\mathbf{S}_{dev} - \mathbf{X})| - Y = \sqrt{\frac{3}{2} (\mathbf{S}_{dev} - \mathbf{X}) : (\mathbf{S}_{dev} - \mathbf{X})} - Y \quad (2.29)$$

2.7 Modelo Difusivo

Após ter apresentado o modelo elástico e conceitos necessários para a deformação plástica, esta seção está voltada para o modelo difusivo. O modelo químico utilizado neste trabalho, assim como em [2] altera a matriz de tensões nas componentes hidrostáticas. A deformação química (ϵ^c) está apresentada na expressão abaixo:

$$\epsilon^c = e(c - c_0) \quad (2.30)$$

onde

e , é um parâmetro constitutivo;
 c , é a concentração de soluto para um instante para uma determinada posição;
 c_0 , é a concentração de soluto inicial para uma determinada posição.

A equação da taxa de concentração de soluto [2] na matriz sólida é relacionado com o divergente do fluxo ($\mathbf{J}$) e está apresentada abaixo:

$$\dot{c} = -\nabla \cdot \mathbf{J} \quad (2.31)$$

$$\mathbf{J} = -D \left(\nabla c - \frac{c}{\alpha} \nabla (tr(\mathbf{S})) \right) \quad (2.32)$$

onde $\dot{c}$ é a taxa de concentração de soluto e D é a difusividade.

Na Equação 2.32 verifica-se o acoplamento existente na difusão através da tensão ($\mathbf{S}$) dependente da deformação e explicitado na equação abaixo:

$$\mathbf{S} = \underbrace{\lambda tr(\boldsymbol{\varepsilon}) \mathbf{I} + 2\mu \boldsymbol{\varepsilon}}_{\sigma_{total}} - \underbrace{2e(\lambda + \mu)(c - c_0) \mathbf{I}}_{\sigma_{química}} - \underbrace{2\mu \boldsymbol{\varepsilon}^p}_{\sigma_{plástica}} \quad (2.33)$$

Através da equação acima, identifica-se claramente as parcelas tanto química, como plástica afetando a tensão, ou seja, um acoplamento fortemente ligado entre fenômenos distintos. Além disto nota-se que através da Equação 2.32 o traço da tensão afeta o fluxo relacionado com o fenômeno da difusão. Logo, tem-se o fenômeno da difusão afetando o fenômeno da elasticidade e vice-versa através das Equações 2.33 e 2.32 respectivamente.

Para prosseguir com o desenvolvimento das equações é necessário apresentar o cálculo do traço do tensor de tensões, que é necessário ao cálculo do fluxo de soluto.

Fazendo o cálculo para o traço de $\mathbf{S}$:

$$tr(\mathbf{S}) = 2\lambda tr(\boldsymbol{\varepsilon}) + 2\mu tr(\boldsymbol{\varepsilon}) - 4e(\lambda + \mu)(c - c_0) - 2\mu tr(\boldsymbol{\varepsilon}^p) \quad (2.34)$$

Como o $\boldsymbol{\varepsilon}^p$ é desviador, então: $tr(\boldsymbol{\varepsilon}^p) = 0$. A Equação 2.34 resulta em:

$$tr(\mathbf{S}) = 2\lambda tr(\boldsymbol{\varepsilon}) + 2\mu tr(\boldsymbol{\varepsilon}) - 4e(\lambda + \mu)(c - c_0) \quad (2.35)$$

Logo, o desviador da tensão $\mathbf{S}$:

$$\begin{aligned} \mathbf{S}_{dev} &= \mathbf{S} - \frac{1}{2} tr(\mathbf{S}) \mathbf{I} = \lambda tr(\boldsymbol{\varepsilon}) \mathbf{I} + 2\mu \boldsymbol{\varepsilon} - 2e(\lambda + \mu)(c - c_0) \mathbf{I} - 2\mu \boldsymbol{\varepsilon}^p - \\ &\quad - \frac{1}{2} (2\lambda tr(\boldsymbol{\varepsilon}) + 2\mu tr(\boldsymbol{\varepsilon}) - 4e(\lambda + \mu)(c - c_0) - 2\mu tr(\boldsymbol{\varepsilon}^p)) \mathbf{I} \end{aligned} \quad (2.36)$$

$$\mathbf{S}_{dev} = 2\mu \boldsymbol{\varepsilon} - \mu tr(\boldsymbol{\varepsilon}) \mathbf{I} - 2\mu \boldsymbol{\varepsilon}^p \quad (2.37)$$

Conclui-se através da Equação 2.37 que a parcela relacionada à difusão não entra no cálculo para a função de plastificação exposta na Equação 2.29.

2.8 Equações para o modelo Elastoplástico com Difusão

Nesta seção serão apresentadas as equações envolvidas para o modelo elastoplástico com difusão, iniciando com a equação de equilíbrio que relaciona o tensor de tensões com o vetor força. As equações de estado estão representadas pelas equações de tensão e endurecimentos cinemático e isotrópico. Além da lei da normalidade da plasticidade associativa, das leis de evolução e da conservação da massa do soluto, estão também expostas a função de plastificação, as condições de Kuhn-Tucker e a condição de consistência.

Equação de equilíbrio:

$$\text{div}(\mathbf{S}) = -\mathbf{b} \quad (2.38)$$

Equações de estado:

$$\mathbf{S} = \underbrace{\lambda \text{tr}(\boldsymbol{\varepsilon}) \mathbf{I} + 2\mu \boldsymbol{\varepsilon}}_{\sigma_{total}} - \underbrace{2e(\lambda + \mu)(c - c_0) \mathbf{I}}_{\sigma_{quimica}} - \underbrace{2\mu \boldsymbol{\varepsilon}^p}_{\sigma_{plastica}} \quad (2.39)$$

$$\mathbf{X} = a\boldsymbol{\beta} \quad (2.40)$$

$$Y = r + \sigma_y \quad (2.41)$$

$$r = b \left(1 - e^{-dp} \right) \quad (2.42)$$

Lei da Normalidade:

$$\mathbf{N} = \frac{\partial f}{\partial \mathbf{S}} \quad (2.43)$$

Leis de Evolução:

$$\dot{\boldsymbol{\beta}} = \dot{\boldsymbol{\varepsilon}}^p - \frac{\varphi}{a} \boldsymbol{\beta} \dot{\alpha} \quad (2.44)$$

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\alpha} \mathbf{N} \quad (2.45)$$

Conservação de massa do soluto:

$$\dot{c} = -\nabla \cdot \mathbf{J} \quad (2.46)$$

$$\mathbf{J} = -D \left(\nabla c - \underbrace{\frac{c}{\alpha} \nabla (\text{tr}(\mathbf{S}))}_{\text{acoplamento}} \right) \quad (2.47)$$

Função de Plastificação:

$$f(\mathbf{S}, \mathbf{X}, Y) = \|(\mathbf{S}_{dev} - \mathbf{X})\| - Y = \sqrt{\frac{3}{2} (\mathbf{S}_{dev} - \mathbf{X}) : (\mathbf{S}_{dev} - \mathbf{X})} - Y \quad (2.48)$$

Condições de Kuhn-Tucker:

$$f\dot{\alpha} = 0 \begin{cases} \dot{\alpha} > 0 & \text{e} & f = 0 \\ \dot{\alpha} = 0 & \text{e} & f < 0 \end{cases} \quad (2.49)$$

Condição de Consistência:

$$f(\mathbf{S}_{\text{dev}} + d\mathbf{S}_{\text{dev}}, \mathbf{X} + d\mathbf{X}, Y + dY) = \frac{\partial f}{\partial \mathbf{S}_{\text{dev}}} : d\mathbf{S}_{\text{dev}} + \frac{\partial f}{\partial \mathbf{X}} : d\mathbf{X} + \frac{\partial f}{\partial Y} \cdot dY = 0 \quad (2.50)$$

Além destas equações são necessárias as condições iniciais e de contorno. Já as equações para a formulação numérica do problema se encontram no Apêndice A, juntamente com o algoritmo implementado nesta dissertação.

Com a apresentação do sistema de equações exposto acima, no próximo capítulo será definido como foi implementado este modelo elastoplástico com difusão fazendo a analogia com as camadas do MPhyScas [9].

Implementação utilizando o MPhyScas

Neste capítulo será apresentado o algoritmo utilizado para o desenvolvimento de um problema elastoplástico com difusão e posteriormente será feita a analogia entre o algoritmo e as camadas do MPhyScas.

3.1 Problema Elastoplástico com Difusão

O problema em questão possui três fenômenos, são eles:

- Elástico;
- Plástico;
- Difusivo.

Para a resolução deste problema foi utilizado a técnica chamada de Métodos de Decomposição do Operador (MDO) e Algoritmos de Produtos Associados, que têm demonstrado ser uma alternativa promissora no tratamento de problemas complexos envolvendo equações diferenciais não lineares [1] e [21].

O método possui como característica básica decompor aditivamente o operador global em operadores mais simples e então construir algoritmos sequenciais que permitam tratar cada operador em particular de forma independente. Decompõe-se o problema geral em uma sequência de outros problemas mais simples de forma que possam ser aplicados métodos numéricos clássicos dos quais se conheça bem o comportamento de estabilidade e convergência. Sendo assim o algoritmo global possui uma aproximação do problema geral e segundo alguns autores [10] e [12], possui propriedades de estabilidade e de consistência se cada algoritmo independente também possuir.

Partindo da decomposição do problema geral, que inicialmente é elastoplástico com difusão, após a decomposição obtém-se 3 problemas que serão resolvidos de forma isolada. Primeiramente será resolvido o problema elástico, em seguida o problema plástico. Considerando que o problema elástico e plástico foram resolvidos, parte-se agora para a solução do problema difusivo.

Lembrando que para a solução do problema plástico os valores de entrada já foram calculados pelo problema elástico, isto também é válido para o problema difusivo, ou seja, possui como situação inicial a solução obtida pelo problema elastoplástico. O entendimento deste problema pode ser visualizado com a Figura 3.1 onde observa-se o esquema numérico que é

trata de uma combinação do problema elastoplástico com difusão. Com ele obtem-se inicialmente a solução do fenómeno da elasticidade, com esta solução utiliza-se como estado de entrada (inicial) para o fenómeno plástico, resolve-se o problema plástico e faz a verificação de convergência do problema envolvendo estes dois fenómenos, ou seja, elastoplástico. Havendo a convergência do problema elastoplástico e sua solução é utilizada como estado de entrada (inicial) para o fenómeno difusivo.

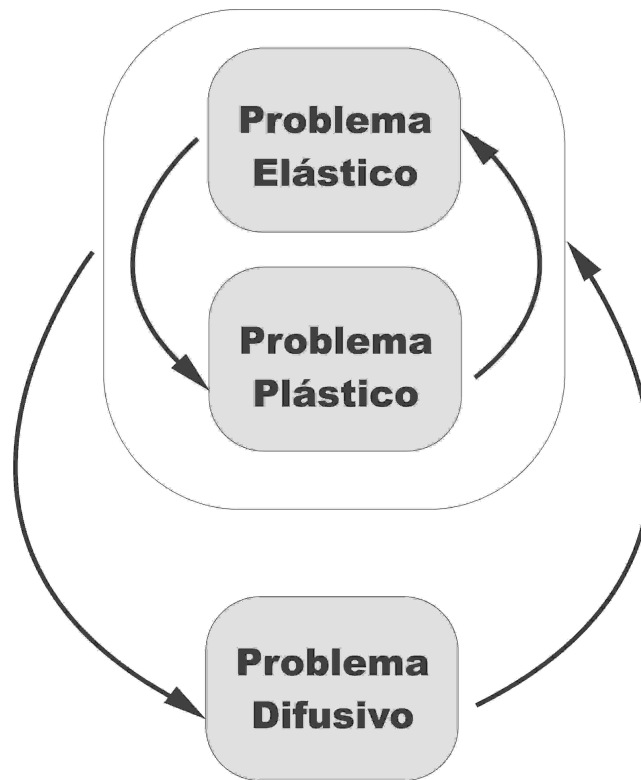


Figura 3.1 Esquema de solução

3.2 Algoritmos

Nesta seção serão apresentados os algoritmos relacionado ao problema elastoplástico com difusão. Inicialmente será exibido o algoritmo no seu mais baixo nível até chegar a níveis mais elevados, passando pelas seções 3.2.1, 3.2.2 e 3.2.3.

3.2.1 Algoritmo de nível I

```
\\ PARA ELASTICIDADE E DIFUSAO  
Inicializar o tempo;  
Calcular tamanho de escalares, vetores e matrizes;  
Alocar escalares, vetores e matrizes;
```

```

Inicializar com zeros todos os escalares, vetores e matrizes;
Calcular a variação do tempo;

\\ PARA DIFUSAO
Montar o vetor de concentração de soluto;
Montar a matriz massa;
Montar o vetor força relacionado com a concentracao;
Acrescentar ao vetor força parcela relacionada ao traco da tensao;
Montar matriz rigidez com valores da difusao;
Acrescentar a matriz rigidez a parcela do traco da concentracao;
Introduzir condicoes de contorno:
    Neumman, Mista;
    Dirichlet, para a taxa da concentracao.
Resolver o sistema para encontrar nova taxa de concentracao;

Atribuir o tempo = variação de tempo;
Enquanto (tempo < intervalo)

    // PARA ELASTICIDADE
    Atualização do vetor deslocamento do elemento;
    Zerar vetor de força auxiliar;
    Deletar e Realocar a matriz rigidez;
    Inicializar a matriz rigidez com zeros;
    Montar a matriz rigidez;
    Somar no vetor força auxiliar a parcela de Feb;
    Introduzir condições de contorno para elasticidade:
        Caso Neumman - vetor força auxiliar;
        Caso Mista - vetor força auxiliar e matriz rigidez.
    Introduzir condições de contorno de Dirichlet;

    // PARA DIFUSAO
    Montar a parcela do vetor força auxiliar para difusao-----
        -----relacionado a taxa de concentracao;
    Acrescentar ao vetor força auxiliar a parcela da concentracao;
    Introduzir condicoes de contorno:
        Neumman, utilizando o vetor força auxiliar;
        Mista, somente afetando o vetor força auxiliar.

Inicializar o erro da iteracao;
Inicializar o contador da iteracao;
Enquanto ((erro iteracao > tolerancia) e (contador < max iteracao))

    // PARA ELASTICIDADE

```

```

Atualizar a forza elastica;
Acrescentar a forza elastica a parcela plástica;
Acrescentar a forza elastica a parcela difusiva;
Introduzir condicoes de contorno Dirichlet apenas afetar-----
                                ----- a forza elastica;
Resolver o sistema para encontrar novo deslocamento;
Calcular o erro da iteracao;
Atualizar o deslocamento;
Calcular a deformacao;
Calcular a tensao;
Calcular o traco da tensao;
Calcular a plasticidade;

// PARA DIFUSAO
Atualizar a forza difusiva;
Deletar e realocar a matriz de difusao auxiliar;
Atribuir a matriz de difusao auxiliar os valores da matriz ----
                                -----todas as parcelas;
Acrescentar a forza difusiva o termo relacionado ao traco;
Acrescentar a matriz de difusao auxiliar termos relacionados --
                                ----- a concentracao;
Introduzir condicoes contorno, mista, apenas afetar a matriz;
Introduzir condicoes de Dirichlet com a forza e matriz auxiliar;
Resolver o sistema para encontrar nova concentracao;
Calcular o erro da iteracao;
Atualizar a concentracao;

Assumir como erro da iteracao o maior entre os dois calculados;
Somar 1 no contador;
Fim do Enquanto
Se (erro da iteracao > tolerancia)
    mensagem de erro e terminar simulacao;
Fim do Se

// PARA ELASTICIDADE
Calcular a solucao exata;
Calcular nova variacao de tempo;
Calcular o erro relacionado com a solucao exata;
Atualizar a tensao;

// PARA DIFUSAO
Calcular a solucao exata;
Calcular a nova variacao de tempo;

```

```

Calcular a taxa de variacao da concentracao;
Calcular o erro da difusao;
Atualizar a concentracao;
Atualizar a taxa de concentracao;

Assume-se a menor variacao de tempo;
Incrementa o tempo;
Fim de Enquanto

```

3.2.2 Algoritmo de nível II

A partir do algoritmo da subseção anterior, pode-se identificar alguns pequenos grupos formados. O algoritmo acima pode ser reescrito também da seguinte maneira:

```

Inicializar o tempo igual a 0;
Elasticidade: execute o seu conjunto de atividades 0;
Difusao: execute o seu conjunto de atividades 0;
Assumir a minima variacao de tempo calculada por Elasticidade e Difusao;
Receber esta minima variacao de tempo;
Difusao: execute o seu conjunto de atividades 1;
Fazer tempo ser igual a variacao de tempo;
Enquanto (tempo < intervalo)
    Elasticidade: execute seu conjunto de atividades 1;
    Difusao: execute seu conjunto de atividades 2;
    Assumir Erro de iteracao igual a 1;
    Inicializar o contador;
    Enquanto ((erro iteracao > tolerancia) e (contador < max iteracao))
        Elasticidade: execute seu conjunto de tarefas 2;
        Elasticidade: execute seu conjunto de tarefas 4;
        Difusao: execute seu conjunto de tarefas 3;
        Erro da iteracao ser maximo valor entre a Elasticidade e Difusao;
        Somar 1 no contador
    Se (erro iteracao > tolerancia)
        Mensagem de erro e terminar a simulacao
    Fim do Se
    Elasticidade: execute seu conjunto de tarefas 3;
    Difusao: execute seu conjunto de tarefas 4;
    Calcular nova variacao minima de tempo entre (Elasticidade e Difusao);
    Assume-se a menor variacao de tempo;
    Incrementa o tempo;
Fim do Enquanto

```

3.2.3 Algoritmo de nível III

Após as subseções anteriores, pode-se ainda expressar o algoritmo em um mais alto nível.

```
Inicializar o tempo igual a 0;  
Bloco: execute o algoritmo 0;  
Receber a variacao de tempo;  
Bloco: execute o seu algoritmo 1;  
Fazer tempo ser igual a variacao de tempo;  
Enquanto (tempo < intervalo)  
    Bloco: execute o seu algoritmo 2;  
    Bloco: execute o seu algoritmo 3;  
    Bloco: execute o seu algoritmo 4;  
    Receber a variacao de tempo;  
    Incrementa o tempo;  
Fim do Enquanto
```

3.3 Implementação no MPhyScas

Com os algoritmos da seção 3.2 faz-se a analogia com o MPhyScas.

A partir do algoritmo exposto na subseção 3.2.1 observa-se a realização de cálculos de matrizes e vetores locais no nível do elemento finito, isto é caracterizado pela camada *Phenomenon* do MPhyScas.

Os conjuntos de cálculos de matrizes e vetores são identificados para 2 problemas, sendo elasticidade e difusão, formado a camada *Group* do MPhyScas. Observando a subseção 3.2.2 verifica-se a existência de algoritmos de solução e cada um deles dita a forma como os sistemas lineares são construídos e resolvidos. Nele também são definidos o tipo de todas as operações envolvendo matrizes, vetores e escalares além do momento em que elas têm de ser executadas, isto é caracterizado pelo *Block* do MPhyScas.

Nota-se que na subseção 3.2.3 são definidos os laços iterativos no tempo, adaptação do modelo e articulação da solução de vários algoritmos do *Block*, sendo assim, caracterizam a camada *Kernel* do MPhyScas.

As camadas do MPhyScas para o problema elastoplástico com difusão estão dispostas abaixo e visualizadas na Figura 3.2

- Um *Kernel*;
- Dois *Blocks*;
- Três *Groups*;
- Três *Phenomenons*.

Com a analogia feita entre o algoritmo global e as camadas do MPhyScas serão definidos os detalhes necessários à cada camada.

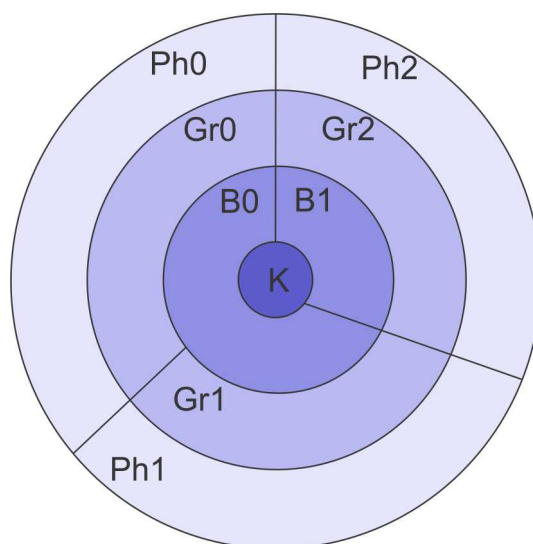


Figura 3.2 Problema Acoplado

K	<i>Kernel</i>
B0	<i>Block da elasticidade e difusão</i>
B1	<i>Block do pós processamento</i>
Gr0	<i>Group da elasticidade</i>
Gr1	<i>Group da difusão</i>
Gr2	<i>Group da gravação</i>
Ph0	<i>Phenomenon da elasticidade</i>
Ph1	<i>Phenomenon da difusão</i>
Ph2	<i>Phenomenon da gravação</i>

Tabela 3.1 Legenda da Figura 3.2

3.3.1 *Kernel*

Como foi dito anteriormente o *Kernel* é o responsável pelos laços iterativos globais. Apresenta somente 1 algoritmo e para o problema em questão pode-se ter 2 *Blocks*, 1 responsável pela execução dos cálculos e o outro para o pós-processamento. Além disto possui também um parâmetro que representa o intervalo. Seu algoritmo está apresentado na subseção 3.2.3.

3.3.2 *Block*

Para o problema elastoplástico com difusão e com os detalhes expostos na subseção 3.3.1, tem-se 2 *Blocks*.

O *Block* do processamento possui 5 algoritmos e o outro *Block* responsável pela gravação apresenta 1 algoritmo.

Para primeiro *Block*, tem-se os seguintes algoritmos com suas respectivas funções:

1. Inicializações e cálculos do tempo;
2. Solução inicial para a difusão;
3. Condições iniciais para a elasticidade e difusão;
4. Solução até a convergência entre dois instantes consecutivos;
5. Solução exata para a elasticidade e difusão, além de cálculo de erros.

Para o segundo *Block*, tem apenas 1 algoritmo e possui a função de gravar.

3.3.3 *Group*

Como foi dito anteriormente e identificado pela Figura 3.2, há 3 *Groups*. Cada *Group* será citado a seguir:

- $\Rightarrow$ *Group* da Elastoplasticidade:

Para este *Group* foram definidos alguns conjuntos de tarefas:

1. Calcular tamanhos, alocar, inicializar e calcular tempo;
2. Estados Iniciais e condições de contorno;
3. Introduzir condições de contorno, resolver o sistema e calcular erro da iteração;
4. Calcular a plasticidade;
5. Calcular a solução exata e erro.

- $\Rightarrow$ *Group* da Difusão:

Para este *Group* foram definidos alguns conjuntos de tarefas:

1. Calcular tempo;
2. Calcular tamanhos, alocar, inicializar, introdução de condições de contorno e resolver sistema;
3. Estados iniciais e introdução de condições de contorno;
4. Introduzir condições de contorno, resolver o sistema e calcular erro da iteração;
5. Calcular a solução exata e erro.

- $\implies$ *Group* da Gravação:

1. Gravar malha, deslocamento, tensão, plasticidade, difusão, erro e tempo.

3.3.4 *Phenomenon*

Como foi dito anteriormente e identificado pela Figura 3.2, há 3 *Phenomenons*. Cada *Phenomenon* será citado a seguir:

- $\implies$ *Phenomenon* da Elastoplasticidade:

1. Calcular tamanhos;
2. Alocar escalares, vetores e matrizes;
3. Inicializar;
4. Calcular os deslocamentos;
5. Calcular a tensão;
6. Calcular o traço da tensão;
7. Calcular a variação do tempo;
8. Atribuir um escalar, vetor ou matriz no seu respectivo correspondente;
9. Deletar e realocar matrizes;
10. Calcular a matriz rigidez;
11. Introduzir condições de contorno;
12. Calcular força plástica;
13. Calcular força difusiva;
14. Calcular a plasticidade;

15. Resolver sistema linear;
16. Calcular erro da iteração;
17. Calcular solução exata.
 - $\implies$ *Phenomenon* da Difusão:
 1. Calcular a variação do tempo;
 2. Calcular tamanhos;
 3. Alocar escalares, vetores e matrizes;
 4. Inicializar;
 5. Calcular a matriz massa;
 6. Calcular parcela da matriz relacionada à difusão;
 7. Calcular parcela da matriz relacionada ao traço da concentração;
 8. Calcular a concentração;
 9. Calcular a taxa de concentração;
10. Atribuir um escalar, vetor ou matriz no seu respectivo correspondente;
11. Deletar e realocar matrizes;
12. Introduzir condições de contorno;
13. Calcular força plástica;
14. Calcular força difusiva;
15. Resolver sistema linear;
16. Calcular erro da iteração;
17. Calcular solução exata.
 - $\implies$ *Phenomenon* da Gravação:
 1. Gravar a malha;
 2. Gravar o deslocamento;
 3. Gravar a tensão;
 4. Gravar a plasticidade acumulada;
 5. Gravar a concentração de soluto;
 6. Gravar o erro;

Construção do Simulador e Simulação

O Simulador e a Simulação são arquivos construídos em *xml* e toda sua construção será apresentada nas seções seguintes. O Simulador e a Simulação para o problema elastoplástico com difusão estão nos Apêndices G e H. Inicialmente será abordado a construção do Simulador e logo em seguida da construção da Simulação. Os arquivos são divididos em partes que estão identificadas na Figura 4.1, onde constata-se a presença do cabeçalho e do corpo. O formato do cabeçalho é o mesmo para os dois arquivos (Simulador e Simulação) e o corpo possui algumas diferenças que serão tratadas nas próximas seções.



Figura 4.1 Representação do Simulador e da Simulação

No cabeçalho deverá ser fornecido todos os dados relativos ao usuário assim como o assunto abordado pelo arquivo do Simulador e/ou da Simulação.

4.1 Construção do Simulador - Corpo

Encontra-se no corpo toda a descrição do Simulador, desde o *Kernel* passando pelo *Block*, *Group* até chegar ao *Phenomenon*. O corpo é ilustrado pela Figura 4.2, que está representando hierarquicamente as camadas do MPhyScas.

Todas as camadas expostas pela Figura 4.2, podem possuir dados expostos e requeridos. Todo dado que estiver exposto por alguma camada, deverá ser requerido por alguma outra, e vice-versa. Por exemplo: o cálculo da variação do tempo, pois o mesmo é calculado pelo *Phenomenon*, se tornando um dado exposto e sendo um dado requerido pela camada do *Group*. Este é apenas um exemplo de como os dados expostos e requeridos podem ser ligados de uma camada com a outra. Podendo ainda citar o cálculo do erro, o tempo, entre outros. Estes dados expostos e requeridos possuem uma definição e um campo que deve ser preenchido, se necessário, em cada camada que está disposta nas próximas subseções.

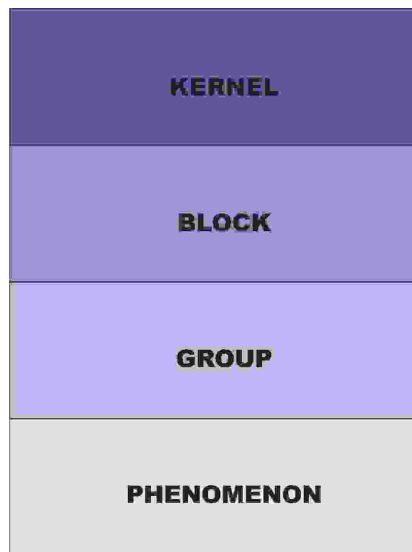


Figura 4.2 Simulador - Construção do Corpo

4.1.1 Construção do Simulador - Corpo - *Kernel*

Esta camada inicia-se com a identificação do *Kernel*, que é representado por uma (*id*), ou seja, uma numeração para identificá-lo.

Os dados expostos e requeridos são aqui definidos pelos dados do sistema do *Kernel*. É necessário inserir a quantidade de dados e diferenciá-los através de um índice e definição do dado. Após a inserção dos dados, apresenta-se a lista de *Blocks* presentes no *Kernel* e são diferenciadas através de sua identificação. A partir de então tem-se o algoritmo do *Kernel*. O algoritmo do *Kernel* possui um código que é lido pelo MPhyScas e executado a partir dos dados que foram fornecidos ao Simulador e à Simulação. Além deste código pode ser fornecido alguns parâmetros. Ainda dentro do algoritmo do *Kernel* são listados os *Blocks* juntamente com seus algoritmos.

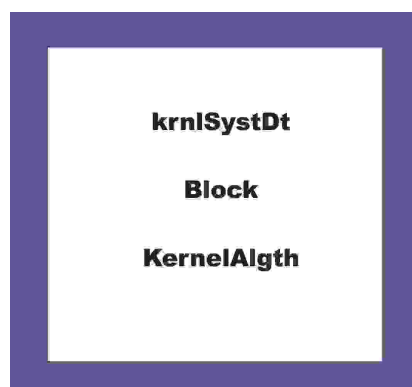


Figura 4.3 Construção do *Kernel*

Conclui-se que a camada Kernel possui as seguintes componentes: dados do sistema (**krnlSystDt**),

a lista de *Block* (**Block**) que o pertencem e o seu algoritmo (**KernelAlgh**). Apresentados na Figura 4.3.

4.1.2 Construção do Simulador - Corpo - *Block*

Esta etapa inicia-se com a identificação do *Block* sendo representado por sua (*id*), ou seja, uma numeração para identificá-lo e diferenciá-lo dos demais, se houver. Os itens necessários para a compreensão dos componentes do *Block* estão demonstrados na Figura 4.4.



Figura 4.4 Construção do Block

Assim como definido na camada anterior, tem-se os dados requeridos e expostos pelo *Block*, sendo necessário inserir a quantidade de dados e diferenciá-los através de um índice e definição do dado.

Constata-se também a presença da lista de *Group* pertencentes ao seu respectivo *Block*. A identificação dos estados globais (**glbSts**) que serão definidos e configurados na camada de cada um dos *Group*. Já na apresentação dos algoritmos do *Block* (**blkAlgh**) além da identificação e do código do algoritmo são apresentadas as tarefas do *Group* e os dados requeridos por cada *Group* pertencente ao algoritmo do *Block* em questão.

4.1.3 Construção do Simulador - Corpo - *Group*

Esta etapa inicia-se com a identificação do *Group* sendo representado por sua (*id*), ou seja, uma numeração para identificá-lo e diferenciá-lo dos demais, se houver. Os itens necessários para a compreensão dos componentes do *Group* estão demonstrados na Figura 4.5

O *gPhen* é o representante do *Group* no nível do *Phenomenon* e é representado pela sua identificação (*id*) e código. Nele são inseridos: dados requeridos originados em outros *Phenomenons*, solucionadores de sistemas e os *Phenomenons* que os pertencem.

Estes dados requeridos são calculados, montados e expostos por outros *Phenomenons*.

Os solucionadores de sistemas ou resolvedores são listados e possuem, além da sua identificação (*id*), um código e definição que são necessários para a utilização do resolvedor correto.

Já no *Phenomenon* são descritas os conjuntos de tarefas para o representante do *Group* no nível do *Phenomenon* (*gPhen*) com o respectivo código de execução e definição. Dentro



Figura 4.5 Construção do *Group*

de cada um dessas tarefas encontram-se as operações que devem ser configuradas e possuem como característica: a identificação, o índice, o código, a definição e se elas estão prontas para serem executadas ou não.

A lista de estados globais e tarefas são representadas pelos códigos e definições. Nos estados globais além do código e definição tem-se: o tipo e o subtipo do estado global, e também a lista de objetos *Phenomenon* que o compartilham. O tipo e o subtipo englobam características relacionadas à matrizes, vetores e escalares, podendo ser densos ou esparsos. Na tarefa é mostrado o *Phenomenon* responsável por executar determinado código que vem seguido pela sua definição.

4.1.4 Construção do Simulador - Corpo - *Phenomenon*

Nesta etapa é necessária a identificação do *Phenomenon* que é representada pela sua identificação, código e definição, respectivamente. Todos os dados listados na Figura 4.6 estão definidos nesta subseção.

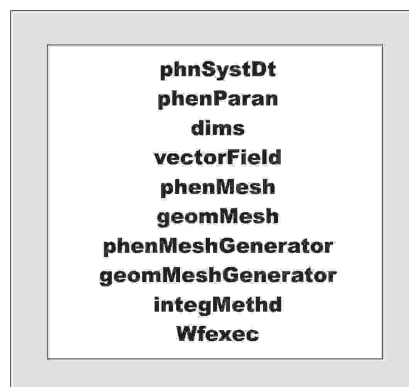


Figura 4.6 Construção do *Phenomenon*

Assim como em todas as camadas anteriores os dados do sistema, como expostos e requeridos, possuem sua apresentação em **phnSystDt**.

Têm-se ainda os parâmetros do *Phenomenon* (**phenParam**), local destinado a parâmetros como módulo de Young, coeficiente de Poisson entre outros. As características referentes a dimensão da malha (**dims**) são divididas em duas: a primeira é relacionada a dimensão do espaço e a segunda está relacionada com a dimensão da estrutura.

Os Campos Vetoriais (**vectorField**) estão definidos através da sua identificação, à qual estado global ele pertence e à qual malha do *Phenomenon* ele está definido.

A malha do *Phenomenon* (**phenMesh**) apresenta a identificação, o código da função de forma, a identificação da malha geométrica e a identificação do gerador da malha do *Phenomenon*.

A malha geométrica (**geomMesh**) apresenta a identificação dela e do gerador da malha geométrica.

O gerador de malha geométrica (**geomMeshGenerator**) necessita de sua identificação, do seu código, da ordem de máxima geometria e do código da função de forma.

O gerador de malha do *Phenomenon* (**phenMeshGenerator**) apresenta sua identificação e um código.

O método de integração (**integMethd**) apresenta a ordem máxima de integração e do código do método de integração que será utilizado.

As tarefas do *Phenomenon* (**WFexec**) são identificadas pelo número do seu código de execução e definição.

Após ser detalhada a construção de todo o Simulador, ou seja, do cabeçalho e corpo, passando pelas camadas do *Kernel*, *Block*, *Group* e *Phenomenon*. Cada constituinte presente em cada uma das camadas pode ser observado através das Figuras 4.3, 4.4, 4.5 e 4.6.

4.2 Construção da Simulação

Como foi dito anteriormente, o arquivo da Simulação apresenta a mesma representação do arquivo do Simulador, ou seja, 4.1. Com relação ao cabeçalho, deve-se ter muita semelhança entre eles, sendo a única diferença no conteúdo abordado. Isto quer dizer alterações apenas da descrição do problema. No corpo da Simulação tem-se além das camadas ditas anteriormente, tem-se também dados relacionados à geometria. A representação da construção do corpo está identificada na Figura 4.7.

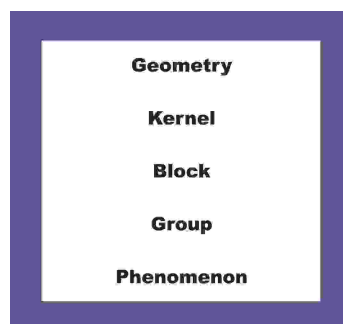


Figura 4.7 Simulação - Construção do Corpo

Inicialmente será analisada a geometria, e em seguida as camadas do *Kernel*, *Block*, *Group* e *Phenomenon*, assim como encontra na Figura 4.7.

4.2.1 Construção da Simulação - Corpo - Geometria

Tratando-se da geometria que é carregada através de um arquivo pela Simulação, têm-se as seguintes necessidades: identificação, dimensão do espaço (**spaceDim**) e da estrutura (**strutDim**).

É na geometria que devem ser definidas as quantidades de volumes, superfícies, curvas e pontos relacionados à geometria (Figura 4.8). Após este conhecimento é necessário entrar em detalhes sobre cada dado definido.

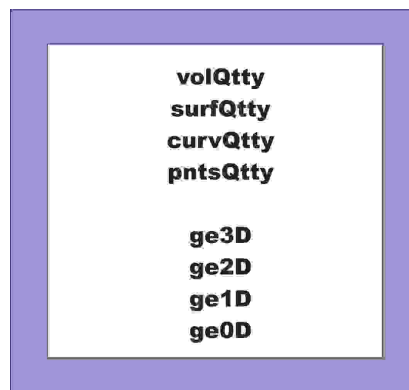


Figura 4.8 Construção da geometria

No arquivo da geometria é necessário inserir as quantidades de volume (**volQty**), superfícies (**surfQty**), curvas (**curvQty**) e pontos (**pntsQty**) que formam a geometria desejada. Para o volume (**ge3D**) é necessário conhecer as superfícies que demarcam o seu contorno. Para a superfície (**ge2D**) deve-se conhecer as curvas que a delimitam. Para a curva (**ge1D**) é necessário identificar os pontos extremos que a definem ou, se for necessário, o centro e raio da curva (para um setor circular). Já para os pontos (**ge0D**) basta identificar as coordenadas que o determinam.

4.2.2 Construção da Simulação - Corpo - *Kernel*

Nesta etapa é necessária sua identificação. No seu interior tem-se a divisão do algoritmo (**KernelAlgh**) do *Kernel* que possui uma identificação e um código predeterminado pelo Simulador.

Todos os parâmetros dos algoritmos, que são definidos para o simulador dado, estão sujeitos a alterações de uma simulação, ou seja, é aqui onde serão inseridos valores para os parâmetros.

4.2.3 Construção da Simulação - Corpo - *Block*

Nesta etapa é necessária sua identificação, isto é, o seu *id*. No seu interior tem-se as divisões dos algoritmos do *Block*(**blkAlgh**) que possuem uma identificação e um código.

Todos os parâmetros dos algoritmos, que são definidos para o simulador dado, estão sujeitos a alterações de uma simulação.

4.2.4 Construção da Simulação - Corpo - *Group*

Nesta etapa é necessária sua identificação, isto é, o seu *id*. No seu interior são definidos os geradores de *Phenomenons* possuindo identificação, código e definição. Além das tarefas com códigos e definições, respectivamente.

4.2.5 Construção da Simulação - Corpo - *Phenomenon*

Nesta etapa é necessária sua identificação, isto é, o seu *id*, juntamente com seu código e definição. No seu interior são definidos os parâmetros do *Phenomenon*, a identificação do grafo da respectiva geometria, com qual ou quais *Phenomenon* aquela geometria será compartilhada. Os dados relacionados para gerar a malha geométrica como valores máximos e mínimos de refinamento, são também fornecidos na Simulação. Assim como a ordem de aproximação do dado para geração da malha do *Phenomenon*. E por fim, com relação as tarefas do *Phenomenon*, são fornecidos os dados para uma determinada simulação.

Malha Geométrica

A malha geométrica apresenta-se de diversas maneiras, podendo possuir uma estrutura tridimensional em um espaço de dimensão três, como possuir uma estrutura unidimensional em um espaço de dimensão um. Propõem-se para este capítulo a ideia de como a malha geométrica é vista pelo MPhyScas, como o entendimento de definições e organização entre as estruturas, além de algumas malhas geométricas são apresentadas.

5.1 Dimensão da Estrutura e Espaço

Esta seção trata de duas definições de suma importância, que são: estrutura e espaço. Tratando-se de malha geométrica, as duas definições estão interligadas, ou seja, elas se complementam.

Primeiramente para se ter uma malha geométrica, deve-se conhecer a dimensão do espaço e da estrutura que se deseja construir. Quando se fala em dimensão da estrutura deve-se ter o prévio conhecimento da dimensão do espaço, pois não se pode construir uma malha de estrutura tridimensional em um espaço de dimensão dois. Conclui-se então que a dimensão do espaço deve ser maior ou igual a dimensão da estrutura. Com base nesta informação têm-se as seguintes possibilidades:

- Espaço de três dimensões:
 - Estrutura tridimensional;
 - Estrutura bidimensional;
 - Estrutura unidimensional;
 - Estrutura de dimensão zero.
- Espaço de duas dimensões:
 - Estrutura bidimensional;
 - Estrutura unidimensional;
 - Estrutura de dimensão zero.
- Espaço de uma dimensão:
 - Estrutura unidimensional;
 - Estrutura de dimensão zero.

- Espaço de dimensão zero:
 - Estrutura de dimensão zero.

Cada um desses itens está explicitado nas subseções seguintes.

5.1.1 Espaço de Dimensão Três

Esta seção é responsável por demonstrar todas as possibilidades existentes para esta configuração. Se tratando de um espaço tridimensional com o sistema de coordenadas definido em (x, y, z) , pode-se formar estruturas de até dimensão três. Na Figura 5.1 observa-se a estrutura e as suas respectivas possibilidades.

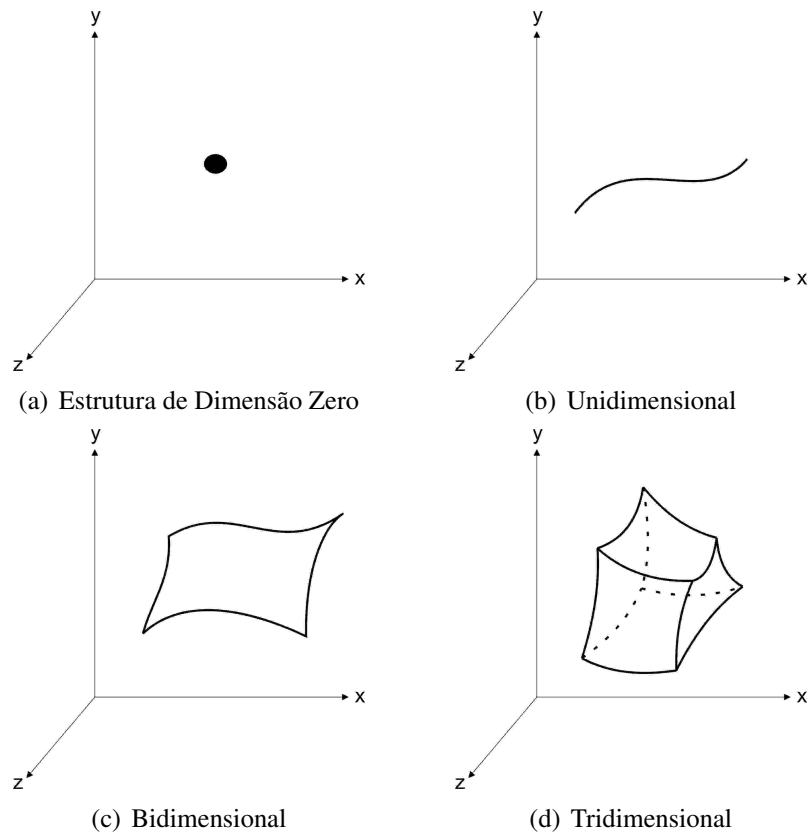


Figura 5.1 Estrutura Tridimensional

5.1.2 Espaço de Dimensão Dois

Sabendo-se o que foi definido para o espaço de dimensão três (seção 5.1.1), assume-se as mesmas definições, só que com uma coordenada a menos, ou seja, o sistema de coordenadas se reduz a (x, y) , podendo-se formar estruturas de dimensão até dois. Na Figura 5.2 observa-se a estrutura bidimensional e suas possibilidades.

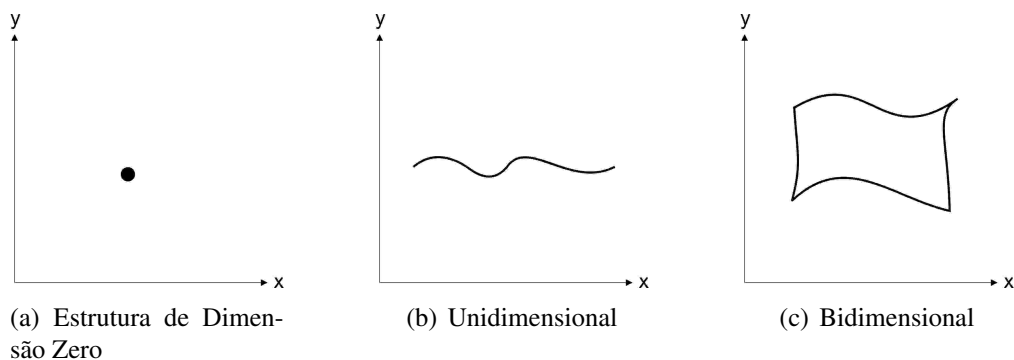


Figura 5.2 Estrutura Bidimensional

5.1.3 Espaço Dimensão Um

Tendo-se apresentado as possíveis configurações para os espaços tridimensional e bidimensional (seções 5.1.1, 5.1.2), agora é feita a apresentação do espaço unidimensional, com o sistema de coordenadas (x), resultando em estruturas de até dimensão um. Na Figura 5.3 observa-se a estrutura unidimensional e suas possibilidades.

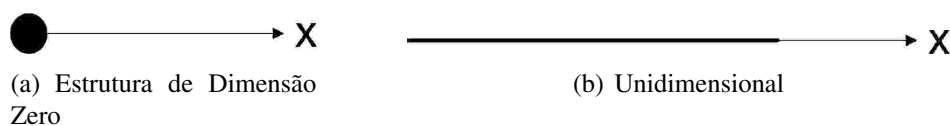


Figura 5.3 Estrutura Unidimensional

5.1.4 Espaço de Dimensão Zero

Para este espaço tem-se a representação feita na Figura 5.4, possuindo apenas coordenada.



Figura 5.4 Estrutura de Dimensão Zero

5.2 Construção da Malha Bidimensional

Uma malha bidimensional apresenta os seguintes entes geométricos:

- $GE0D_2D$, responsável por representar os pontos;
- $GE1D_2D$, responsável por representar as curvas;
- $GE2D_2D$, responsável por representar as superfícies.

A partir deste conhecimento para se construir uma malha bidimensional que possui superfícies $GE2D_2D$, então necessariamente ela será formada por entes $GE1D_2D$ e por consequência de entes $GE0D_2D$. A Figura 5.5 apresenta uma malha bidimensional com as suas componentes demarcadas, sendo cada uma das componentes explicitadas a seguir:

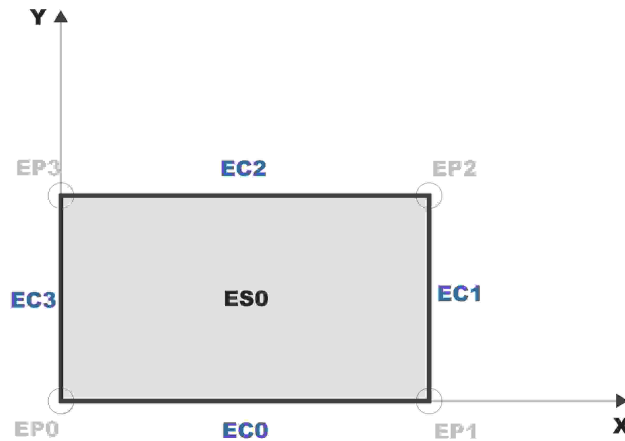


Figura 5.5 Malha Bidimensional

- $EP0, EP1, EP2, EP3$. São os entes $GE0D_2D$, sendo necessárias para sua posição as coordenadas (x,y) ;
- $EC0, EC1, EC2, EC3$. São os entes $GE1D_2D$. Sendo necessário os seus pontos de origem e destino.
- $ES0$. É o ente relacionado à superfície $GE2D_2D$. Nota-se a necessidade dos entes curva para delimitar a superfície formada.

Nota-se que para a formação de uma placa bidimensional foram necessários quatro entes $GE0D_2D$, quatro entes $GE1D_2D$, além de um ente $GE2D_2D$, que estão claramente identificados como um grafo na Figura 5.6.

Analisando-se como um grafo tem-se que o ente da superfície se localiza como raiz do grafo ($ES0$), onde seus filhos são os entes definidos hierarquicamente inferior à sua estrutura, ou seja, neste caso são os entes $GE1D_2D$ (curvas - $EC0, EC1, EC2, EC3$) e por fim seus filhos $GE0D_2D$ (pontos - $EP0, EP1, EP2, EP3$).

O MPhyScas apresenta as seguintes malhas geométricas:

- A malha geométrica de triângulos - *GeomMesh_Tri*;
- A malha geométrica de linhas - *GeomMesh_Line*;
- A malha geométrica de pontos - *GeomMesh_Pointer*.

Cada uma das malha geométricas acima possui funções específicas, sendo a malha de triângulos definida sobre o domínio. A malha linha possui a função dos cálculos necessários

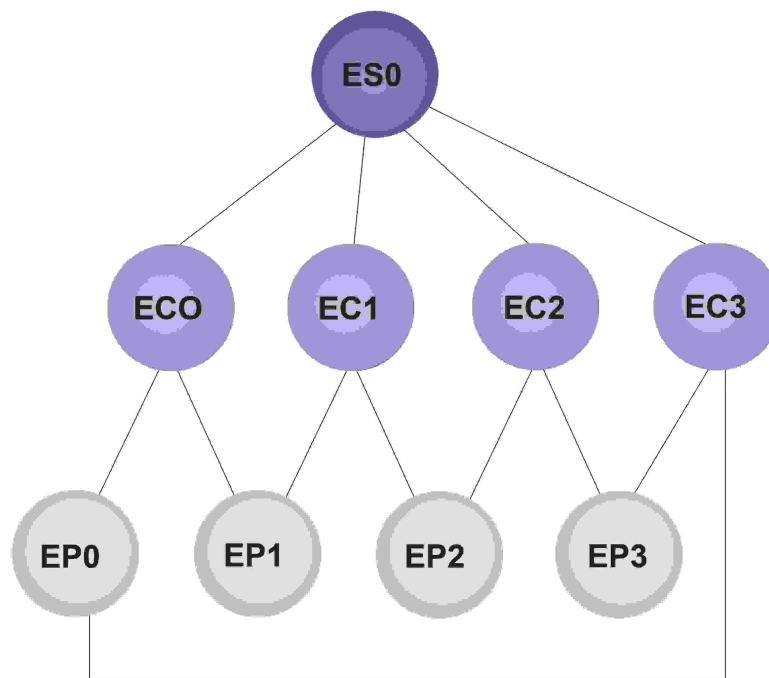


Figura 5.6 Grafo de uma Malha Geométrica Bidimensional

realizados na fronteira do domínio. Já a malha ponto identifica a origem e o destino das malhas linha. Alguns dos principais métodos das malhas geométricas citadas anteriormente estão definidas nas próximas subseções.

5.2.1 Malha de Triângulos (*GeomMesh_Tri*)

A malha de triângulos possui métodos envolvendo vértices, curvas e faces, dentre muitos, alguns serão expostos abaixo:

- inserir vértices → O vértice é formado com o valor de suas coordenadas (x, y) .
- inserir curvas → A curva é inserida através de uma estrutura de pontos já existentes (os dois pontos), ou seja, a origem e o destino.
- inserir faces → A face é inserida através da estrutura de índices das curvas que a formam (as três curvas).
- desenhar malha → A malha pode ser desenhada através das informações fornecidas por uma série de coordenadas.
- remover → Com respeito à remoção de faces é verificada a existência de curvas e pontos que devem ou não ser excluídos. No caso de remoção de uma aresta, necessariamente há a exclusão de uma face e para o vértice é verificada a possibilidade de excluí-lo. No caso de remoção de um ponto, deve-se excluir as faces e arestas que o contém.

- calcular o baricentro da face → A coordenada (x,y) do baricentro é calculada através do fornecimento da identificação da face.
- calcular a área da face → O escalar relativo à área de uma determinada face previamente identificada é calculado.

5.2.2 Malha Linha (*GeomMesh_Line*)

A malha linha possui métodos envolvendo vértices e curvas, dentre muitos, alguns serão expostos abaixo:

- inserir vértices com valores nodais → cada ponto é inserido com suas coordenadas (x,y) .
- inserir arestas com valores dos vértices → as arestas são inseridas através dos índices dos pontos que as formam.
- desenhar → Assim como definido na malha de triângulos, a malha linha é desenhada através do fornecimento de uma série de coordenadas.
- remover → A remoção pode ser feita tanto de arestas como de vértices.
- calcular a normal e tangente ao vértice → Os vetores normal e tangente são determinados por meio de cálculos seguindo a orientação natural, ou seja, com o conhecimento dos vértices de origem e destino.
- calcular a normal e tangente à aresta → Os vetores normal e tangente são determinados por meio de cálculos seguindo a orientação natural, ou seja, com o conhecimento da orientação da curva.

5.2.3 Malha Ponto (*GeomMesh_Pointer*)

A malha ponto possui métodos envolvendo vértices, dentre muitos, alguns serão expostos abaixo:

- inserir valor do vértice → o ponto é inserido através do valor de sua coordenada.
- recuperar vértice → Acessar as coordenadas de um vértice predeterminado.

CAPÍTULO 6

Resultados e Análises

Neste capítulo serão apresentados os resultados obtidos nas simulações para três problemas, sendo eles: elástico, difusivo e elastoplástico difusivo. Além de algumas malhas geométricas geradas.

6.1 Malhas Geométricas

Esta seção está focada na apresentação de algumas malhas geradas utilizando arquivos de geometria diferentes. Primeiramente será mostrada algumas malhas geradas através da geometria especificada na seção 5.2, possuindo: 1 superfície, 4 Curvas, e 4 pontos.

Logo após estes resultados, apresenta-se uma geometria muito semelhante, mas diferenciando desta última apenas pela presença de um furo central. Esta segunda geometria apresenta a seguinte geometria especificada: 1 superfície, 6 curvas e 6 pontos. Esta superfície apresenta 4 curvas externas, que delimitam o domínio e mais duas curvas internas (cada uma representa o semi arco de circunferência). Dos seis pontos, quatro deles são encontrados como a origem e destino das curvas externas e os outros dois são necessários para as duas curvas internas.

6.1.1 Malha da Placa

Alguns refinamentos são apresentados nas Figuras 6.1, 6.2 e 6.3. Na Figura 6.1 foi feito um refinamento de 0.25, ou seja, a maior distância entre os pontos deve ser de 0.25 unidades. Os valores relacionados à quantidade de vértices, arestas e faces encontram-se na Tabela 6.1.

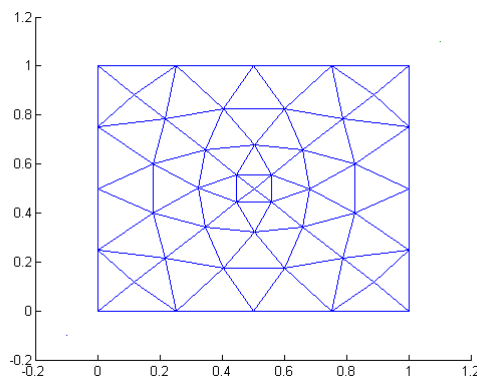


Figura 6.1 Placa com Refinamento de 0.25

Para a Figura 6.2 foi feito um refinamento de 0.125, ou seja, a maior distância existente entre os pontos deve ser de 0.05 unidades. Os valores relacionados à quantidade de vértices, arestas e faces encontram-se na Tabela 6.1.

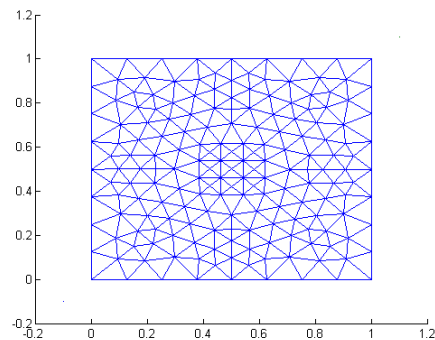


Figura 6.2 Placa com Refinamento de 0.125

Para a Figura 6.3 foi feito um refinamento de 0.05, ou seja, a maior distância existente entre os pontos deve ser de 0.05 unidades. Os valores relacionados à quantidade de vértices, arestas e faces encontram-se na Tabela 6.1.

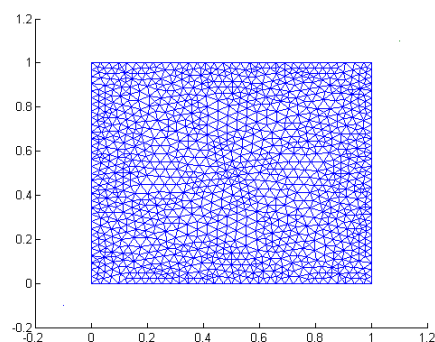


Figura 6.3 Placa com Refinamento de 0.05

Refinamento	0.25	0.125	0.05
Vértices	45	181	1205
Arestas	45	508	3484
Faces	72	328	2280

Tabela 6.1 Valores para a geometria da placa

6.1.2 Malha da Placa com Furo Interno

Para a Figura 6.4 foi feito um refinamento de 0.25, ou seja, a maior distância existente entre os pontos não deve exceder 0.25 unidades. Os valores relacionados à quantidade de vértices, arestas e faces encontram-se na Tabela 6.2.

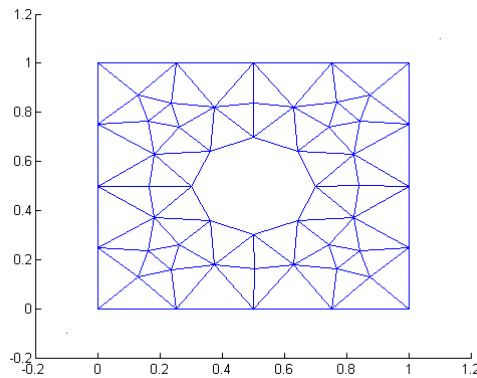


Figura 6.4 Placa com Furo e Refinamento de 0.25

Para a Figura 6.5 foi feito um refinamento de 0.125, ou seja, a maior distância existente entre os pontos não deve exceder 0.125 unidades. Os valores relacionados à quantidade de vértices, arestas e faces encontram-se na Tabela 6.2.

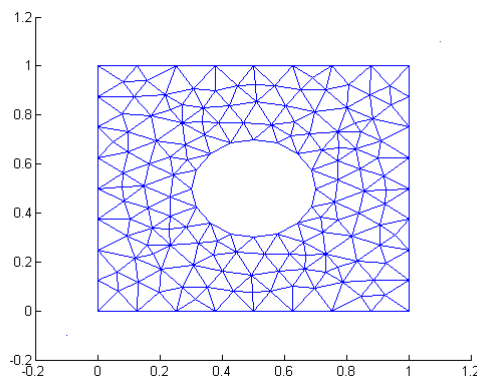


Figura 6.5 Placa com Furo e Refinamento de 0.125

Para a Figura 6.6 foi feito um refinamento de 0.05, ou seja, a maior distância existente entre os pontos não deve exceder 0.05 unidades. Os valores relacionados à quantidade de vértices, arestas e faces encontram-se na Tabela 6.2.

Tomando-se como base os exemplos das seções 6.1.1 e 6.1.2, conclui-se que o gerador de malha apresenta um bom comportamento, pois após os refinamentos a geometria continua sendo respeitada.

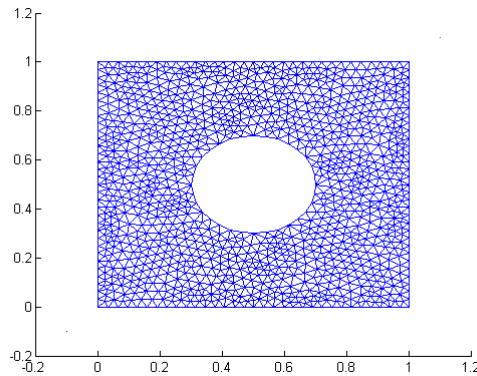


Figura 6.6 Placa com Furo e Refinamento de 0.05

Refinamento	0.25	0.125	0.05
Vértices	52	168	1180
Arestas	132	456	3380
Faces	80	288	2200

Tabela 6.2 Valores para a geometria da placa com furo interno

6.2 Simulador para o Problema Elástico

O Simulador para o problema elástico é constituído pelo *Kernel*, por dois *Blocks*, por dois *Groups* e por dois *Phenomenon*. Esta divisão foi feita justamente para que o primeiro *Block* e suas respectivas camadas hierarquicamente inferiores tratem do processamento, que neste caso são responsáveis pelo cálculo do tamanho dos campos vectoriais, alocação de memória para os mesmos, entre outros. Já o segundo *Block* e suas respectivas camadas hierarquicamente inferiores tratam do pós-processamento, que neste caso são responsáveis pela gravação da malha, do deslocamento, do tempo e do erro calculado.

O problema abordado foi comparado com a sua solução analítica que se encontra no Apêndice B (assumindo o parâmetro ($e = 0$), para desconsiderar a parcela difusiva). Foi construída e calculada uma *WeakForm* específica para o cálculo da solução analítica, pertencente ao *Phenomenon* do processamento, ou seja, a solução exata é calculada pelo *Phenomenon* e sua solução é armazenada e comparada com a solução obtida pelo método do elemento finito. A geometria utilizada para este problema foi uma placa quadrada de dimensões unitárias e condições de contorno de Neumann e Dirichlet. Assim como apresentado na Figura 6.7.

A configuração da malha, assim como a sua geometria são facilmente alteráveis, bastando modificá-los nos dados da Simulação. Os parâmetros inseridos no algoritmo do *Kernel*, nos algoritmos do *Block* e no *Phenomenon* são sujeitos a alterações pela Simulação. Com isto, torna-se bastante simples e viável qualquer estudo de caso. Para o Simulador em questão foram testadas 4 refinamentos para a malha criada pelo *Phenomenon* da elasticidade e comparada com o *Phenomenon* da gravação. Alguns destes parâmetros utilizados no Simulador do

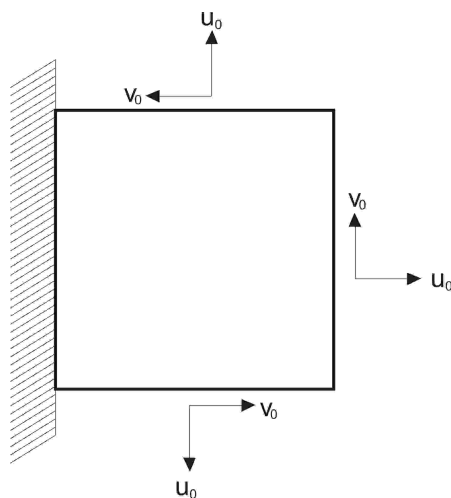


Figura 6.7 Problema para o Simulador Elástico

problema de elasticidade estão citados abaixo:

- intervalo e número de passos no tempo;
- módulo de Young e coeficiente de Poisson.

Para as condições de contorno é necessário atenção, pois são criadas *WeakForm's* iguais, só que com valores e tipos diferentes. Nesta Simulação a *WeakForm* responsável pelas condições de contorno possui os seguintes parâmetros definidos na Tabela 6.3, onde são identificados algumas definições.

<i>geom</i>	identificação da geometria na qual a <i>WeakForm</i> será implementada
<i>type</i>	identificação do tipo
<i>direction</i>	identificação da direção
<i>geomEnt</i>	identificação do ente geométrico
<i>valuesAreGiven?</i>	se os valores serão dados ou não
<i>value0</i>	primeiro valor prescrito
<i>value1</i>	segundo valor prescrito

Tabela 6.3 Parâmetros para Condição de Contorno

O *type* corresponde à numeração de qual código deverá ser fornecido para identificar a condição de contorno existente, podendo ser Dirichlet, Neuman ou Mista. Com relação à *direction* tem-se três possibilidades: normal, tangencial ou ambas. Seguindo a sequência de definições tem-se: *geomEnt*, ou seja é o ente geométrico no qual a *WeakForm* será calculada. E por fim, tratando-se dos valores, eles podem ser fornecidos ou pode-se receber uma função como dado de entrada.

E por fim os parâmetros dos resolvedores também serão exibidos (Tabela 6.4). Inicialmente haverá a apresentação, para posterior explicação do significado de suas definições:

<i>solverId</i>	identificação do resolvidor
<i>type</i>	tipo de resolvidor
<i>method</i>	método utilizado para o resolvidor
<i>preCond</i>	pré condicionador
<i>iter</i>	tolerância
<i>restart</i>	reinício da iteração
<i>maxIter</i>	número máximo de iterações
<i>kFillIn</i>	somente utilizado para <i>preCond</i> →incomplete
<i>threshold</i>	somente utilizado para <i>preCond</i> →incomplete

Tabela 6.4 Parâmetros para os Resolvedores

Considerando as definições da Tabela 6.4, algumas já são autoexplicativas, no caso de *type* refere-se ao nome da biblioteca que está sendo utilizada. O método pode corresponder ao Gradiente Conjugado (CG), Generalização do mínimo resíduo (GMRES), LeastSquare, entre outros, lembrando que estes métodos são implementados pela biblioteca que está declarada no tipo.

As simulações foram criadas com os seguintes dados: ($E = 2.1e^{11} Pa$, $\nu = 0.3$, $u_0 = v_0 = 1.0e^{-5}$). Após as simulações criou-se uma tabela, que apresenta o erro encontrado em cada um dos quatro casos rodados e eles estão apresentados na Tabela 6.5.

Distância entre os pontos	Erro Relativo
0.25	0.0276106
0.125	0.00706793
0.06125	0.00164509
0.03125	0.000411662

Tabela 6.5 Erros para o problema do Simulador Elástico

Através da Tabela 6.5 verifica-se que com um maior número de pontos, ou seja, um maior refinamento da malha obtém-se um erro relativo menor e isto era de se esperar, pois os resultados da solução exata e através do método do elementos finitos, se aproximam a medida que se refina a malha.

6.3 Simulador para o Problema Difusivo

O Simulador responsável pelo problema difusivo, foi o segundo Simulador construído e a partir da experiência adquirida com o Simulador anterior verificou-se a necessidade de fazer algumas alterações com o objetivo de melhorar o desempenho do sistema. Estas mudanças estão relacionadas aos parâmetros de entrada das *WeakForm*'s, pois como foi dito anteriormente, os parâmetros dos *PhenParameter* eram os dados de entrada, sendo assim, muitos parâme-

tros eram lidos e carregados pelas *WeakForm*, com a criação deste novo simulador apenas o *PhenParameter* é lido e carregado.

Os resolvidores utilizados para o problema difusivo estão apresentadas na Tabela 6.6.

<i>solverId</i>	identificação do resolvidor
<i>type</i>	<i>GMM++</i>
<i>method</i>	<i>CG</i>
<i>preCond</i>	<i>Incomplete ldtl</i>
<i>iter</i>	$1.0e^{-8}$
<i>restart</i>	$1.0e^4$
<i>maxIter</i>	$1.0e^5$
<i>kFillIn</i>	$3.0e^0$
<i>threshold</i>	$1.0e^{-3}$

Tabela 6.6 Resolvedor para Problema Difusivo - Resultados

A solução analítica do problema difusivo encontra-se no Apêndice C (fazendo ($e = 0$) para anular a parte elástica do desenvolvimento) e para o Simulador em questão foram testadas 4 refinamentos para a malha criada pelo *Phenomenon* da difusão e compartilhada com o *Phenomenon* da gravação. O problema difusivo está apresentado na Figura 6.8. Este problema apresenta as características apresentadas na Tabela 6.7. Com condições de contorno nas 4 arestas numeradas no problema:

- Aresta 4: condição de Dirichlet;
- Arestas 5, 6 e 7: condição de Neuman.

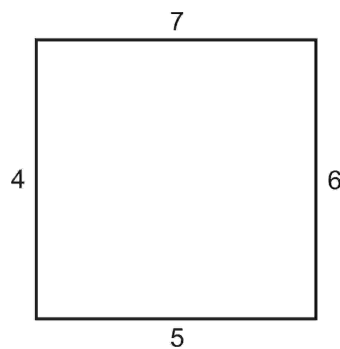


Figura 6.8 Problema para o Simulador Difusivo

Após realizar as simulações, encontra-se na Tabela 6.8 os valores para os erros relativos relacionados à primeira variação de tempo para cada uma das quatro malhas geradas.

$d_0(m^2/s)$	$5.0e^0$
$\rho(kg/m^3)$	$1.0e^0$
$c_p(J/kgK)$	$1.0e^0$
$\alpha(J/m^3)$	$1.94e^{10}$
w_0	$1.0e^{-5}$
γ	$2.0e^0$
c_0	$1.0e^{-6}$

Tabela 6.7 Características para o Problema Difusivo

Distância entre os pontos	Erro Relativo
0.25	0.129299
0.125	0.0567869
0.06125	0.0289165
0.03125	0.0144003

Tabela 6.8 Erros para o problema do Simulador Difusivo

6.4 Problema Elástico e Difusivo

Para o problema elástico difusivo foi rodado um caso, onde sua configuração está apresentada na Figura 6.9.

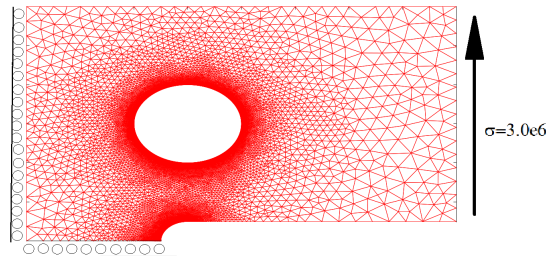


Figura 6.9 Problema Elástico Difusivo - Configuração

Para o caso rodado obteve-se as seguintes Figuras 6.10, 6.11 e 6.12 que correspondem respectivamente à: concentração de soluto, tensão de Von Mises e tensão hidrostática para três instantes de tempo. Pode-se perceber um aumento da concentração de soluto em determinadas regiões, onde as concentrações de tensão são localizadas e uma diminuição da concentração de soluto em outras.

Uma outra visualização importante destes resultados encontra-se na Figura 6.13, onde constata-se a(s) região(ões) que deve(m) ter uma maior atenção e um maior cuidado no trato de casos reais. Os valores máximos e mínimos, para o último passo de tempo, encontrados para este problema encontram-se na Tabela 6.9.

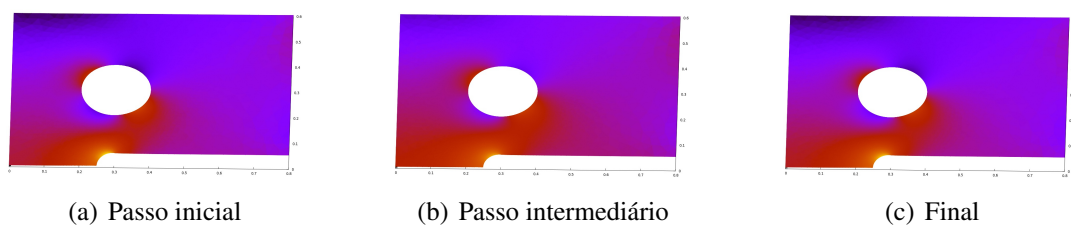


Figura 6.10 Problema Elástico Difusivo - Concentração do soluto

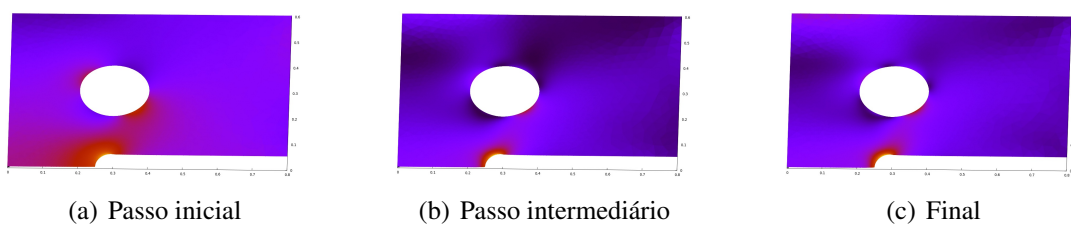


Figura 6.11 Problema Elástico Difusivo - Tensão de Von Mises

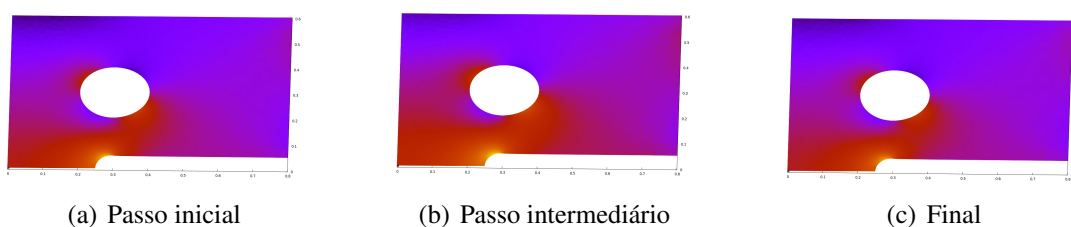


Figura 6.12 Problema Elástico Difusivo - Tensão Hidrostática

	Valor Mínimo	Valor Máximo
Concentração de Solute	$2.099e^{-5}$	$3.015e^{-5}$
Tensão de Von Mises(Pa)	$0.0e^0$	$1.1e^8$
Tensão Hidrostática(Pa)	$-6.0e^7$	$1.2e^8$

Tabela 6.9 Valores Máximos e Mínimos para o problema Elástico Difusivo

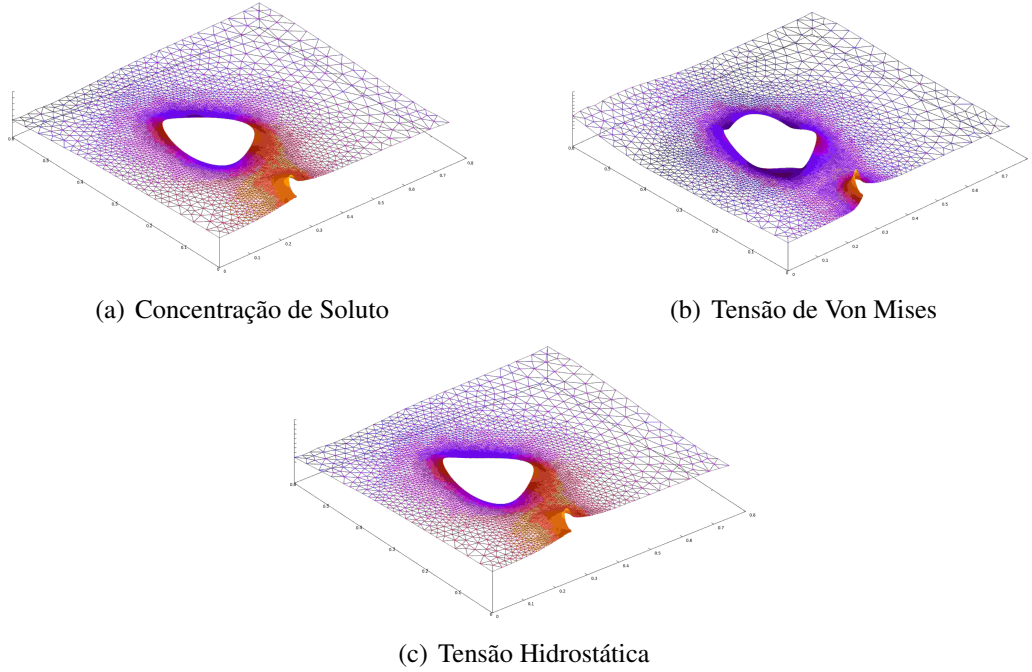


Figura 6.13 Problema Elástico Difusivo - Estado Final

6.5 Simulador para Problema Elastoplástico com Difusão

Após a criação dos Simuladores expostos neste capítulo, torna-se mais fácil a construção do Simulador em questão, sendo necessário a elaboração de *WeakForms* de acoplamento, além de alterações nos níveis hierarquicamente mais elevados que o *Phenomenon*, como por exemplo nos algoritmos do *Block* e *Kernel*. O algoritmo relativo a este Simulador já está exposto e discutido no Capítulo 3.

Para o problema elastoplástico com difusão foi rodado o caso da Figura 6.9. Nesta seção encontra-se a apresentação dos resultados referentes à concentração de soluto, tensão de Von Mises, tensão hidrostática e a deformação plástica acumulada para três instantes de tempo para todo o domínio. Os resultados estão respectivamente nas Figuras 6.14, Figura 6.15, Figura 6.16 e na Figura 6.17.

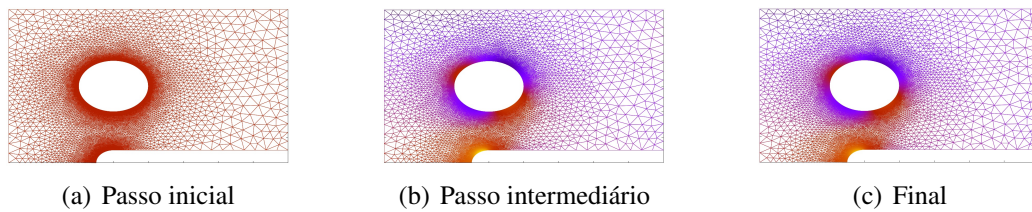


Figura 6.14 Problema Elastoplástico com Difusão - Concentração do soluto

Os valores máximos e mínimos, para o último passo de tempo, encontrados para este problema encontram-se na Tabela 6.10.

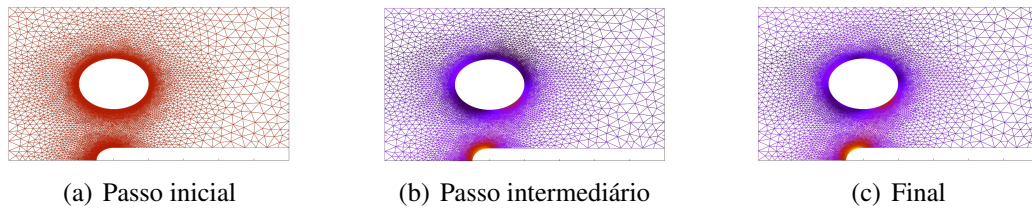


Figura 6.15 Problema Elastoplástico com Difusão - Tensão de Von Mises

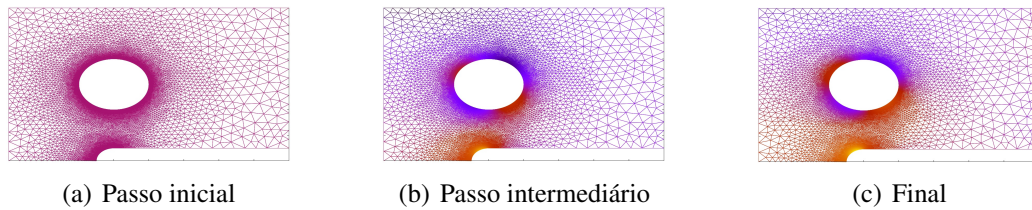


Figura 6.16 Problema Elastoplástico com Difusão - Tensão Hidrostática

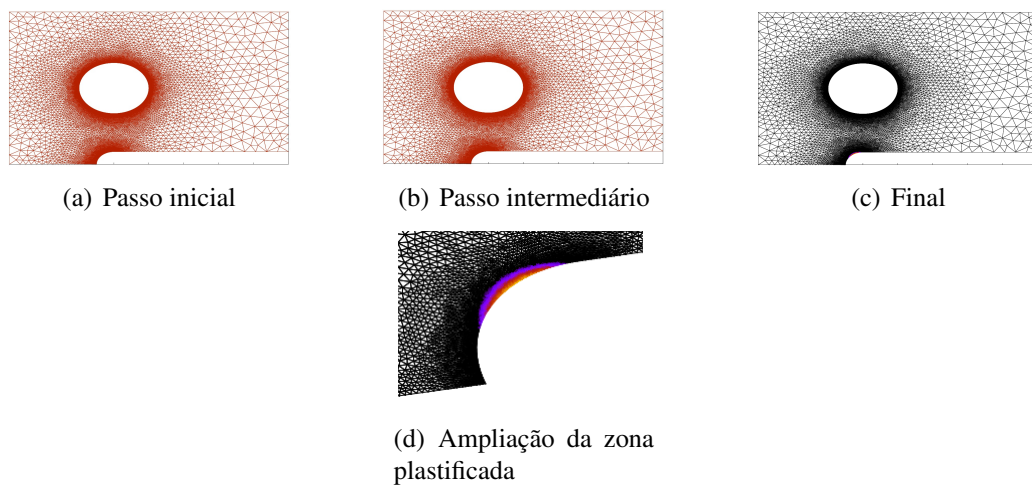


Figura 6.17 Problema Elastoplástico com Difusão - Deformação Plástica Acumulada

	Valor Mínimo	Valor Máximo
Concentração de Solute	$2.99e^{-5}$	$3.015e^{-5}$
Tensão de Von Mises(Pa)	$0.0e^0$	$1.0e^8$
Tensão Hidrostática(Pa)	$-6.0e^7$	$1.0e^8$
Deformação Plástica Acumulada	$0.0e^0$	$8.0e^{-5}$

Tabela 6.10 Valores Máximos e Mínimos para o problema Elastoplástico com Difusão

Conclusão e Trabalhos Futuros

Este capítulo aborda as considerações finais sobre a utilização do ambiente do simulador e da simulação implementados. Apresenta-se também algumas propostas e desafios para trabalhos futuros.

7.1 Conclusões

No que diz respeito à construção de Simuladores utilizando o MPhyScas, constatou-se que para fenômenos acoplados poderem acessar os campos vetoriais de outros fenômenos foi necessário a implementação de um método utilizado em todas as *WeakForms*. Este método só é executado após a construção de todos os fenômenos, pois os campos vetoriais são criados com a criação dos fenômenos. Sem este método seria muito inviável o compartilhamento e acesso de campos vetoriais de outros fenômenos. Lembrando que o acesso só é permitido aos campos vetoriais necessários através de parâmetros lidos no Simulador.

Nesta dissertação foram construídos diversos simuladores e todos eles foram utilizados e tiveram uma parcela para a evolução tanto do MPhyScas, como na apresentação para a construção de outros simuladores.

O último a ser criado e o foco deste trabalho foi o desenvolvimento de um simulador para o problema elastoplástico com difusão e para isto viu-se a necessidade de estudar cada uma destas partes, começando com o problema elástico, passando pelo problema difusivo, problema elástico difusivo e por fim o problema elastoplástico com difusão.

Levando em consideração os fenômenos responsáveis pelos processamentos (exclui-se o fenômeno da gravação) têm-se os dois primeiros simuladores que foram implementados, testados e comparados com soluções analíticas apresentadas nos apêndices e a partir de seus resultados partiu-se para a construção de um simulador mais complexo. Esta experiência necessitou de fenômenos que interagissem mutuamente e para o mesmo foi implementado, testado e comparado com uma solução analítica apresentada no apêndice. Finalizando a construção dos simuladores têm-se o simulador para o problema elastoplástico com difusão. A construção tanto do simulador como da simulação estão apresentadas em apêndices desta dissertação.

As simulações que envolvem os três primeiros simuladores foram realizadas com todos os dados fixos, exceto o número de pontos da malha que corresponde ao tamanho do sistema linear que deverá ser resolvido. Quando o sistema dobra de tamanho têm-se o erro dividido por dois, o que era de se esperar pois o sistema está mais aproximado tratando de elementos finitos.

Para o problema elastoplástico com difusão proposto, notou-se com os resultados obtidos que a concentração do agente químico se deslocou para as posições onde têm-se maiores con-

centrações de tensão. A deformação plástica acumulada encontrada foi muito bem localizada e esperada. Constatou-se que ela vai aumentando com o passar do tempo, o que era de se esperar, pois ela guarda a história do material desde a sua formação.

Em todas as Simulações realizadas e casos analisados tratou-se de utilizar a biblioteca GMM++, pois com ela é possível a criação de vetores e matrizes tanto densos como esparsos, além disto, ela já possui algumas operações e resolvidores de sistemas lineares que foram utilizadas repetidas vezes.

No ambiente do MPhyScas são definidos matrizes e vetores totalmente independentes de qual for a biblioteca de entrada, ou seja, pode-se variar de GMM++ para Seldon, SparseLib, IML++, entre outras, sem nenhuma outra alteração tornando o sistema fácil de testar e usar.

7.2 Trabalhos Futuros

Nesta seção serão abordados alguns trabalhos que estão em andamento assim como sugestões para novos desafios e questionamentos.

- Implementação da paralelização;
- Desenvolvimento de uma interface gráfica para trabalhar com os Simuladores e as Simulações;
- Desenvolvimento de Simuladores mais elaborados, englobando problemas complexos, incluindo alterações em esquemas numéricos;
- Implementar problemas tridimensional;
- Gerar malhas para problemas tridimensionais;
- Fazer a construção do arquivo da geometria através de outros sistemas.

Referências Bibliográficas

- [1] J. M. A. Barbosa. *Estudo da Localização da Deformação em Materiais Elasto-Viscoplásticos Levando-se em Conta Efeitos Térmicos e de Inércia*. Doutorado em engenharia mecânica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1998.
- [2] E. R. R. JR. Brito. *Desenvolvimento de um simulador para problemas Multi-físicas*. Mestrado em engenharia mecânica, 2007.
- [3] J. L. Chaboche and Lemaitre J. *Mechanics of solids materials*. 1990.
- [4] A. Constantinescu, E. Charkaluk, Lederer G., and L. Verger. A computational approach to thermomechanical fatigue. *International Journal of Fatigue*, (26):805–818, 2004.
- [5] F. P. Duda, L. J. Guimarães, A. C. Souza, and J. M. A. Basbosa. On the modeling of deformation-diffusion-damage coupling in elastic solids. *III European Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering*, 2006.
- [6] F. Dune and N. Petrinic. *Introduction to computational plasticity*. Oxford University Press, 2006.
- [7] L. F. Hoffman, L. T. e Perondi. *Estudo de simuladores computacionais aplicados ao ciclo de desenvolvimento de plataformas orbitais*. 2010.
- [8] Koers R. W. J. Bakker A. Krom, A. H. M. Hydrogen transport near a blunting crack tip. *Mechanics and Physics of Solids*, 47:971–992, 1998.
- [9] M. Lencastre. *Conceptualisation of an Environment for the Development of FEM Simulators*. Doutorado em ciências da computação, Universidade Federal de Pernambuco, Recife, Pernambuco, 2004.
- [10] Shaidunov V.V. Marchuk, G.I. Difference methods and their extrapolations. *Application of Mathematics*, 19, 1983.
- [11] P. E. V. Oliveira, S. P. e Miranda. *Concentração crítica de hidrogênio para a fragilização pelo hidrogênio em aços*. 2000.
- [12] Pinsk P. M. Ortiz, M. and R. L Taylor. Operator split methods for numerical solutions of the elastoplastic dynamic problems. *Comp. Meth. Appl. Mech. Engrg.*, 39:137–157, 1983.

- [13] J. N. Reddy and D. K. Gartling. The finite element method in heat transfer and fluid dynamics. 1994.
- [14] C. T. F. Ross. Finite element programs in structural engineering & continuum mechanics. 1996.
- [15] M.H. Saad. Elasticity - theory, applications and numerics. *Elsevier-Academic Press*, 2005.
- [16] F. Santos and M Lencastre. An approach for fem simulator development. *Journal of Computational and Applied Mathematics*, 185(2):326–346, 2006.
- [17] F. Santos, M. Lencastre, and I. Rodrigues. Fem simulator based on skeletons for coupled phenomena. *Proceedings of the 2nd Latin American Conference on Pattern Languages of Programming*, I, 2002.
- [18] F. Santos, M. Lencastre, and M. Vieira. Workflow for simulators based on finite element method. *Proceedings of the International Conference on Computational Science (ICCS)*, 2003.
- [19] F. C. G. Santos, E. R. R. JR. Brito, and J. M. A. Barbosa. Simulação do problema de evolução do dano em uma barra elasto-viscoplástica com acoplamento termomecânico empregando grafo de interface genérica (gig). *7º Congresso Iberoamericano de Engenharia Mecânica*, 2005.
- [20] F. C. G. Santos, E. R. R. JR. Brito, and J. M. A. Barbosa. Dealing with coupled phenomena in the finite element method. *XXVII Latin American Congress on Computational Methods in Engineering*, page 461, 2006.
- [21] J. DC. Simo and C. Mieche. Associative coupled thermoplasticity at finite strains: Formulation, numerical analysis and implementation. *Computer Methods in Applied Mechanics and Engineering*, (98):41–104, 1992.
- [22] J.C. Simo and T.J.R. Hughes. Computational inelasticity. *Springer*, 2000.
- [23] Perić D. Souza Neto, E.A. and D.R.J. Owen. Computational methods for plasticity. *Wiley*, 2008.
- [24] J. Woodtli and R. Kieselbach. Damage due to hydrogen embrittlement and stress corrosion cracking. *Engineering Failure Analysis*, (7):427–450, 2000.
- [25] T. I. Zohdi. Modeling and simulation of a class of coupled thermo-chemo-mechanical processes in multiphase solids. *Computer methods in applied mechanics and engineering*, (193):679–699, 2004.

APÊNDICE A

Formulação Numérica do Problema ElastoPlástico com Difusão

Este apêndice tem o objetivo de demonstrar a formulação numérica para o problema elastoplástico com difusão, juntamente com o seu algoritmo implementado.

- BLK 0 ALGTH 0 - Inicialização

(Group0 - Calcular Tamanhos)
(Group0 - Alocar Escalares, Vetores e Matrizes)
(Group0 - Inicializar Escalares, Vetores e Matrizes)

(Group1 - Calcular Tamanhos)
(Group1 - Alocar Escalares, Vetores e Matrizes)
(Group1 - Inicializar Escalares, Vetores e Matrizes)

- BLK 0 ALGTH 1 - Cálculos Temporais

(Group0 - Calcular Variação de Tempo do Phenomenon)
(Group0 - Receber Variação de Tempo para o Group)

(Group1 - Calcular Variação de Tempo do Phenomenon)
(Group1 - Receber Variação de Tempo para o Group)

$$\Delta t_b = \min(\Delta t_{g0}, \Delta t_{g1})$$

$$\Delta t = \Delta t_b;$$

- BLK 0 ALGTH 2 - Cálculo da Taxa de Concentração

(Group1 - Zerar Matriz e Vetores)
- Zero $M = 0$; - Zero $B = 0$; - Zero $G = 0$;
(Group1 - Calcular Gradiente)
- Calculate $Gr = \nabla U_n$;
- Calculate $tr(\sigma)$;

(Group1 - Cálculo de quantias)

- Calculate $M \quad + = \int_e \rho_i \varphi_i \varphi_J d\Omega;$
- Calculate $B \quad + = \int_e D [\nabla \varphi_J^T \nabla \varphi_i d\Omega];$
- Calculate $B \quad + = \int_e D [\frac{e^2(\lambda+\mu)}{\alpha} \nabla \varphi_J^T \nabla \varphi_i C_n d\Omega];$
- Calculate $\mathbf{B} \quad + = - \int_e \frac{C}{\alpha} D [\nabla \varphi_J^T \cdot \nabla \text{tr} \sigma] \varphi_i d\Omega;$
- Calculate $\mathbf{G} \quad + = \int_e g \varphi_J d\Omega;$

(Group1 - Operação)

- Calculate $G = \text{blas_Mvaxpy}(\text{target}(G), \text{Mat}(B), \text{vec}_1(C_n), \text{vec}_2(G), \alpha(-1), \beta(1));$

(Group1 - Introduções de Contorno)

- Introduce Boundary conditions for $\dot{\mathbf{C}}_n;$ (modify M and G)

(Group1 - Resolvedor da Taxa de Concentração da Difusão)

- Solve $\mathbf{M} \dot{\mathbf{C}}_n = \mathbf{G}$

$t_1 = t_0 = 0.0;$

- BLK 1 ALGTH 0 - Gravação de Dados

(Group2)

- Gravar malha;
- Gravar $C_n;$
- Gravar $U_n;$
- Gravar t_1 and $\Delta t;$

While ($t_1 \leq I$)

$t_1 = t_0 + \Delta t;$

- BLK 0 ALGTH 3 - Atualizar dados para a iteração

(Group0 - Zerar Matriz Rigidez e Vetor Força)

- Zero $K = 0;$
- Zero $F = 0;$

(Group0 - Atribuições)

- Update $U_K = U_n;$
- Update $X_K = X_n;$ (At the integration points)
- Update $\epsilon_K^p = \epsilon_n^p;$ (At the integration points)
- Update $p_K = p_n;$ (At the integration points)

(Group1 - Zerar Matriz e Vetores)

- Zero $M = 0$;
- Zero $B = 0$;
- Zero $G = 0$;

(Group1 - Atualizar instante)

- Update $C_K = C_n$;

- BLK 0 ALGTH 4 - Cálculo Mecânico Difusivo

tol ;

$tol_{\Delta p}$;

$nItersMax$;

$nItersMax_{\Delta p}$;

npt ;

- Initialize $err = 1.0$;

While $((err > tol) \&\& (nIters \leq nItersMax))$

- BLK 0 ALGTH 4.1 - Elasticidade

(Group0 - Cálculo de quantias)

- Calculate $\mathbf{K} \quad + = \int_e \boldsymbol{\varepsilon}^T(\Psi_J) \cdot \mathbf{C} \cdot \boldsymbol{\varepsilon}(\Psi_i) d\Omega$;
- Calculate $\mathbf{F} \quad + = \int_e \Psi_J \cdot \mathbf{b} \cdot d\Omega$;
- Calculate $\mathbf{F} \quad + = \int_e e(3\lambda + 2\mu)(C_K - C_0)(\nabla \Psi_i) d\Omega$;
- Calculate $\mathbf{F} \quad + = \int_e 2\mu \boldsymbol{\varepsilon}_K^p : \nabla \Psi_i d\Omega$; (At the integration points)

(Group0 - Introdução de Condições de Contorno)

- Introduce Boundary conditions (modify K and/or F);

(Group0 - Resolvedor para o problema Elástico)

- Solve $\mathbf{K} \cdot \mathbf{U}_K = \mathbf{F}$

(Group0 - Cálculos precedentes à Plasticidade)

- Calculate $\mathbf{Gr} = \nabla U$;
- Calculate $\boldsymbol{\varepsilon}_K = \frac{1}{2}(\mathbf{Gr} + \mathbf{Gr}^T)$;
- Calculate $te = tr(\boldsymbol{\varepsilon}_K)$;
- Calculate $\boldsymbol{\sigma}_K = \lambda te \mathbf{I} + 2\mu \boldsymbol{\varepsilon}_K$;
- Calculate $\mathbf{S}_q = \boldsymbol{\sigma}_K - e(3\lambda + 2\mu)(C_K - C_0)\mathbf{I}$; (At the integration points)
- Calculate $tr(\boldsymbol{\sigma}) = \boldsymbol{\sigma}_{tr}$;

- BLK 0 ALGTH 4.2 - Plasticidade

(Group0 - $WF_PhenPlast1$)

- Zero $ipVec$
- Loop on all the integration points
 - Calculate $S_K = S_q - 2\mu\epsilon_K^p$;
 - Calculate $ts = tr(S_K)$;
 - Calculate $S_K = S_K - \frac{1}{3}tsI$;
 - Calculate $L = (S_K - X_K)$;
 - Calculate $J = \sqrt{\frac{3}{2}L : L}$;
 - Calculate $z_0 = e^{-bp_K}$;
 - Calculate $r_K = d(1 - z_0)$;
 - Calculate $f = J - \sigma_y - r_K$;
 - If** $f < 0$
 - Update $X_{(K+1)} = X_K$;
 - Update $\epsilon_{(K+1)}^p = \epsilon_K^p$;
 - Update $p_{(K+1)} = p_K$;
 - Else**
 - Calculate $N_K = \frac{L}{J}$;
 - Calculate $\Delta\epsilon = \text{blas_AddMat}(\text{target}(\Delta\epsilon), \text{Mat}_1(\epsilon_K), \text{Mat}_2(\epsilon_n), \alpha(1), \beta(-1))$;
 - Calculate $A1 = (N_K : X_K)$
 - Calculate $tc = tr(\Delta\epsilon)$;
 - Calculate $A2 = tcI - 3\Delta\epsilon$
 - Calculate $A3 = A2 : N_K$;
 - Calculate $\Delta p_{cur} =$

$$= \frac{\mu A3}{\left[-\frac{9}{2}\left(\mu + \frac{1}{2}\right) + \frac{3}{2}\frac{\phi}{a}A1 - dbz_0\right]}$$
 - Calculate $S1' = 2\mu\epsilon_K - \frac{2}{3}teI$;
 - Calculate $DS = -2\mu N_K$
 - $ipVec[indice] = 1$;

End

End of loop

(Group0 - $WF_PhenPlas2$)

- $Virtual_LisPlas$ from $ipVec$;

- Calculate $z_1 = \sigma_y + d$;

While $((Virtual_LisPlas! = \phi) \&\& (nIters_{\Delta p} \leq nItersMax_{\Delta p}))$

For $(it = Virtual_LisPlas.begin(); it! = Virtual_LisPlas.end(); it + +)$ (At the integration points)

- Calculate $z_2 = p_K + \Delta p_{cur}$
- Calculate $YY = z_1 - de^{-bz_2}$

- Calculate $SS = S1' + \Delta p_{cur} DS$
- Calculate $z_3 = 1 + \frac{\phi}{a} \Delta p_{cur}$
- Calculate $XX = X_K + N_K \Delta p_{cur}$
- Calculate $XX = \frac{1}{z_3} XX$
- Calculate $LL = (SS - XX);$
- Calculate $JJ = \sqrt{\frac{3}{2} LL : LL}$
- Calculate $F1 = JJ - YY$
- Calculate $DY = dbe^{-bz_1}$
- Calculate $DX = \frac{\phi}{a} \frac{XX}{z_3} + \frac{N_K}{z_3}$
- Calculate $DX = DS - DX$
- Calculate $DJ = \frac{3}{2} (N_K : DX)$
- Calculate $F2 = DJ - DY$
- Calculate $\Delta p_{new} = \Delta p_{cur} - \frac{F1}{F2}$
- Calculate $err_{\Delta p_{new}} = \frac{|\Delta p_{new} - \Delta p_{cur}|}{|\Delta p_{new}|}$
- If** ($err_{\Delta p_{new}} < tol_{\Delta p}$)
 - $ipVec[indice] = 0;$
- End**
- Update $\Delta p_{cur} = \Delta p_{new};$
- End**
- $nIters_{\Delta p} ++;$
- End**
- If** ($nIters_{\Delta p} > nItersMax_{\Delta p}$)
 - break simulation
- End**

(Group0 - *WF_PhenPlast3*)

Recuperar $ipVec, \Delta p_{cur}, \dots$

$int\ sz = ipVec \rightarrow size()$

For ($int\ i = 0; i < sz; i ++$) (At the integration points)

- Calculate $\Delta \epsilon^p = N_K \Delta p_{cur};$
- Calculate $\epsilon_{(K+1)}^p = \epsilon_n^p + \Delta \epsilon^p;$
- Update $p_{(K+1)} = p_n + \Delta p_{cur};$
- Update $X_{(K+1)} = X_n \left(1 - \frac{\phi}{a} \Delta p_{cur}\right) + \Delta \epsilon^p;$

End

- BLK 0 ALGTH 4.3 - Cálculos de Difusão

(Group1 - Cálculo de quantias)

- Calculate $\mathbf{M} += \int_e \rho \phi_i \phi_j d\Omega;$
- Calculate $\mathbf{B} += \int_e D [\nabla \phi_j^T \nabla \phi_i] d\Omega;$

- Calculate $\mathbf{B} \quad + = \int_e D \left[\frac{e^2(\lambda+\mu)}{\alpha} \nabla \phi_j^T \nabla \phi_i \mathbf{C}_K d\Omega \right];$
- Calculate $\mathbf{B} \quad + = - \int_e \frac{C}{\alpha} D [\nabla \phi_j^T \cdot \nabla \text{tr}(\sigma_K)] \phi_i d\Omega;$
- Calculate $\mathbf{G} \quad + = \int_e g \phi_j d\Omega;$

- BLK 0 ALGTH 4.4 - Calcular C_K

$$par = \frac{2}{\Delta t};$$

(Group1 - Operações)

- Calculate $M = \text{blas_AddMat}(\text{target}(M), \text{Mat}_1(M), \text{Mat}_2(B), \alpha(par), \beta(1));$
- Calculate $C_{aux} = \text{blas_AddVec}(\text{target}(C_{aux}), \text{vec}_1(C_n), \text{vec}_2(\dot{C}_n), \alpha(par), \beta(1));$
- Calculate $G = \text{blas_Mvaxpy}(\text{target}(G), \text{Mat}(M), \text{vec}_1(C_{aux}), \text{vec}_2(G), \alpha(1), \beta(1));$

(Group1 - Introdução de Condições de Contorno)

- Introduce Boundary conditions (modify $\mathbf{M}$ and $\mathbf{G}$);

(Group1 - Resolvedor para o problema Difusivo)

- Solve $\mathbf{M}\mathbf{C}_K = \mathbf{G};$

- BLK 0 ALGTH 4.5 - Cálculo do Erro

(Group1 - Calcular erro da Difusão)

- Calculate $D = C_{K+1} - C_K;$
- Calculate $a = ||D||;$
- Calculate $b = ||C_{K+1}||;$
- err $C_{K+1} = \frac{a}{b};$

(Group0 - Calcular erro da Elasticidade)

- Calculate $D = U_{K+1} - U_K;$
- Calculate $a = ||D||;$
- Calculate $b = ||U_{K+1}||;$
- err $U_{K+1} = \frac{a}{b};$

- err = max { err U_{K+1} , err C_{K+1} };

- BLK 0 ALGTH 4.6 - Atribuições

(Group0 - Elasticidade)

$$U_{K+1} = U_K;$$

(Group1 - Difusão)

$$C_{K+1} = C_K;$$

$nIters++$;

End

If ($nIters < nItersMax$)

- BLK 0 ALGTH 4.7 - Calcular $\dot{C}_n$

(Group1 - Operações)

- Calculate $C_n = \text{blas_AddVec}(\text{target}(C_n), \text{vec}_1(C_{(K+1)}), \text{vec}_2(C_n), \alpha(1), \beta(-1))$;

- Calculate $\dot{C}_n = \text{blas_AddVec}(\text{target}(\dot{C}_n), \text{vec}_1(C_n), \text{vec}_2(\dot{C}_n), \alpha(par), \beta(-1))$;

- BLK 0 ALGTH 4.8 - Atribuições

(Group1 - Difusão)

- $C_n = C_{(K+1)}$;

(Group0 - Elasticidade)

- $U_n = U_{(K+1)}$;

- $X_n = X_{(K+1)}$; (At the integration points)

- $\varepsilon_n^p = \varepsilon_{(K+1)}^p$; (At the integration points)

- $p_n = p_{(K+1)}$; (At the integration points)

Else

Break Simulation

End

- BLK 1 ALGTH 0 - Gravação de Dados

(Group2)

- Gravar malha;

- Gravar C_n ;

- Gravar U_n ;

- Gravar t_1 and Δt ;

- BLK 0 ALGTH 5 - Novos Cálculos Temporais

(Group0 - Calcular nova Variação de Tempo do Phenomenon)

(Group0 - Receber nova Variação de Tempo para o Group)

(Group1 - Calcular nova Variação de Tempo do Phenomenon)

(Group1 - Receber nova Variação de Tempo para o Group)

$$\Delta t_b = \min(New\Delta t_{g0}, New\Delta t_{g1})$$

$$\Delta t = \Delta t_b;$$

$$t_0 = t_1;$$

End

APÊNDICE B

Solução exata para o Problema Elástico

Este apêndice tem o objetivo de apresentar a solução exata para o problema elástico analisado nesta dissertação.

Considerando (x,y) as coordenadas do ponto e assume-se a solução abaixo:

$$\mathbf{U}(x,y) = x * y \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + ew_0 \sin(\gamma t) \begin{bmatrix} A_1 (3x^2y - y^3) + B (x^3 - 3xy^2) \\ -B (3x^2y - y^3) - B_2 (x^3 - 3xy^2) \end{bmatrix} \quad (\text{B.1})$$

onde:

- $\mathbf{U}$, é um vetor solução que está em função das coordenadas (x,y) ;
- A_1 , B e B_2 , são constantes;
- u_0 e v_0 , são as intensidades nas direções x e y respectivamente;
- w_0 e γ , são a amplitude e o período respectivamente;
- t , é o tempo.

A expressão para o vetor força está apresentada abaixo:

$$\mathbf{F} = \lambda + \mu \begin{bmatrix} v_0 \\ u_0 \end{bmatrix} \quad (\text{B.2})$$

onde, λ e μ são parâmetros de Lamé.

As deformações estão apresentadas abaixo em notação Kelvin-Voigt:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} u_0 y \\ v_0 x \\ \frac{1}{2} (u_0 x + v_0 y) \end{bmatrix} + \begin{bmatrix} ew_0 \sin(\gamma t) (6A_1 xy + 3B (x^2 - y^2)) \\ ew_0 \sin(\gamma t) (-3B (x^2 - y^2) - 6B_2 xy) \\ \frac{ew_0 \sin(\gamma t)}{2} (3(A_1 + B_2) (x^2 - y^2) - 12B xy) \end{bmatrix} \quad (\text{B.3})$$

E por último as tensões:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{E}{1-\nu^2} (\varepsilon_{xx} + \mu \varepsilon_{yy}) \\ \frac{E}{1-\nu^2} (\varepsilon_{yy} + \mu \varepsilon_{xx}) \\ 2G \varepsilon_{xy} \end{bmatrix} \quad (\text{B.4})$$

onde:

- E , é o módulo de Young;
- G , é o módulo de elasticidade transversal.

APÊNDICE C

Solução exata para o Problema Difusivo

Este apêndice tem o objetivo de apresentar a solução exata para o problema difusivo analisado nesta dissertação.

Considerando (x,y) as coordenadas do ponto e assume-se a solução abaixo:

$$C(x,y) = w_0xy \sin(\gamma t) \quad (C.1)$$

onde:

C , é a solução que está em função das coordenadas (x,y) ;
 w_0 e γ , são a amplitude e o período respectivamente;
 t , é o tempo.

O gradiente da difusão está apresentado abaixo:

$$\nabla C = \begin{bmatrix} w_0y \sin(\gamma t) \\ w_0x \sin(\gamma t) \end{bmatrix} \quad (C.2)$$

Derivando a Equação C.1 em relação ao tempo, encontra-se:

$$\dot{C} = w_0xy\gamma \cos(\gamma t) \quad (C.3)$$

A força apresenta a seguinte forma:

$$F = \rho c_p w_0 \gamma \cos(\gamma t) xy + \frac{De}{\alpha} w_0 \sin(\gamma t) \frac{E}{1-\nu} (u_0x + v_0y) \quad (C.4)$$

onde:

D , é o parâmetro da difusão;
 e e α , são parâmetros constitutivos;
 E , é o módulo de Young;
 G , é o módulo de elasticidade transversal.

O fluxo fica da seguinte forma:

$$J = \mathbf{n}D \left(\nabla C - \frac{e}{\alpha} C \nabla \text{tr}(\mathbf{S}) \right) \quad (C.5)$$

onde: $\mathbf{S}$, é a tensão decrescida das parcelas plástica e difusiva.

Desenvolvimento da Variação da Plasticidade

Este apêndice tem o objetivo de apresentar o desenvolvimento da variação da plasticidade utilizado nesta dissertação. Lembrando que a plasticidade do material é acumulativa e está expressa abaixo:

$$p = \int \|\dot{\varepsilon}^p\| dt \quad (D.1)$$

onde $\dot{\varepsilon}^p$, é a taxa da deformação plástica.

Apresenta-se abaixo as seguintes tensões:

$$\begin{aligned} \sigma &= 2\mu\varepsilon + \lambda tr(\varepsilon)\mathbf{I} \\ \sigma^c &= 2e(\lambda + \mu)(C - c_0)\mathbf{I} \\ \sigma^p &= 2\mu\varepsilon^p \end{aligned} \quad (D.2)$$

onde:

- μ e λ , são parâmetros de Lamé;
- $\mathbf{I}$, é a matriz identidade;
- c_0 , é a concentração química para o instante inicial;
- C , é a concentração química para um instante qualquer;
- ε , é a deformação total;
- ε^p , é a deformação plástica.

A partir das equações expostas em D.2, apresenta-se a tensão total (σ) decrescida das tensões geradas pelas parcelas química (σ^c) e plástica (σ^p).

$$\mathbf{S} = \sigma - (\sigma^c + \sigma^p) \quad (D.3)$$

Encontrando o tensor desviador ($\mathbf{S}_{dev}$) de D.3, tem-se:

$$\mathbf{S}_{dev} = 2\mu\varepsilon - \mu tr(\varepsilon)\mathbf{I} - 2\mu\varepsilon^p \quad (D.4)$$

A variação do tensor desviador expresso em D.4, resulta em:

$$\Delta\mathbf{S}_{dev} = 2\mu\Delta\varepsilon - \mu tr(\Delta\varepsilon)\mathbf{I} - 2\mu\Delta\varepsilon^p \quad (D.5)$$

A equação do endurecimento isotrópico (Y) possui a seguinte característica:

$$Y = b \left(1 - e^{-dp}\right) + \sigma_y \quad (D.6)$$

onde:

b e d , são parâmetros constitutivos;
 p , é a plasticidade acumulada;
 σ_y , é a tensão de proporcionalidade do material.

O variação do endurecimento isotrópico, está apresentado abaixo:

$$\Delta Y = b d e^{-dp} \cdot \Delta p \quad (D.7)$$

O variação do endurecimento cinemático está expreso abaixo:

$$\Delta \mathbf{X} = (a \mathbf{N} - \varphi \mathbf{X}) \cdot \Delta p \quad (D.8)$$

onde:

$\mathbf{N}$, é a normal;
 a e φ , são parâmetros constitutivos.

A função de plastificação (f), está apresentada abaixo:

$$f = \sqrt{\frac{3}{2} (\mathbf{S}_{dev} - \mathbf{X}) : (\mathbf{S}_{dev} - \mathbf{X})} - Y \quad (D.9)$$

A partir da função de plastificação encontra-se:

$$\frac{\partial f}{\partial \mathbf{S}_{dev}} = \frac{\frac{3}{2} (\mathbf{S}_{dev} - \mathbf{X})}{J} \quad (D.10)$$

$$\frac{\partial f}{\partial \mathbf{X}} = -\frac{\frac{3}{2} (\mathbf{S}_{dev} - \mathbf{X})}{J} \quad (D.11)$$

$$\frac{\partial f}{\partial Y} = -1 \quad (D.12)$$

Tendo as Equações D.5, D.7, D.8 e D.10, D.11 e D.12, deve-se substituí-las em:

$$f(S_{dev} + dS_{dev}, X + dX, Y + dY) = \frac{\partial f}{\partial S_{dev}} \cdot dS_{dev} + \frac{\partial f}{\partial X} \cdot dX + \frac{\partial f}{\partial Y} \cdot dY = 0 \quad (D.13)$$

encontrando:

$$\Delta p = \frac{-3\mu (\mathbf{N} : \Delta \boldsymbol{\varepsilon})}{-\frac{9}{2}(\mu + a) - \frac{3}{2}\varphi (\mathbf{N} : \mathbf{X}) + d b e^{-bp_K}} \quad (D.14)$$

Geometria para uma Placa Bidimensional

Neste apêndice será apresentada a construção do arquivo da geometria para uma placa quadrada de lado unitário. Através deste arquivo, o MPhyScas faz a geração da malha.

```
<GeomSim version="1.0" platform="Linux">
  <Properties name="Geometry for simulation">
    <Responsible name="Felipe A. Cruz" email="email@hotmail.com"/>
    <Description>Simulation geometry</Description>
    <Documentation>doc.pdf</Documentation>
  </Properties>
  <!-- Global Geometry -->
  <volQtty qtty="0"/>
  <surfQtty qtty="1"/>
  <curvQtty qtty="4"/>
  <pntsQtty qtty="4"/>

  <ge2D parameters="no" typ="0" code="2" id="8">
    <!--plane region-->
    <extChldrn>
      <childG id="4"/>
      <childG id="5"/>
      <childG id="6"/>
      <childG id="7"/>
    </extChldrn>
  </ge2D>
  <ge1D parameters="no" typ="1" code="1" id="4">
    <childG id="0"/>
    <childG id="1"/>
  </ge1D>
  <ge1D parameters="no" typ="1" code="1" id="5">
    <childG id="1"/>
    <childG id="2"/>
  </ge1D>
  <ge1D parameters="no" typ="1" code="1" id="6">
    <childG id="2"/>
    <childG id="3"/>
  </ge1D>
```

```

<ge1D parameters="no" typ="1" code="1" id="7">
  <childG id="3"/>
  <childG id="0"/>
</ge1D>

<ge0D parameters="yes" code="0" id="0">
  <geomPar value="0.0e0" def="coord_x"/>
  <geomPar value="0.0e0" def="coord_y"/>
</ge0D>
<ge0D parameters="yes" code="0" id="1">
  <geomPar value="1.0e0" def="coord_x"/>
  <geomPar value="0.0e0" def="coord_y"/>
</ge0D>
<ge0D parameters="yes" code="0" id="2">
  <geomPar value="1.0e0" def="coord_x"/>
  <geomPar value="1.0e0" def="coord_y"/>
</ge0D>
<ge0D parameters="yes" code="0" id="3">
  <geomPar value="0.0e0" def="coord_x"/>
  <geomPar value="1.0e0" def="coord_y"/>
</ge0D>
</GeomSim>

```

APÊNDICE F

Geometria para uma Placa Bidimensional com Furo Interno

Neste apêndice será apresentada a construção do arquivo da geometria para uma placa quadrada de lado unitário com furo interno. Através deste arquivo, o MPhyScas faz a geração da malha.

```
<GeomSim version="1.0" platform="Linux">
  <Properties name="Geometry for simulation">
    <Responsible name="Felipe A. Cruz" email="email@hotmail.com"/>
    <Description>Simulation geometry II</Description>
    <Documentation>doc.pdf</Documentation>
  </Properties>
  <!-- Global Geometry -->
  <volQtty qtty="0"/>
  <surfQtty qtty="1"/>
  <curvQtty qtty="6"/>
  <pntsQtty qtty="6"/>

  <ge2D parameters="no" typ="0" code="2" id="8">
    <!--plane region-->
    <extChldrn>
      <childG id="6"/>
      <childG id="7"/>
      <childG id="9"/>
      <childG id="10"/>
    </extChldrn>
    <intChldrn>
      <childG id="11"/>
      <childG id="12"/>
    </intChldrn>
  </ge2D>
  <gelD parameters="no" typ="1" code="1" id="6">
    <childG id="0"/>
    <childG id="1"/>
  </gelD>
  <gelD parameters="no" typ="1" code="1" id="7">
```

```

        <childG id="1"/>
        <childG id="2"/>
    </ge1D>
    <ge1D parameters="no" typ="1" code="1" id="9">
        <childG id="2"/>
        <childG id="3"/>
    </ge1D>
    <ge1D parameters="no" typ="1" code="1" id="10">
        <childG id="3"/>
        <childG id="0"/>
    </ge1D>
    <ge1D parameters="yes" typ="2" code="1" id="11"> <!-- Circun-->
        <childG id="4"/>
        <childG id="5"/>
        <geomPar value="0.2e0" def="radius"/>
        <geomPar value="0.5e0" def="center_x"/>
        <geomPar value="0.5e0" def="center_y"/>
        <geomPar value="1.0e0" def="centerDir"/>
    </ge1D>
    <ge1D parameters="yes" typ="2" code="1" id="12"> <!-- Circun-->
        <childG id="5"/>
        <childG id="4"/>
        <geomPar value="0.2e0" def="radius"/>
        <geomPar value="0.5e0" def="center_x"/>
        <geomPar value="0.5e0" def="center_y"/>
        <geomPar value="1.0e0" def="centerDir"/>
    </ge1D>
    <ge0D parameters="yes" code="0" id="0">
        <geomPar value="0.0e0" def="coord_x"/>
        <geomPar value="0.0e0" def="coord_y"/>
    </ge0D>
    <ge0D parameters="yes" code="0" id="1">
        <geomPar value="1.0e0" def="coord_x"/>
        <geomPar value="0.0e0" def="coord_y"/>
    </ge0D>
    <ge0D parameters="yes" code="0" id="2">
        <geomPar value="1.0e0" def="coord_x"/>
        <geomPar value="1.0e0" def="coord_y"/>
    </ge0D>
    <ge0D parameters="yes" code="0" id="3">
        <geomPar value="0.0e0" def="coord_x"/>
        <geomPar value="1.0e0" def="coord_y"/>
    </ge0D>

```

```
<ge0D parameters="yes" code="0" id="4">
  <geomPar value="0.3e0" def="coord_x"/>
  <geomPar value="0.5e0" def="coord_y"/>
</ge0D>
<ge0D parameters="yes" code="0" id="5">
  <geomPar value="0.7e0" def="coord_x"/>
  <geomPar value="0.5e0" def="coord_y"/>
</ge0D>
</GeomSim>
```

Desenvolvimento do Simulador Elastoplástico com Difusão

Este apêndice tem o objetivo de apresentar o arquivo utilizado pelo MPhyScas para o problema desenvolvido nesta dissertação.

```
<Simulator version="1.0" platform="Linux">
  <Properties name="Parallel Testing">
    <Responsible name="Felipe A. Cruz" email="email@email.com"/>
    <Description>Test Simulator Elasticity and Diffusion</Description>
    <Documentation>doc.pdf</Documentation>
  </Properties>

  <!-- Kernel description -->
  <Kernel id="0">
    <krnlSystData size="4">
      <!-- data exposed by the Kernel -->
      <exposedDt krnlIndex="0" def="t_1"/>
      <exposedDt krnlIndex="1" def="delta_t"/>
      <exposedDt krnlIndex="2" def="interval"/>
      <!-- data required by the Kernel -->
      <requiredDt krnlIndex="3" def="delta_tb"/>
    </krnlSystData>
    <!-- list of Blocks -->
    <Block id="0"/>
    <Block id="1"/>
    <!-- End list of Block ids -->
    <KernelAlgth id="0" code="6">
      <sysPar def="interval" value="0.0e0"/>
      <!-- local indices for Block exposed data -->
      <localGlobalExposedDt def="t_1" localInd="0"/>
      <localGlobalExposedDt def="delta_t" localInd="1"/>
      <localGlobalExposedDt def="interval" localInd="2"/>
      <localGlobal blkId="0" localId="0">
        <!-- local indices for required data from this Block -->
        <localGlobalRequiredDt def="delta_tb" localInd="0"/>
        <!-- list of local id and respective BlockAlgth id -->
```

```

        <BlkAlgthVec localInd="0" indInBlk="0"/>
        <BlkAlgthVec localInd="1" indInBlk="1"/>
        <BlkAlgthVec localInd="2" indInBlk="2"/>
        <BlkAlgthVec localInd="3" indInBlk="3"/>
        <BlkAlgthVec localInd="4" indInBlk="4"/>
    </localGlobal>
    <localGlobal blkId="1" localId="1">
        <!-- local indices for required data from this Block -->
        <!-- list of BlockAlgth id and respective local id -->
        <BlkAlgthVec localInd="0" indInBlk="0"/> <!-- Record -->
    </localGlobal>
</KernelAlgth>
</Kernel>
<!-- End Kernel description -->

<!-- Block description -->
<Block id="0">
    <blkSystDt size="5">
        <!-- data exposed by this Block -->
        <exposedDt def="delta_tb" blkIndex="0"/>
        <!-- data required by this Block -->
        <requiredDt def="delta_tg0" blkIndex="1"/>
        <requiredDt def="delta_tg1" blkIndex="2"/>
        <requiredDt def="err_K_gph0" blkIndex="3"/>
        <requiredDt def="err_D_gph2" blkIndex="4"/>
    </blkSystDt>
    <!-- List of Groups and respective data -->
    <Group id="0"/>
    <Group id="1"/>
    <!-- list of codes for global states owned by Group objects-->
    <glbSts id="0"/> <!-- Ke -->
    <glbSts id="1"/> <!-- Ue -->
    <glbSts id="2"/> <!-- Uekp1 -->
    <glbSts id="3"/> <!-- Ueex -->
    <glbSts id="4"/> <!-- Fe -->
    <glbSts id="5"/> <!-- FeAux -->
    <glbSts id="6"/> <!-- Xn -->
    <glbSts id="7"/> <!-- Xnp1 -->
    <glbSts id="8"/> <!-- epsP_n -->
    <glbSts id="9"/> <!-- epsP_np1 -->
    <glbSts id="10"/> <!-- Trace -->
    <glbSts id="11"/> <!-- Sigma_n -->
    <glbSts id="12"/> <!-- Sigma_np1 -->

```

```

<glbSts id="13"/> <!-- P_n -->
<glbSts id="14"/> <!-- P_npl -->
<glbSts id="15"/> <!-- M -->
<glbSts id="16"/> <!-- C -->
<glbSts id="17"/> <!-- Kd -->
<glbSts id="18"/> <!-- Fd -->
<glbSts id="19"/> <!-- FdAux -->
<glbSts id="20"/> <!-- Udn -->
<glbSts id="21"/> <!-- Udnpl -->
<glbSts id="22"/> <!-- Udpn -->
<glbSts id="23"/> <!-- Udpnp1 -->
<glbSts id="24"/> <!-- Udex -->
<glbSts id="25"/> <!-- Udk -->

<!-- list of BlockAlgths -->
<blkAlgth id="0" code="28">
  <!-- local indices for Block exposed data -->
  <localGlobalExposedDt def="delta_tb" localInd="0"/>
  <localGlobal grpId="0">
    <!-- local indices for required data from this Group -->
    <localGlobalRequiredDt def="delta_tg0" localInd="0"/>
    <!-- list of local indices for GroupTask objects used-->
    <localGlobalGrpTsk localInd="0" idInGrp="0"/>
  </localGlobal>
  <localGlobal grpId="1">
    <!-- local indices for required data from this Group -->
    <localGlobalRequiredDt def="delta_tg1" localInd="0"/>
    <!-- list of local indices for GroupTask objects used-->
    <localGlobalGrpTsk localInd="0" idInGrp="0"/>
  </localGlobal>
</blkAlgth>
<blkAlgth id="1" code="29">
  <localGlobal grpId="1">
    <!-- list of local indices for GroupTask objects used-->
    <localGlobalGrpTsk localInd="0" idInGrp="1"/>
  </localGlobal>
</blkAlgth>
<blkAlgth id="2" code="30">
  <localGlobal grpId="0">
    <!-- local indices for required data from this Group -->
    <localGlobalGrpTsk localInd="0" idInGrp="1"/>
  </localGlobal>
  <localGlobal grpId="1">

```

```

        <!-- local indices for required data from this Group -->
        <localGlobalGrpTsk localInd="0" idInGrp="2"/>
    </localGlobal>
</blkAlgth>
<blkAlgth id="3" code="31">
    <algthPar def="iterTol" value="0.0e0"/>
    <algthPar def="iterMax" value="0.0e0"/>
    <localGlobal grpId="0">
        <!-- local indices for required data from this Group -->
        <localGlobalRequiredDt def="err_K_gph0" localInd="0"/>
        <localGlobalGrpTsk localInd="0" idInGrp="2"/>
        <localGlobalGrpTsk localInd="1" idInGrp="4"/>
    </localGlobal>
    <localGlobal grpId="1">
        <localGlobalRequiredDt def="err_D_gph2" localInd="0"/>
        <localGlobalGrpTsk localInd="0" idInGrp="3"/>
    </localGlobal>
</blkAlgth>
<blkAlgth id="4" code="32">
    <localGlobalExposedDt def="delta_tb" localInd="0"/>
    <localGlobal grpId="0">
        <localGlobalRequiredDt def="delta_tg0" localInd="0"/>
        <localGlobalGrpTsk localInd="0" idInGrp="3"/>
    </localGlobal>
    <localGlobal grpId="1">
        <localGlobalRequiredDt def="delta_tg1" localInd="0"/>
        <localGlobalGrpTsk localInd="0" idInGrp="4"/>
    </localGlobal>
</blkAlgth>
</Block>
<Block id="1">
    <blkSystDt size="0">
        <!-- data exposed by this Block -->
    </blkSystDt>
    <Group id="2"/>
    <glbSts id="0"/>
    <!-- list of BlockAlgths -->
    <blkAlgth id="0" code="25">    <!-- Record Datas-->
        <localGlobal grpId="2">
            <localGlobalGrpTsk localInd="0" idInGrp="0"/>
        </localGlobal>
    </blkAlgth>
</Block>

```

```

<!-- End Block description -->

<!-- Group description -->
<Group id="0">
  <grpSystDt size="5">
    <!-- data exposed by this Group -->
    <exposedDt def="delta_tg0" grpIndex="0"/>
    <exposedDt def="err_K_gph0" grpIndex="1"/>
    <exposedDt def="err_U" grpIndex="2"/>
    <!-- data required by this Group -->
    <requiredDt def="delta_tp1" grpIndex="3"/>
    <requiredDt def="delta_t" grpIndex="4"/>
  </grpSystDt>
  <gPhen id="0" code="0">
    <!-- required data that does not come from a Phen -->
    <solver id="0" def="ElSolver_U" code="4"/>
    <!-- List of Phenomena -->
    <Phenomenon id="1"/>
    <!-- list of WeakForms for gPhen - They are at of simulator -->
    <WFexec def="ElCompute_SizeAllocAll" execCode="0">
      <WeakForm id="0" def="WF_ElgPhenCpteSizes" code="37" ---
        --- ready="yes" index="0">
        <wfData def="ComputeSizes" flag="0" data="0"/>
      </WeakForm>
      <WeakForm id="1" def="WF_ElgPhenAllocGvf" code="38" ---
        --- ready="yes" index="0">
        <wfData def="allocType" flag="0" data="1"/>
        <wfData def="nonzeroEntries" flag="0" data="3"/>
      </WeakForm>
    </WFexec>
    <WFexec def="ElCompute_delta_tg0" execCode="1">
      <WeakForm id="2" def="WF_gPhenComputeDeltat" code="52"---
        --- ready="yes" index="0">
        <wfData def="delta_tp1_in" flag="1" data="delta_tp1"/>
        <wfData def="delta_tg0_out" flag="1" data="delta_tg0"/>
      </WeakForm>
    </WFexec>
    <WFexec def="ElInitTimeIter" execCode="2">
      <WeakForm id="3" def="WF_ElgPhenAttrib" code="65"---
        --- ready="yes" index="0">
        <wfData def="Uekp1_Id" flag="0" data="2"/>
        <wfData def="Ue_Id" flag="0" data="1"/>
      </WeakForm>

```

```

<WeakForm id="4" def="WF_gPhenSetValue" code="63"---
    --- ready="yes" index="0">
    <wfData def="FeAux_Id" flag="0" data="5"/>
    <wfData def="value" flag="0" data="0.0e0"/>
</WeakForm>
<WeakForm id="5" def="WF_gPhenDelReall" code="67"---
    --- ready="yes" index="0">
    <wfData def="Ke_Id" flag="0" data="0"/>
</WeakForm>
</WFexec>
<WFexec def="ElKeDefinitive" execCode="3">
    <WeakForm id="6" def="WF_gPhenDefinitive" code="64"---
        --- ready="yes" index="0">
        <wfData def="Ke_Id" flag="0" data="0"/>
    </WeakForm>
</WFexec>
<WFexec def="ElAttrib0" execCode="4">
    <WeakForm id="7" def="WF_ElgPhenAttrib0" code="65"---
        --- ready="yes" index="0">
        <wfData def="Fe_Id" flag="0" data="4"/>
        <wfData def="FeAux_Id" flag="0" data="5"/>
    </WeakForm>
</WFexec>
<WFexec def="ElSolveUErrAttrib" execCode="5">
    <WeakForm id="8" def="WF_gPhenSolver" code="73"---
        --- ready="yes" index="0">
        <wfData def="Ke_Id" flag="0" data="0"/>
        <wfData def="Uekpl_Id" flag="0" data="2"/>
        <wfData def="Fe" flag="0" data="4"/>
    </WeakForm>
    <WeakForm id="9" def="WF_ElgPhenCompute_errU" code="40"---
        --- ready="yes" index="0">
        <wfData def="Ue_Id" flag="0" data="1"/>
        <wfData def="Uekpl_Id" flag="0" data="2"/>
        <wfData def="err_K_out" flag="1" data="err_K_gph0"/>
    </WeakForm>
    <WeakForm id="10" def="WF_ElgPhenAttrib1" code="65"---
        --- ready="yes" index="0">
        <wfData def="Ue_Id" flag="0" data="1"/>
        <wfData def="Uekpl_Id" flag="0" data="2"/>
    </WeakForm>
</WFexec>
<WFexec def="ElAttribErrDt" execCode="6">

```

```

    <WeakForm id="11" def="WF_ElgPhenCompute_errEx" code="40"---
        --- ready="yes" index="0">
        <wfData def="Ue_Id" flag="0" data="1"/>
        <wfData def="Ueex" flag="0" data="3"/>
        <wfData def="err_E_out" flag="1" data="err_U"/>
    </WeakForm>
    <WeakForm id="12" def="WF_ElgPhenAttrib3" code="65"---
        --- ready="yes" index="0">
        <wfData def="Sigma_n_Id" flag="0" data="11"/>
        <wfData def="Sigma_npl_Id" flag="0" data="12"/>
    </WeakForm>
    <WeakForm id="13" def="WF_gPhenComputeDeltat2" code="52"---
        --- ready="yes" index="0">
        <wfData def="delta_tpl_in" flag="1" data="delta_tpl"/>
        <wfData def="delta_tg0_out" flag="1" data="delta_tg0"/>
    </WeakForm>
</WFexec>
</gPhen>

<!-- list of codes for global states owned by this Group -->
<glbSts id="0" def="Ke">
    <glbData subType="1" type="2"/>
    <!-- list of Phn objects sharing this global state -->
    <owner vectorFieldId="4" phnId="1">
        <vectorFields vecfieldId="0"/>
    </owner>
    <coupled vectorFieldId="2" phnId="1"/>
</glbSts>
<glbSts id="1" def="Ue">
    <glbData subType="0" type="1"/>
    <!-- list of Phn objects sharing this global state -->
    <owner vectorFieldId="1" phnId="1"/>
</glbSts>
<glbSts id="2" def="Uekp1">
    <glbData subType="0" type="1"/>
    <!-- list of Phn objects sharing this global state -->
    <owner vectorFieldId="2" phnId="1"/>
</glbSts>
<glbSts id="3" def="Ueex">
    <glbData subType="0" type="1"/>
    <!-- list of Phn objects sharing this global state -->
    <owner vectorFieldId="3" phnId="1"/>
</glbSts>

```

```

<glbSts id="4" def="Fe">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="4" phnId="1"/>
</glbSts>
<glbSts id="5" def="FeAux">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="5" phnId="1"/>
</glbSts>
<glbSts id="6" def="X_n">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="6" phnId="1"/>
</glbSts>
<glbSts id="7" def="X_np1">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="7" phnId="1"/>
</glbSts>
<glbSts id="8" def="EpsP_n">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="8" phnId="1"/>
</glbSts>
<glbSts id="9" def="EpsP_np1">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="9" phnId="1"/>
</glbSts>
<glbSts id="10" def="Trace">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="10" phnId="1"/>
</glbSts>
<glbSts id="11" def="Sigma_n">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->
  <owner vectorFieldId="11" phnId="1"/>
</glbSts>
<glbSts id="12" def="Sigma_np1">
  <glbData subType="0" type="1"/>
  <!-- list of Phn objects sharing this global state -->

```

```

    <owner vectorFieldId="12" phnId="1"/>
</glbSts>
<glbSts id="13" def="P_n">
    <glbData subType="0" type="1"/>
    <!-- list of Phn objects sharing this global state -->
    <owner vectorFieldId="13" phnId="1"/>
</glbSts>
<glbSts id="14" def="P_np1">
    <glbData subType="0" type="1"/>
    <!-- list of Phn objects sharing this global state -->
    <owner vectorFieldId="14" phnId="1"/>
</glbSts>
<GroupTask id="0" def="Grp0_gtsk0">
    <grpTskDt def="ElCompute_SizeAllocAll" execCode="0" phnId="0"/>
    <grpTskDt def="ElInit" execCode="0" phnId="1"/>
    <grpTskDt def="ElCompute_delta_tg0" execCode="1" phnId="0"/>
</GroupTask>
<GroupTask id="1" def="Grp0_gtsk1">
    <grpTskDt def="ElInitTimeIter" execCode="2" phnId="0"/>
    <grpTskDt def="ElKeFeBC" execCode="1" phnId="1"/>
    <grpTskDt def="ElKeDefinitive" execCode="3" phnId="0"/>
</GroupTask>
<GroupTask id="2" def="Grp0_gtsk2">
    <grpTskDt def="ElAttrib0" execCode="4" phnId="0"/>
    <grpTskDt def="ElFeBC" execCode="2" phnId="1"/>
    <grpTskDt def="ElSolveUErrAttrib" execCode="5" phnId="0"/>
    <grpTskDt def="ElStrainStressTr" execCode="3" phnId="1"/>
</GroupTask>
<GroupTask id="3" def="Grp0_gtsk3">
    <grpTskDt def="ElUeexDt" execCode="4" phnId="1"/>
    <grpTskDt def="ElAttribErrDt" execCode="6" phnId="0"/>
</GroupTask>
<GroupTask id="4" def="Grp0_gtsk4">
    <grpTskDt def="ElPlast" execCode="5" phnId="1"/>
</GroupTask>
</Group>
<!-- Group description -->
<Group id="1">
    <grpSystDt size="4">
        <!-- data exposed by this Group -->
        <exposedDt def="delta_tg1" grpIndex="0"/>
        <exposedDt def="err_D_gph2" grpIndex="1"/>
        <!-- data required by this Group -->

```

```

    <requiredDt def="delta_tp3" grpIndex="2"/>
    <requiredDt def="delta_t" grpIndex="3"/>
</grpSystDt>
<gPhen id="2" code="0">
  <!-- required data that does not come from a Phen -->
  <solver id="0" def="DiffSolver_U" code="4"/>
  <!-- List of Phenomena -->
  <Phenomenon id="3"/>
  <WFexec def="DCompute_delta_tg1" execCode="0">
    <WeakForm id="0" def="WF_gPhenComputeDeltat0" code="52"---
      --- ready="yes" index="0">
      <wfData def="delta_tp3_in" flag="1" data="delta_tp3"/>
      <wfData def="delta_tg1_out" flag="1" data="delta_tg1"/>
    </WeakForm>
  </WFexec>
  <WFexec def="DCompute_SizeAllocAll" execCode="1">
    <WeakForm id="1" def="WF_DgPhenCpteSizes" code="37"---
      --- ready="yes" index="0">
      <wfData def="ComputeSizes" flag="0" data="0"/>
    </WeakForm>
    <WeakForm id="2" def="WF_DgPhenAllocGvf" code="38"---
      --- ready="yes" index="0">
      <wfData def="allocType" flag="0" data="1"/>
      <wfData def="nonzeroEntries" flag="0" data="3"/>
    </WeakForm>
  </WFexec>
  <WFexec def="DCompute_FdMKd" execCode="2">
    <WeakForm id="3" def="WF_gPhenAplusB" code="72"---
      --- ready="yes" index="0">
      <wfData def="Kd_Id" flag="0" data="17"/>
      <wfData def="Kd_Id" flag="0" data="17"/>
      <wfData def="C_Id" flag="0" data="16"/>
    </WeakForm>
    <WeakForm id="4" def="WF_DgPhenDefinitive_" code="64"---
      --- ready="yes" index="0">
      <wfData def="Kd_Id" flag="0" data="17"/>
    </WeakForm>
    <WeakForm id="5" def="WF_gPhenMatVecProd" code="66"---
      --- ready="yes" index="0">
      <wfData def="Udk_Id" flag="0" data="25"/>
      <wfData def="Kd_Id" flag="0" data="17"/>
      <wfData def="Udn_Id" flag="0" data="20"/>
    </WeakForm>
  </WFexec>

```

```

<WeakForm id="6" def="WF_gPhenVecSubtract" code="68"---
    --- ready="yes" index="0">
    <wfData def="Fd_Id" flag="0" data="18"/>
    <wfData def="Fd_Id" flag="0" data="18"/>
    <wfData def="Udk_Id" flag="0" data="25"/>
</WeakForm>
<WeakForm id="7" def="WF_gPhenDelReall0" code="67"---
    --- ready="yes" index="0">
    <wfData def="Kd_Id" flag="0" data="17"/>
</WeakForm>
<WeakForm id="8" def="WF_gPhenAttrib" code="65"---
    --- ready="yes" index="0">
    <wfData def="Kd_Id" flag="0" data="17"/>
    <wfData def="M_Id" flag="0" data="15"/>
</WeakForm>
</WFexec>
<WFexec def="DCompute_KdCDefinitSolver" execCode="3">
    <WeakForm id="9" def="WF_DgPhenDefinitive0" code="64"---
        --- ready="yes" index="0">
        <wfData def="M_Id" flag="0" data="15"/>
    </WeakForm>
    <WeakForm id="10" def="WF_gPhenSolverUdpn" code="73"---
        --- ready="yes" index="0">
        <wfData def="M_Id" flag="0" data="15"/>
        <wfData def="Udpn" flag="0" data="22"/>
        <wfData def="Fd" flag="0" data="18"/>
    </WeakForm>
    <WeakForm id="11" def="WF_gPhenDelReall1" code="67"---
        --- ready="yes" index="0">
        <wfData def="M_Id" flag="0" data="15"/>
    </WeakForm>
    <WeakForm id="12" def="WF_gPhenAttrib1" code="65"---
        --- ready="yes" index="0">
        <wfData def="M_Id" flag="0" data="15"/>
        <wfData def="Kd_Id" flag="0" data="17"/>
    </WeakForm>
    <WeakForm id="13" def="WF_DgPhenCFinalForm" code="69"---
        --- ready="yes" index="0">
        <wfData def="C_Id" flag="0" data="16"/>
        <wfData def="Kd_Id" flag="0" data="17"/>
        <wfData def="delta_t" flag="1" data="delta_t"/>
    </WeakForm>
    <WeakForm id="14" def="WF_DgPhenDefinitive1" code="64"---

```

```

        --- ready="yes" index="0">
        <wfData def="C_Id" flag="0" data="16"/>
    </WeakForm>
    <WeakForm id="15" def="WF_DgPhenDefinitive2" code="64"---
        --- ready="yes" index="0">
        <wfData def="M_Id" flag="0" data="15"/>
    </WeakForm>
</WFexec>
<WFexec def="DCompute_FdAuxInitIter" execCode="4">
    <WeakForm id="16" def="WF_gPhenSetValue" code="63"---
        --- ready="yes" index="0">
        <wfData def="FdAux_Id" flag="0" data="19"/>
        <wfData def="value" flag="0" data="0.0e0"/>
    </WeakForm>
    <WeakForm id="17" def="WF_DgPhenMUp" code="70"---
        --- ready="yes" index="0">
        <wfData def="FdAux_Id" flag="0" data="19"/>
        <wfData def="M_Id" flag="0" data="15"/>
        <wfData def="Udk_Id" flag="0" data="25"/>
        <wfData def="Udn_Id" flag="0" data="20"/>
        <wfData def="Udpn_Id" flag="0" data="22"/>
        <wfData def="delta_t" flag="1" data="delta_t"/>
    </WeakForm>
    <WeakForm id="18" def="WF_gPhenAttrib0" code="65"---
        --- ready="yes" index="0">
        <wfData def="Udk_Id" flag="0" data="25"/>
        <wfData def="Udn_Id" flag="0" data="20"/>
    </WeakForm>
</WFexec>
<WFexec def="DCompute_InitFdKd" execCode="5">
    <WeakForm id="19" def="WF_gPhenAttrib1" code="65"---
        --- ready="yes" index="0">
        <wfData def="Fd_Id" flag="0" data="18"/>
        <wfData def="FdAux_Id" flag="0" data="19"/>
    </WeakForm>
    <WeakForm id="20" def="WF_gPhenDelReall_" code="67"---
        --- ready="yes" index="0">
        <wfData def="Kd_Id" flag="0" data="17"/>
    </WeakForm>
    <WeakForm id="21" def="WF_gPhenAttrib1_" code="65"---
        --- ready="yes" index="0">
        <wfData def="Kd_Id" flag="0" data="17"/>
        <wfData def="C_Id" flag="0" data="16"/>

```

```

    </WeakForm>
</WFexec>
<WFexec def="DCompute_SolveUdErr" execCode="6">
    <WeakForm id="22" def="WF_gPhenDefinitive2" code="64"---
        --- ready="yes" index="0">
        <wfData def="Kd_Id" flag="0" data="17"/>
    </WeakForm>
    <WeakForm id="23" def="WF_gPhenSolverUdnp1" code="73"---
        --- ready="yes" index="0">
        <wfData def="Kd_Id" flag="0" data="17"/>
        <wfData def="Udnp1" flag="0" data="21"/>
        <wfData def="Fd" flag="0" data="18"/>
    </WeakForm>
    <WeakForm id="24" def="WF_DgPhenCompute_errD" code="40"---
        --- ready="yes" index="0">
        <wfData def="Udk_Id" flag="0" data="25"/>
        <wfData def="Udnp1_Id" flag="0" data="21"/>
        <wfData def="err_D_out" flag="1" data="err_D_gph2"/>
    </WeakForm>
    <WeakForm id="25" def="WF_gPhenAttrib2" code="65"---
        --- ready="yes" index="0">
        <wfData def="Udk_Id" flag="0" data="25"/>
        <wfData def="Udnp1_Id" flag="0" data="21"/>
    </WeakForm>
</WFexec>
<WFexec def="DCompute_Udnpn1AttribErr" execCode="7">
    <WeakForm id="26" def="WF_gPhenCompute_errD" code="40"---
        --- ready="yes" index="0">
        <wfData def="Udnp1_Id" flag="0" data="21"/>
        <wfData def="Udex_Id" flag="0" data="24"/>
        <wfData def="err_D_out" flag="0" data="0.0e0"/>
    </WeakForm>
    <WeakForm id="27" def="WF_DgPhenComputeUdnpn1" code="71"---
        --- ready="yes" index="0">
        <wfData def="Udnpn1_Id" flag="0" data="23"/>
        <wfData def="Udnpn_Id" flag="0" data="22"/>
        <wfData def="Udnp1_Id" flag="0" data="21"/>
        <wfData def="Udn_Id" flag="0" data="20"/>
        <wfData def="delta_t" flag="1" data="delta_t"/>
    </WeakForm>
    <WeakForm id="28" def="WF_DgPhenAttrib3" code="65"---
        --- ready="yes" index="0">
        <wfData def="Udn_Id" flag="0" data="20"/>

```

```

        <wfData def="Udnp1_Id" flag="0" data="21"/>
    </WeakForm>
    <WeakForm id="29" def="WF_DgPhenAttrib4" code="65"---
        --- ready="yes" index="0">
        <wfData def="Udnp_Id" flag="0" data="22"/>
        <wfData def="Udnp1_Id" flag="0" data="23"/>
    </WeakForm>
    <WeakForm id="30" def="WF_DgPhenComputeDeltat1" code="52"---
        --- ready="yes" index="0">
        <wfData def="delta_tp3_in" flag="1" data="delta_tp3"/>
        <wfData def="delta_tg1_out" flag="1" data="delta_tg1"/>
    </WeakForm>
</WFexec>
</gPhen>

<!-- list of codes for global states owned by this Group -->
<glbSts id="15" def="M">
    <glbData subType="1" type="2"/>
    <!-- list of Phen objects sharing this global state -->
    <owner vectorFieldId="3" phnId="3">
        <vectorFields vecfieldId="0"/>
    </owner>
    <coupled vectorFieldId="6" phnId="3"/>
</glbSts>
<glbSts id="16" def="C">
    <glbData subType="1" type="2"/>
    <!-- list of Phen objects sharing this global state -->
    <owner vectorFieldId="3" phnId="3">
        <vectorFields vecfieldId="1"/>
    </owner>
    <coupled vectorFieldId="6" phnId="3"/>
</glbSts>
<glbSts id="17" def="Kd"> <glbData subType="1" type="2"/>
    <!-- list of Phen objects sharing this global state -->
    <owner vectorFieldId="3" phnId="3">
        <vectorFields vecfieldId="2"/>
    </owner>
    <coupled vectorFieldId="6" phnId="3"/>
</glbSts>
<glbSts id="18" def="Fd">
    <glbData subType="0" type="1"/>
    <!-- list of Phen objects sharing this global state -->
    <owner vectorFieldId="3" phnId="3"/>

```

```

</glbSts>
<glbSts id="19" def="FdAux">
  <glbData subType="0" type="1"/>
  <!-- list of Phen objects sharing this global state -->
  <owner vectorFieldId="4" phnId="3"/>
</glbSts>
<glbSts id="20" def="Udn">
  <glbData subType="0" type="1"/>
  <!-- list of Phen objects sharing this global state -->
  <owner vectorFieldId="5" phnId="3"/>
</glbSts>
<glbSts id="21" def="Udnp1">
  <glbData subType="0" type="1"/>
  <!-- list of Phen objects sharing this global state -->
  <owner vectorFieldId="6" phnId="3"/>
</glbSts>
<glbSts id="22" def="Udnp">
  <glbData subType="0" type="1"/>
  <!-- list of Phen objects sharing this global state -->
  <owner vectorFieldId="7" phnId="3"/>
</glbSts>
<glbSts id="23" def="Udpnp1">
  <glbData subType="0" type="1"/>
  <!-- list of Phen objects sharing this global state -->
  <owner vectorFieldId="8" phnId="3"/>
</glbSts>
<glbSts id="24" def="Udex">
  <glbData subType="0" type="1"/>
  <!-- list of Phen objects sharing this global state -->
  <owner vectorFieldId="9" phnId="3"/>
</glbSts>
<glbSts id="25" def="Udk">
  <glbData subType="0" type="1"/>
  <!-- list of Phen objects sharing this global state -->
  <owner vectorFieldId="10" phnId="3"/>
</glbSts>
<GroupTask id="0" def="Grp1_gtsk0">
  <grpTskDt def="DCompute_delta_tp3" execCode="0" phnId="3"/>
  <grpTskDt def="DCompute_delta_tgl" execCode="0" phnId="2"/>
</GroupTask>
<GroupTask id="1" def="Grp1_gtsk1">
  <grpTskDt def="DCompute_SizeAllocAll" execCode="1" phnId="2"/>
  <grpTskDt def="DiffInitManyMC" execCode="1" phnId="3"/>

```

```

    <grpTskDt def="DCompute_FdMKd" execCode="2" phenId="2"/>
    <grpTskDt def="DiffFdKdBC" execCode="2" phenId="3"/>
    <grpTskDt def="DCompute_KdDefinitSolver" execCode="3" ---
        --- phenId="2"/>
</GroupTask>
<GroupTask id="2" def="Grp1_gtsk2">
    <grpTskDt def="DCompute_FdAuxInitIter" execCode="4" phenId="2"/>
    <grpTskDt def="DiffFdAuxBC" execCode="3" phenId="3"/>
</GroupTask>
<GroupTask id="3" def="Grp1_gtsk3">
    <grpTskDt def="DCompute_InitFdKd" execCode="5" phenId="2"/>
    <grpTskDt def="DiffFdKdBC0" execCode="4" phenId="3"/>
    <grpTskDt def="DCompute_SolveUdErr" execCode="6" phenId="2"/>
</GroupTask>
<GroupTask id="4" def="Grp1_gtsk4">
    <grpTskDt def="DiffUdexDt" execCode="5" phenId="3"/>
    <grpTskDt def="DCompute_UdnpnlAttribErr" execCode="7" ---
        --- phenId="2"/>
</GroupTask>
</Group>

<Group id="2">
    <grpSystDt size="0">
        <!-- data exposed by this Group -->
    </grpSystDt>
    <gPhen id="4" code="0">
        <requiredDt def="t_1" grpIndex="0"/>
        <requiredDt def="delta_t" grpIndex="1"/>
        <!-- List of Phenomena -->
        <Phenomenon id="5"/>
    </gPhen>
    <!-- list of codes for global states owned by this Group -->
    <glbSts id="0" def="Glbph5">
        <glbData subType="0" type="1"/>
        <!-- list of Phen objects sharing this global state -->
        <phnShring vectorFieldId="0" phnId="5"/>
    </glbSts>
    <GroupTask id="0" def="Record">
        <grpTskDt def="ElRec_Mesh" execCode="90" phenId="5"/>
        <grpTskDt def="ElRec_U" execCode="91" phenId="5"/>
        <grpTskDt def="ElRec_t1_deltat" execCode="92" phenId="5"/>
        <grpTskDt def="ElRec_errU" execCode="93" phenId="5"/>
        <grpTskDt def="ElRec_Plasticity" execCode="94" phenId="5"/>
    </GroupTask>

```

```

    </GroupTask>
</Group>
<!-- End Group description -->

<Phenomenon id="1" def="Elastic" code="4">
  <phnSystDt size="3">
    <exposedDt def="delta_tpl" phnIndex="0"/>
    <requiredDt def="interval" phnIndex="1"/>
    <requiredDt def="t_1" phnIndex="2"/>
  </phnSystDt>
  <phenParam id="1" def="pP_ElElasticTensor" code="2">
    <parData def="par_E" value="0.0"/>
    <parData def="par_ni" value="0.0"/>
    <parData def="par_e" value="0.0"/>
  </phenParam>
  <phenParam id="2" def="ConstCoupling" code="3">
    <parData def="par_cpl0" value="0.0e0"/>
    <parData def="par_cpl1" value="0.0e0"/>
  </phenParam>
  <phenParam id="3" def="pP_phenDataExata" code="4">
    <parData def="u0" value="0.0e0"/>
    <parData def="v0" value="0.0e0"/>
  </phenParam>
  <phenParam id="4" def="pP_phenPlast" code="5">
    <parData def="sigy" value="0.0e0"/>
    <parData def="varphi" value="0.0e0"/>
    <parData def="aa" value="0.0e0"/>
    <parData def="bb" value="0.0e0"/>
    <parData def="dd" value="0.0e0"/>
  </phenParam>
  <dims spaceDim="2" strutDim="2"/>
  <!-- For "scalars" (nvDim = 0), contDim provides dimension. -->
  <!-- OBs.: globalStateId is different from vectorFieldId -->
  <vectorField id="0" def="Ke" phenMesh="0" nvDim="2" ---
    --- contDim="2" glbSt="0"/> <!-- Ke -->
  <vectorField id="1" def="Ue" phenMesh="0" nvDim="2" ---
    --- contDim="2" glbSt="1"/> <!-- Ue -->
  <vectorField id="2" def="Uekp1" phenMesh="0" nvDim="2" ---
    --- contDim="2" glbSt="2"/> <!-- Uekp1 -->
  <vectorField id="3" def="Ueex" phenMesh="0" nvDim="2" ---
    --- contDim="2" glbSt="3"/> <!-- Ueex -->
  <vectorField id="4" def="Fe" phenMesh="0" nvDim="2" ---
    --- contDim="2" glbSt="4"/> <!-- Fe -->

```

```

<vectorField id="5" def="FeAux" phenMesh="0" nvDim="2"
    --- --- contDim="2" glbSt="5"/> <!-- FeAux -->
<vectorField id="6" def="Xn" phenMesh="0" nvDim="3"
    --- contDim="3" glbSt="6"/> <!-- X_n -->
<vectorField id="7" def="Xnp1" phenMesh="0" nvDim="3" ---
    --- contDim="3" glbSt="7"/> <!-- X_np1 -->
<vectorField id="8" def="EpsPn" phenMesh="0" nvDim="3" ---
    --- contDim="3" glbSt="8"/> <!-- EpsP_n -->
<vectorField id="9" def="EpsPnp1" phenMesh="0" nvDim="3" ---
    --- contDim="3" glbSt="9"/> <!-- EpsP_np1 -->
<vectorField id="10" def="Trace" phenMesh="0" nvDim="1" ---
    --- contDim="1" glbSt="10"/> <!-- Trace -->
<vectorField id="11" def="Sigman" phenMesh="0" nvDim="3" ---
    --- contDim="3" glbSt="11"/> <!-- Sigma_n -->
<vectorField id="12" def="Sigmanp1" phenMesh="0" nvDim="3" ---
    --- contDim="3" glbSt="12"/> <!-- Sigma_np1 -->
<vectorField id="13" def="Pn" phenMesh="0" nvDim="1" ---
    --- contDim="1" glbSt="13"/> <!-- P_n -->
<vectorField id="14" def="Pnp1" phenMesh="0" nvDim="1" ---
    --- contDim="1" glbSt="14"/> <!-- P_np1 -->

<phenMesh id="0" maxPhenOrder="1" phenShapeFuncCode="0" ---
    --- geomMeshId="0" phenMeshGenId="0"/>
<geomMesh id="0" geomMeshGenId="0"/>
<!-- masterGeomMesh is the mesh upon which all others are based -->
<masterGeomMesh meshId="0"/>
<!-- only one geomMeshGenerator-->
<geomMeshGenerator id="0" code="1" maxGeomOrder="0" ---
    --- geomShapeFuncCode="0"/>
<!-- There might be more than one phenMeshGenerator -->
<phenMeshGenerator id="0" code="3"/>
<!-- Only one integration method per Phenomenon -->
<integMethd maxIntegOrder="0" integMethCode="0"/>

<WFexec def="ElInit" execCode="0">
    <WeakForm id="0" def="WF_ElPhenInitAll" code="0" ---
        --- ready="no" index="1">
        <wfData def="value" flag="0" data="0.0e0"/>
    </WeakForm>
    <WeakForm id="1" def="WF_ElPhenComputeDeltatp1" code="4" ---
        --- ready="yes" index="1">
        <wfData def="delta_tpl_out" flag="1" data="delta_tpl"/>
        <wfData def="par_interval" flag="1" data="interval"/>
    </WeakForm>
</WFexec>

```

```

    </WeakForm>
</WFexec>
<WFexec def="ElKeFeBC" execCode="1">
    <WeakForm id="2" def="WF_PhenSetValue" code="8" ---
        --- ready="no" index="1">
        <wfData def="Ke_Id" flag="0" data="0"/>
        <wfData def="value" flag="0" data="0.0e0"/>
    </WeakForm>
    <WeakForm id="3" def="WF_ElPhenComputeK" code="1" ---
        --- ready="no" index="1">
        <wfData def="Ke_Id" flag="0" data="0"/>
        <wfData def="PhenPar" flag="0" data="1"/>
        <wfData def="consPar" flag="0" data="2"/>
    </WeakForm>
    <WeakForm id="4" def="WF_ElPhenComputeF" code="2" ---
        --- ready="no" index="1">
        <wfData def="FeAux_Id" flag="0" data="5"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="3"/>
    </WeakForm>
    <WeakForm id="5" def="WF_ElPhenIBC_KF" code="3" ---
        --- ready="no" index="1">
        <wfData def="Ke_Id" flag="0" data="0"/>
        <wfData def="FeAux_Id" flag="0" data="5"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="3"/>
    </WeakForm>
    <WeakForm id="6" def="WF_ElPhenIBC_KFEspecial" code="3" ---
        --- ready="no" index="1">
        <wfData def="Ke_Id" flag="0" data="0"/>
        <wfData def="FeAux_Id" flag="0" data="5"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="3"/>
    </WeakForm>
</WFexec>
<WFexec def="ElFeBC" execCode="2">
    <WeakForm id="7" def="WF_ElPhenComputeFPlas" code="11" ---
        --- ready="no" index="1">
        <wfData def="F_e_Id" flag="0" data="4"/>
        <wfData def="EpsP_n_Id" flag="0" data="8"/>
    </WeakForm>
    <WeakForm id="8" def="WF_ElPhenComputeFDiff" code="6" ---
        --- ready="no" index="1">

```

```

        <wfData def="Fe_Id" flag="0" data="4"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="3"/>
        <wfData def="Udk_Id" flag="0" data="10"/>
    </WeakForm>
    <WeakForm id="9" def="WF_ElPhenIBC_FEspecial0" code="3" ---
        --- ready="no" index="1">
        <wfData def="Ke_Id" flag="0" data="0"/>
        <wfData def="Fe_Id" flag="0" data="4"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="3"/>
    </WeakForm>
</WFexec>
<WFexec def="ElStrainStressTr" execCode="3">
    <WeakForm id="10" def="WF_ElPhenComputeSigma" code="7" ---
        --- ready="no" index="1">
        <wfData def="Sigma_npl_Id" flag="0" data="12"/>
        <wfData def="Uekpl_Id" flag="0" data="2"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="3"/>
    </WeakForm>
    <WeakForm id="11" def="WF_ElPhenComputeTrace" code="9" ---
        --- ready="no" index="1">
        <wfData def="Sigma_npl_Id" flag="0" data="12"/>
        <wfData def="Trace_Id" flag="0" data="10"/>
    </WeakForm>
</WFexec>
<WFexec def="ElUeexDt" execCode="4">
    <WeakForm id="12" def="WF_ElPhenComputeSolExata" code="5" ---
        --- ready="no" index="1">
        <wfData def="Ueex_Id" flag="0" data="3"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="3"/>
    </WeakForm>
    <WeakForm id="13" def="WF_ElPhenComputeDeltatp1_" code="4" ---
        --- ready="yes" index="1">
        <wfData def="delta_tp1_out" flag="1" data="delta_tp1"/>
        <wfData def="par_Interval" flag="1" data="interval"/>
    </WeakForm>
</WFexec>
<WFexec def="ElPlast" execCode="5">
    <WeakForm id="14" def="WF_ElPhenComputePlast2" code="10" ---
        --- ready="no" index="1">

```

```

        <wfData def="Sigma_npl_Id" flag="0" data="12"/>
        <wfData def="Sigma_n_Id" flag="0" data="11"/>
        <wfData def="epsP_n_Id" flag="0" data="8"/>
        <wfData def="epsP_npl_Id" flag="0" data="9"/>
        <wfData def="X_n_Id" flag="0" data="6"/>
        <wfData def="X_npl_Id" flag="0" data="7"/>
        <wfData def="P_n_Id" flag="0" data="13"/>
        <wfData def="P_npl_Id" flag="0" data="14"/>
    </WeakForm>
</WFexec>
</Phenomenon>
<Phenomenon id="3" def="Diffusion" code="20">
    <phnSystDt size="3">
        <exposedDt def="delta_tp3" phnIndex="0"/>
        <requiredDt def="interval" phnIndex="1"/>
        <requiredDt def="t_1" phnIndex="2"/>
    </phnSystDt>
    <phenParam id="0" def="pP_DiffusionMatrix" code="4">
        <parData def="par_d0" value="0.0e0"/>
        <parData def="par_h" value="0.0e0"/>
        <parData def="par_alpha" value="0.0e0"/>
    </phenParam>
    <phenParam id="1" def="pP_MassMatrix" code="5">
        <parData def="par_rho" value="0.0e0"/>
        <parData def="par_cp" value="0.0e0"/>
    </phenParam>
    <phenParam id="2" def="pP_DiffSolExata" code="6">
        <parData def="par_w0" value="0.0e0"/>
        <parData def="par_gam" value="0.0e0"/>
        <parData def="par_c0" value="0.0e0"/>
    </phenParam>
    <phenParam id="3" def="pP_PntInt" code="7">
        <parData def="par_nptint" value="0.0e0"/>
        <parData def="par_wght0" value="0.0e0"/>
        <parData def="par_wght1" value="0.0e0"/>
        <parData def="par_x0" value="0.0e0"/>
        <parData def="par_x1" value="0.0e0"/>
        <parData def="par_x2" value="0.0e0"/>
        <parData def="par_x3" value="0.0e0"/>
        <parData def="par_x4" value="0.0e0"/>
        <parData def="par_x5" value="0.0e0"/>
    </phenParam>
</Phenomenon>
<dims spaceDim="2" strutDim="2"/>

```

```

<vectorField id="0" def="M" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="15"/> <!-- M -->
<vectorField id="1" def="C" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="16"/> <!-- C -->
<vectorField id="2" def="Kd" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="17"/> <!-- Kd -->
<vectorField id="3" def="Fd" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="18"/> <!-- Fd -->
<vectorField id="4" def="FdAux" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="19"/> <!-- FdAux -->
<vectorField id="5" def="Udn" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="20"/> <!-- Udn -->
<vectorField id="6" def="Udnpl" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="21"/> <!-- Udnpl -->
<vectorField id="7" def="Udpn" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="22"/> <!-- Udpn -->
<vectorField id="8" def="Udpnpl" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="23"/> <!-- Udpnpl -->
<vectorField id="9" def="Udex" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="24"/> <!-- Udex -->
<vectorField id="10" def="Udk" phenMesh="3" nvDim="1" ---
    --- contDim="1" glbSt="25"/> <!-- Udk -->

<phenMesh id="3" maxPhenOrder="0" phenShapeFuncCode="0" ---
    --- geomMeshId="3" phenMeshGenId="1"/>

<geomMesh id="3" geomMeshGenId="1"/>

<!-- masterGeomMesh is the mesh upon which all others are based -->
<masterGeomMesh meshId="3"/>
<!-- only one geomMeshGenerator-->
<geomMeshGenerator id="1" code="1" maxGeomOrder="0" ---
    --- geomShapeFuncCode="0"/>
<!-- There might be more than one phenMeshGenerator -->
<phenMeshGenerator id="1" code="3"/>

<!-- Only one integration method per Phenomenon -->
<integMethd maxIntegOrder="0" integMethCode="0"/>

<Wfexec def="DCompute_delta_tp3" execCode="0">
    <WeakForm id="0" def="WF_DPhenComputeDeltatp3" code="1" ---
        --- ready="yes" index="1">

```

```

        <wfData def="delta_tp3_out" flag="1" data="delta_tp3"/>
        <wfData def="par_Interval" flag="1" data="interval"/>
    </WeakForm>
</WFexec>
<WFexec def="DiffInitManyMC" execCode="1">
    <WeakForm id="1" def="WF_DPhenInitAll" code="0" ---
        --- ready="no" index="1">
        <wfData def="value" flag="0" data="0.0e0"/>
    </WeakForm>
    <WeakForm id="2" def="WF_DPhenCompute_U" code="16" ---
        --- ready="no" index="1">
        <wfData def="Udn_Id" flag="0" data="5"/>
        <wfData def="t_1" flag="1" data="t_1"/>
    </WeakForm>
    <WeakForm id="3" def="WF_DPhenCompute_M" code="2" ---
        --- ready="no" index="1">
        <wfData def="M_Id" flag="0" data="0"/>
    </WeakForm>
    <WeakForm id="4" def="WF_DPhenCompute_C" code="3" ---
        --- ready="no" index="1">
        <wfData def="C_Id" flag="0" data="1"/>
    </WeakForm>
    <WeakForm id="5" def="WF_DPhenCompute_Fn_1" code="4" ---
        --- ready="no" index="1">
        <wfData def="Fd_Id" flag="0" data="3"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="1"/>
    </WeakForm>
    <WeakForm id="6" def="WF_DPhenCompute_Fc" code="20" ---
        --- ready="no" index="1">
        <wfData def="Fd_Id" flag="0" data="3"/>
        <wfData def="Udn_Id" flag="0" data="5"/>
        <wfData def="phen" flag="0" data="1"/>
        <wfData def="Trace_Id" flag="0" data="10"/>
    </WeakForm>
    <WeakForm id="7" def="WF_DPhenCompute_Kdiff" code="17" ---
        --- ready="no" index="1">
        <wfData def="Kd_Id" flag="0" data="2"/>
        <wfData def="Udn_Id" flag="0" data="5"/>
        <wfData def="phen" flag="0" data="1"/>
    </WeakForm>
</WFexec>
<WFexec def="DiffFdMBC" execCode="2">

```

```

<WeakForm id="8" def="WF_DPhenCompute_IBC0" code="9" ---
    --- ready="no" index="1">
    <wfData def="M_Id" flag="0" data="0"/>
    <wfData def="Fd_Id" flag="0" data="3"/>
    <wfData def="t_1" flag="1" data="t_1"/>
    <wfData def="phen" flag="0" data="1"/>
</WeakForm>
</WFexec>
<WFexec def="DiffFdAuxBC" execCode="3">
    <WeakForm id="9" def="WF_DPhenCompute_Fn_10" code="4" ---
        --- ready="no" index="1">
        <wfData def="FdAux_Id" flag="0" data="4"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="1"/>
    </WeakForm>
    <WeakForm id="10" def="WF_DPhenCompute_IBC1" code="9" ---
        --- ready="no" index="1">
        <wfData def="Kd_Id" flag="0" data="2"/>
        <wfData def="FdAux_Id" flag="0" data="4"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="1"/>
    </WeakForm>
</WFexec>
<WFexec def="DiffFdKdBC0" execCode="4">
    <WeakForm id="11" def="WF_DPhenCompute_Fc0" code="20" ---
        --- ready="no" index="1">
        <wfData def="Fd_Id" flag="0" data="3"/>
        <wfData def="Udk_Id" flag="0" data="10"/>
        <wfData def="phen" flag="0" data="1"/>
        <wfData def="Trace_Id" flag="0" data="10"/>
    </WeakForm>
    <WeakForm id="12" def="WF_DPhenCompute_Kdiff0" code="17" ---
        --- ready="no" index="1">
        <wfData def="Kd_Id" flag="0" data="2"/>
        <wfData def="Udk_Id" flag="0" data="10"/>
        <wfData def="phen" flag="0" data="1"/>
    </WeakForm>
    <WeakForm id="13" def="WF_DPhenCompute_IBC2" code="9" ---
        --- ready="no" index="1">
        <wfData def="Kd_Id" flag="0" data="2"/>
        <wfData def="Fd_Id" flag="0" data="3"/>
        <wfData def="t_1" flag="1" data="t_1"/>
        <wfData def="phen" flag="0" data="1"/>

```

```

        </WeakForm>
    </WFexec>
    <WFexec def="DiffUdexDt" execCode="5">
        <WeakForm id="14" def="WF_DPhenCompute_SolExata" code="15" ---
            --- ready="no" index="1">
            <wfData def="Udex_Id" flag="0" data="9"/>
            <wfData def="t_1" flag="1" data="t_1"/>
        </WeakForm>
        <WeakForm id="15" def="WF_DPhenComputeDeltatp3_" code="1" ---
            --- ready="yes" index="1">
            <wfData def="delta_tp3_out" flag="1" data="delta_tp3"/>
            <wfData def="par_Interval" flag="1" data="interval"/>
        </WeakForm>
    </WFexec>
</Phenomenon>
<Phenomenon id="5" def="Record" code="5">
    <phnSystDt size="3">
        <requiredDt def="t_1" phnIndex="0"/>
        <requiredDt def="delta_t" phnIndex="1"/>
        <requiredDt def="err_U" phnIndex="2"/>
    </phnSystDt>
    <phenParam id="0" def="pP_Elfict" code="0">
        <parData def="teste" value="0.0e0"/>
    </phenParam>

    <dims spaceDim="2" strutDim="2"/>
    <vectorField id="0" phenMesh="5" nvDim="1" contDim="1" glbSt="0"/>
    <phenMesh id="5" maxPhenOrder="0" phenShapeFuncCode="0" ---
        --- geomMeshId="5" phenMeshGenId="2"/>
    <geomMesh id="5" geomMeshGenId="2"/>
    <masterGeomMesh meshId="5"/>
    <geomMeshGenerator id="2" code="1" maxGeomOrder="0" ---
        --- geomShapeFuncCode="0"/>
    <phenMeshGenerator id="2" code="3"/>
    <integMethd maxIntegOrder="0" integMethCode="0"/>
    <WFexec def="ElRec_Mesh" execCode="90">
        <WeakForm id="0" def="WF_ElPhenRecMesh" code="0" ---
            --- ready="yes" index="3">
            <wfData def="Mesh" flag="1" data="0"/>
        </WeakForm>
    </WFexec>
    <WFexec def="ElRec_U" execCode="91">
        <WeakForm id="1" def="WF_ElPhenRecU" code="1" ---

```

```

        --- ready="yes" index="3">
        <wfData def="phen" flag="0" data="1"/>
        <wfData def="vf_U" flag="0" data="1"/>
    </WeakForm>
</WFexec>
<WFexec def="ElRec_t1_deltat" execCode="92">
    <WeakForm id="2" def="WF_ElPhenRect1deltat" code="2" ---
        --- ready="yes" index="3">
        <wfData def="t_1_out" flag="1" data="t_1"/>
        <wfData def="delta_t_out" flag="1" data="delta_t"/>
    </WeakForm>
</WFexec>
<WFexec def="ElRec_errU" execCode="93">
    <WeakForm id="3" def="WF_ElPhenRecerrU" code="3" ---
        --- ready="yes" index="3">
        <wfData def="errU_out" flag="1" data="err_U"/>
    </WeakForm>
</WFexec>
<WFexec def="ElRec_Plasticity" execCode="94">
    <WeakForm id="4" def="WF_ElPhenRecPlasticity" code="4" ---
        --- ready="yes" index="3">
        <wfData def="phen" flag="0" data="1"/>
        <wfData def="Pn_Id" flag="0" data="13"/>
        <wfData def="t_1_out" flag="1" data="t_1"/>
    </WeakForm>
</WFexec>
</Phenomenon>
</Simulator>

```

Desenvolvimento do Simulador Elastoplástico com Difusão

Este apêndice tem o objetivo de apresentar o arquivo utilizado para a simulação, utilizando o MPhyScas, para o problema desenvolvido nesta dissertação.

```
<Simulation version="1.0" platform="Linux">
  <Properties name="Simulation Testing">
    <Responsible name="Felipe A. Cruz" email="email@email.com"/>
    <Description>Test Simulation Elastoplastic and Diffusion</Description>
    <Documentation>doc.pdf</Documentation>
  </Properties>

  <!-- Global Geometry -->
  <geometry strutDim="2" spaceDim="2" ifFile="yes" id="0">
    <geomFile name="geomSim.xml"/>
  </geometry>

  <!-- Kernel description -->
  <Kernel id="0">
    <KernelAlgth id="0" code="6">
      <sysPar value="2.0e0" def="interval"/>
    </KernelAlgth>
  </Kernel>

  <!-- Block description -->
  <Block id="0">
    <blkAlgth id="0" code="28">
    </blkAlgth>
    <blkAlgth id="1" code="29">
    </blkAlgth>
    <blkAlgth id="2" code="30">
    </blkAlgth>
    <blkAlgth id="3" code="31">
      <algthPar value="1.0e-8" def="iterTol"/>
      <algthPar value="10.0e0" def="iterMax"/>
    </blkAlgth>
  </Block>
</Simulation>
```

```

        <blkAlgth id="4" code="32">
        </blkAlgth>
</Block>
<Block id="1">
    <blkAlgth id="0" code="25">
    </blkAlgth>
</Block>
<Group id="0">
    <gPhen id="0">
        <solver id="0" code="4" def="ElSolver_U"/>
        <WFexec def="ElCompute_Solver_U" execCode="5">
            <WeakForm id="8" def="WF_gPhenSolver">
                <wfParam value="0" def="solverId" flag="0"/>
                <wfParam value="GMM++" def="type" flag="0"/>
                <wfParam value="0.0e0" def="method" flag="0"/>
                <wfParam value="1.0e0" def="preCond" flag="0"/>
                <wfParam value="1.0e-10" def="iter" flag="0"/>
                <wfParam value="1.0e4" def="restart" flag="0"/>
                <wfParam value="1.0e5" def="maxIter" flag="0"/>
                <wfParam value="8.0e0" def="kFillIn" flag="0"/>
                <wfParam value="1.0e-3" def="threshold" flag="0"/>
            </WeakForm>
        </WFexec>
    </gPhen>
</Group>
<Group id="1">
    <gPhen id="1">
        <solver id="0" code="6" def="DiffSolver_U"/>
        <WFexec def="DiffCompute_SolverUpn" execCode="3">
            <WeakForm id="10" def="WF_DiffgPhenSolverUpn">
                <wfParam value="0" def="solverId" flag="0"/>
                <wfParam value="GMM++" def="type" flag="0"/>
                <wfParam value="0.0e0" def="method" flag="0"/>
                <wfParam value="1.0e0" def="preCond" flag="0"/>
                <wfParam value="1.0e-10" def="iter" flag="0"/>
                <wfParam value="1.0e4" def="restart" flag="0"/>
                <wfParam value="1.0e5" def="maxIter" flag="0"/>
                <wfParam value="2.0e0" def="kFillIn" flag="0"/>
                <wfParam value="1.0e-3" def="threshold" flag="0"/>
            </WeakForm>
        </WFexec>
        <WFexec def="DiffCompute_SolverUnpl" execCode="6">
            <WeakForm id="23" def="WF_DiffgPhenSolverUnpl">

```

```

        <wfParam value="0" def="solverId" flag="0"/>
        <wfParam value="GMM++" def="type" flag="0"/>
        <wfParam value="0.0e0" def="method" flag="0"/>
        <wfParam value="1.0e0" def="preCond" flag="0"/>
        <wfParam value="1.0e-10" def="iter" flag="0"/>
        <wfParam value="1.0e4" def="restart" flag="0"/>
        <wfParam value="1.0e5" def="maxIter" flag="0"/>
        <wfParam value="2.0e0" def="kFillIn" flag="0"/>
        <wfParam value="1.0e-3" def="threshold" flag="0"/>
    </WeakForm>
</WFexec>
</gPhen>
</Group>
<Phenomenon id="1" code="4" def="Elastic">
    <phenParam id="1" def="pP_ElElasticTensor">
        <parData value="2.1e11" def="par_E"/>
        <parData value="0.3e0" def="par_ni"/>
        <parData value="5.2e0" def="par_e"/>
    </phenParam>
    <phenParam id="2" def="ConstCoupling">
        <parData value="1.0e0" def="par_cpl0"/>
        <parData value="1.0e0" def="par_cpl1"/>
    </phenParam>
    <phenParam id="3" def="pP_phenDataExata">
        <parData value="1.0e-5" def="u0"/>
        <parData value="1.0e-5" def="v0"/>
    </phenParam>
    <phenParam id="4" def="pP_phenPlast">
        <parData value="1.0e6" def="sigy"/>
        <parData value="1.0e0" def="varphi"/>
        <parData value="1.0e0" def="aa"/>
        <parData value="1.0e0" def="bb"/>
        <parData value="1.0e0" def="dd"/>
    </phenParam>
    <geomGraph id="0">
        <geomRel geomId="8"/>
        <boundingBox y1="12.0e0" x1="12.0e0" y0="-1.0e0" x0="-1.0e0"/>
    </geomGraph>
    <geomMeshShring phenId="3">
        <geomEnt geomEntId="8"/>
    </geomMeshShring>
    <geomMeshShring phenId="5">
        <geomEnt geomEntId="8"/>
    </geomMeshShring>

```

```

</geomMeshShring>
<geomMeshGenData id="0">
  <gmngData def="geomTol" data="1.0e-6"/>
  <gmngData def="maxSize_1D" data="0.125e0"/>
  <gmngData def="minSize_1D" data="0.125e0"/>
  <gmngData def="maxSize_2D" data="0.125e0"/>
  <gmngData def="minSize_2D" data="0.125e0"/>
  <gmngData def="controlA_2D" data="0.8e0"/>
  <gmngData def="controlB_2D" data="1.2e0"/>
  <gmngData def="scale_2D" data="0.7e0"/>
  <gmngData def="scalePower_2D" data="2.0e0"/>
</geomMeshGenData>
<phenMeshGenData id="0">
  <pmngData def="ordAp" data="1"/>
</phenMeshGenData>

<WFexec def="ElInit" execCode="0">
  <WeakForm id="0" def="WF_ElPhenInit_All">
    <wfData def="zeroInit" flag="0" data="2"/>
    <wfData def="value" flag="0" data="0.0e0"/>
    <newWFobjects/>
  </WeakForm>
  <WeakForm id="1" def="WF_ElPhenComputeDeltatp1">
    <wfData def="geom" flag="0" data="8"/>
    <wfData def="nInt" flag="0" data="40"/>
    <newWFobjects/>
  </WeakForm>
</WFexec>
<WFexec def="ElKeFeBC" execCode="1">
  <WeakForm id="2" def="WF_PhenSetValue">
    <wfData def="value" flag="0" data="0.0e0"/>
    <newWFobjects/>
  </WeakForm>
  <WeakForm id="3" def="WF_ElPhenComputeK">
    <wfData def="geom" flag="0" data="8"/>
    <wfData def="parCons" flag="0" data="1"/>
    <newWFobjects/>
  </WeakForm>
  <WeakForm id="4" def="WF_ElPhenComputeF">
    <wfData def="geom" flag="0" data="8"/>
    <wfData def="phen_parExat" flag="0" data="3"/>
    <wfData def="phen_parElast" flag="0" data="1"/>
    <wfData def="pP_DiffSolExata" flag="0" data="2"/>
  </WeakForm>
</WFexec>

```

```

    <newWFobjects/>
</WeakForm>
<WeakForm id="5" def="WF_ElPhenIBC_KF">
  <wfData def="geom" flag="0" data="8"/>
  <wfData def="type" flag="0" data="1"/>
  <wfData def="direction" flag="0" data="2"/>
  <wfData def="geomEnt" flag="0" data="4"/>
  <wfData def="valuesAreGiven?" flag="0" data="no"/>
  <wfData def="phen_parExat" flag="0" data="3"/>
  <wfData def="phen_parElast" flag="0" data="1"/>
  <wfData def="pP_DiffSolExata" flag="0" data="2"/>
  <wfData def="ifEspecial" flag="0" data="0"/>
</newWFobjects>
  <newWeakForm def="WF_ElPhenIBC_KF_2" newId="15">
    <wfData def="geom" flag="0" data="8"/>
    <wfData def="type" flag="0" data="1"/>
    <wfData def="direction" flag="0" data="2"/>
    <wfData def="geomEnt" flag="0" data="5"/>
    <wfData def="valuesAreGiven?" flag="0" data="no"/>
    <wfData def="phen_parExat" flag="0" data="3"/>
    <wfData def="phen_parElast" flag="0" data="1"/>
    <wfData def="pP_DiffSolExata" flag="0" data="2"/>
    <wfData def="ifEspecial" flag="0" data="0"/>
  </newWeakForm>
  <newWeakForm def="WF_ElPhenIBC_KF_3" newId="16">
    <wfData def="geom" flag="0" data="8"/>
    <wfData def="type" flag="0" data="1"/>
    <wfData def="direction" flag="0" data="2"/>
    <wfData def="geomEnt" flag="0" data="6"/>
    <wfData def="valuesAreGiven?" flag="0" data="no"/>
    <wfData def="phen_parExat" flag="0" data="3"/>
    <wfData def="phen_parElast" flag="0" data="1"/>
    <wfData def="pP_DiffSolExata" flag="0" data="2"/>
    <wfData def="ifEspecial" flag="0" data="0"/>
  </newWeakForm>
</newWFobjects>
</WeakForm>
<WeakForm id="6" def="WF_ElPhenIBC_KFEspecial">
  <wfData def="geom" flag="0" data="8"/>
  <wfData def="type" flag="0" data="0"/>
  <wfData def="direction" flag="0" data="2"/>
  <wfData def="geomEnt" flag="0" data="7"/>
  <wfData def="valuesAreGiven?" flag="0" data="no"/>

```

```

        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <wfData def="ifEspecial" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElFeBC" execCode="2">
    <WeakForm id="7" def="WF_ElPhenComputeFPlas">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="8" def="WF_ElPhenComputeFDiff">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="9" def="WF_ElPhenIBC_FEspecial0">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="type" flag="0" data="0"/>
        <wfData def="direction" flag="0" data="2"/>
        <wfData def="geomEnt" flag="0" data="7"/>
        <wfData def="valuesAreGiven?" flag="0" data="no"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <wfData def="ifEspecial" flag="0" data="2"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElStrainStressTr" execCode="3">
    <WeakForm id="10" def="WF_ElPhenComputeSigma">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="11" def="WF_ElPhenComputeTrace">
        <wfData def="geom" flag="0" data="8"/>

```

```

        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElUeexDt" execCode="4">
    <WeakForm id="12" def="WF_ElPhenComputeSolExata">
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="13" def="WF_ElPhenComputeDeltatp1_">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="nInt" flag="0" data="40"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElPlast" execCode="5">
    <WeakForm id="14" def="WF_ElPhenComputePlast2">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="phen_parPlast" flag="0" data="4"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
</Phenomenon>
<Phenomenon id="3" code="20" def="Diffusion">
    <phenParam id="0" def="pP_DiffusionMatrix">
        <parData value="1.0e0" def="par_d0"/>
        <parData value="0.0e0" def="par_h"/>
        <parData value="1.94e10" def="par_alpha"/>
    </phenParam>
    <phenParam id="1" def="pP_MassMatrix">
        <parData value="1.0e0" def="par_rho"/>
        <parData value="1.0e0" def="par_cp"/>
    </phenParam>
    <phenParam id="2" def="pP_DiffSolExata">
        <parData value="1.0e-5" def="par_w0"/>
        <parData value="1.0e0" def="par_gam"/>
        <parData value="0.0e0" def="par_c0"/>
    </phenParam>
    <phenParam id="3" def="pP_PntInt">
        <parData value="3.0e0" def="par_nptint"/>

```

```

    <parData value="0.3333333333333333e0" def="par_wght0"/>
    <parData value="0.5000000000000000e0" def="par_x0"/>
    <parData value="0.5000000000000000e0" def="par_x1"/>
    <parData value="0.0000000000000000e0" def="par_x2"/>
  </phenParam>
  <geomGraph id="0">
    <geomRel geomId="8"/>
    <boundingBox y1="2.0e0" x1="2.0e0" y0="-1.0e0" x0="-1.0e0"/>
  </geomGraph>
  <geomMeshGenData id="1">
    <gmgData def="geomTol" data="1.0e-6"/>
    <gmgData def="maxSize_1D" data="0.05e0"/>
    <gmgData def="minSize_1D" data="0.05e0"/>
    <gmgData def="maxSize_2D" data="0.05e0"/>
    <gmgData def="minSize_2D" data="0.05e0"/>
    <gmgData def="controlA_2D" data="0.8e0"/>
    <gmgData def="controlB_2D" data="1.2e0"/>
    <gmgData def="scale_2D" data="0.7e0"/>
    <gmgData def="scalePower_2D" data="2.0e0"/>
  </geomMeshGenData>
  <phenMeshGenData id="1">
    <pmgData def="ordAp" data="1"/>
  </phenMeshGenData>

  <WFexec def="DCompute_delta_tp3" execCode="0">
    <WeakForm id="0" def="WF_DPhenComputeDeltatp3">
      <wfData def="geom" flag="0" data="8"/>
      <wfData def="nInt" flag="0" data="40"/>
      <newWFobjects/>
    </WeakForm>
  </WFexec>

  <WFexec def="DiffInitManyMC" execCode="1">
    <WeakForm id="1" def="WF_DPhenInitAll">
      <wfData def="zeroInit" flag="0" data="2"/>
      <wfData def="value" flag="0" data="0.0e0"/>
      <newWFobjects/>
    </WeakForm>
    <WeakForm id="2" def="WF_DPhenCompute_U">
      <wfData def="geom" flag="0" data="8"/>
      <wfData def="pP_DiffSolExata" flag="0" data="2"/>
      <newWFobjects/>
    </WeakForm>
    <WeakForm id="3" def="WF_DPhenCompute_M">

```

```

        <wfData def="geom" flag="0" data="8"/>
        <wfData def="pP_MassMatrix" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="4" def="WF_DPhenCompute_C">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="5" def="WF_DPhenCompute_Fn_1">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <wfData def="pP_MassMatrix" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="6" def="WF_DPhenCompute_Fc">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="7" def="WF_DPhenCompute_Kdiff">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="DiffFdKdBC" execCode="2">
    <WeakForm id="8" def="WF_DPhenCompute_IBC0">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="geomEnt" flag="0" data="4"/>
        <wfData def="type" flag="0" data="1"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <wfData def="pP_MassMatrix" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="ifEspecial" flag="0" data="0"/>
        <newWFobjects>

```

```

<newWeakForm def="WF_DPhenCompute_IBC0" newId="16">
  <wfData def="geom" flag="0" data="8"/>
  <wfData def="geomEnt" flag="0" data="5"/>
  <wfData def="type" flag="0" data="1"/>
  <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
  <wfData def="pP_MassMatrix" flag="0" data="1"/>
  <wfData def="pP_DiffSolExata" flag="0" data="2"/>
  <wfData def="phen_parElast" flag="0" data="1"/>
  <wfData def="phen_parExat" flag="0" data="3"/>
  <wfData def="ifEspecial" flag="0" data="0"/>
</newWeakForm>
<newWeakForm def="WF_DPhenCompute_IBC0" newId="17">
  <wfData def="geom" flag="0" data="8"/>
  <wfData def="geomEnt" flag="0" data="6"/>
  <wfData def="type" flag="0" data="1"/>
  <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
  <wfData def="pP_MassMatrix" flag="0" data="1"/>
  <wfData def="pP_DiffSolExata" flag="0" data="2"/>
  <wfData def="phen_parElast" flag="0" data="1"/>
  <wfData def="phen_parExat" flag="0" data="3"/>
  <wfData def="ifEspecial" flag="0" data="0"/>
</newWeakForm>
<newWeakForm def="WF_DPhenCompute_IBC0" newId="18">
  <wfData def="geom" flag="0" data="8"/>
  <wfData def="geomEnt" flag="0" data="7"/>
  <wfData def="type" flag="0" data="0"/>
  <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
  <wfData def="pP_MassMatrix" flag="0" data="1"/>
  <wfData def="pP_DiffSolExata" flag="0" data="2"/>
  <wfData def="phen_parElast" flag="0" data="1"/>
  <wfData def="phen_parExat" flag="0" data="3"/>
  <wfData def="ifEspecial" flag="0" data="1"/>
</newWeakForm>
</newWFobjects>
</WeakForm>
</WFexec>
<WFexec def="DiffFdAuxBC" execCode="3">
  <WeakForm id="9" def="WF_DPhenCompute_Fn_10">
    <wfData def="geom" flag="0" data="8"/>
    <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
    <wfData def="pP_MassMatrix" flag="0" data="1"/>
    <wfData def="pP_DiffSolExata" flag="0" data="2"/>
    <wfData def="phen_parExat" flag="0" data="3"/>
  </WeakForm>
</WFexec>

```

```

        <wfData def="phen_parElast" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="10" def="WF_DPhenCompute_IBC1">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="geomEnt" flag="0" data="4"/>
        <wfData def="type" flag="0" data="1"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <wfData def="pP_MassMatrix" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="ifEspecial" flag="0" data="0"/>
        <newWFobjects>
            <newWeakForm def="WF_DPhenCompute_IBC1" newId="19">
                <wfData def="geom" flag="0" data="8"/>
                <wfData def="geomEnt" flag="0" data="5"/>
                <wfData def="type" flag="0" data="1"/>
                <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
                <wfData def="pP_MassMatrix" flag="0" data="1"/>
                <wfData def="pP_DiffSolExata" flag="0" data="2"/>
                <wfData def="phen_parElast" flag="0" data="1"/>
                <wfData def="phen_parExat" flag="0" data="3"/>
                <wfData def="ifEspecial" flag="0" data="0"/>
            </newWeakForm>
            <newWeakForm def="WF_DPhenCompute_IBC1" newId="20">
                <wfData def="geom" flag="0" data="8"/>
                <wfData def="geomEnt" flag="0" data="6"/>
                <wfData def="type" flag="0" data="1"/>
                <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
                <wfData def="pP_MassMatrix" flag="0" data="1"/>
                <wfData def="pP_DiffSolExata" flag="0" data="2"/>
                <wfData def="phen_parElast" flag="0" data="1"/>
                <wfData def="phen_parExat" flag="0" data="3"/>
                <wfData def="ifEspecial" flag="0" data="0"/>
            </newWeakForm>
        </newWFobjects>
    </WeakForm>
</WFexec>
<WFexec def="DiffFdKdBC0" execCode="4">
    <WeakForm id="11" def="WF_DPhenCompute_Fc0">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>

```

```

        <wfData def="phen_parElast" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="12" def="WF_DPhenCompute_Kdiff0">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="13" def="WF_DPhenCompute_IBC2">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="geomEnt" flag="0" data="7"/>
        <wfData def="type" flag="0" data="0"/>
        <wfData def="pP_DiffusionMatrix" flag="0" data="0"/>
        <wfData def="pP_MassMatrix" flag="0" data="1"/>
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <wfData def="phen_parElast" flag="0" data="1"/>
        <wfData def="phen_parExat" flag="0" data="3"/>
        <wfData def="ifEspecial" flag="0" data="0"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="DiffUdexDt" execCode="5">
    <WeakForm id="14" def="WF_DPhenCompute_SolExata">
        <wfData def="pP_DiffSolExata" flag="0" data="2"/>
        <newWFobjects/>
    </WeakForm>
    <WeakForm id="15" def="WF_DPhenComputeDeltatp3_">
        <wfData def="geom" flag="0" data="8"/>
        <wfData def="nInt" flag="0" data="40"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
</Phenomenon>
<Phenomenon id="5" code="5" def="Record">
    <phenParam id="0" def="pP_Elfict">
        <parData value="1.0e0" def="teste"/>
    </phenParam>
    <geomGraph id="1">
        <geomRel geomId="8"/>
        <boundingBox y1="12.0e0" x1="12.0e0" y0="-1.0e0" x0="-1.0e0"/>
    </geomGraph>
    <geomMeshShring phenId="1">

```

```

    <geomEnt geomEntId="8"/>
</geomMeshShring>
<geomMeshGenData id="2">
    <gmgData def="geomTol" data="1.0e-6"/>
    <gmgData def="maxSize_1D" data="0.009e0"/>
    <gmgData def="minSize_1D" data="0.009e0"/>
    <gmgData def="maxSize_2D" data="0.009e0"/>
    <gmgData def="minSize_2D" data="0.009e0"/>
    <gmgData def="controlA_2D" data="0.8e0"/>
    <gmgData def="controlB_2D" data="1.2e0"/>
    <gmgData def="scale_2D" data="0.7e0"/>
    <gmgData def="scalePower_2D" data="2.0e0"/>
</geomMeshGenData>
<phenMeshGenData id="2">
    <pmgData def="ordAp" data="1"/>
</phenMeshGenData>
<WFexec def="ElRec_Mesh" execCode="90">
    <WeakForm id="0" def="WF_ElPhenRecMesh">
        <wfData def="Mesh" flag="1" data="0"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElRec_U" execCode="91">
    <WeakForm id="1" def="WF_ElPhenRecU">
        <wfData def="U_out" flag="0" data="1"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElRec_t1_deltat" execCode="92">
    <WeakForm id="2" def="WF_ElPhenRect1deltat">
        <wfData def="t_1_out" flag="1" data="t_1"/>
        <wfData def="delta_t_out" flag="1" data="delta_t"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElRec_errU" execCode="93">
    <WeakForm id="3" def="WF_ElPhenRecerrU">
        <wfData def="errU_out" flag="1" data="err_U"/>
        <newWFobjects/>
    </WeakForm>
</WFexec>
<WFexec def="ElRec_Plasticity" execCode="94">
    <WeakForm id="4" def="WF_ElPhenRecPlasticity">

```

```
        <newWFobjects/>
      </WeakForm>
    </WFexec>
  </Phenomenon>
</Simulation>
```