

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS EDUARDO VALLADARES DA MOTA

SIMULAÇÃO NUMÉRICA DE CISALHAMENTO SOBRE UM CORPO ELÁSTICO:
Problema da Deformação

RIO DE JANEIRO
2026

CARLOS EDUARDO VALLADARES DA MOTA

SIMULAÇÃO NUMÉRICA DE CISALHAMENTO SOBRE UM CORPO ELÁSTICO:
Problema da Deformação

Trabalho de conclusão de curso de graduação
apresentado ao Instituto de Computação da
Universidade Federal do Rio de Janeiro como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Orientador: Prof. Marcello Goulart Teixeira

RIO DE JANEIRO

2026

CIP - Catalogação na Publicação

M176s Mota, Carlos Eduardo Valladares da
Simulação Numérica de Cisalhamento sobre um Corpo
Elástico: Problema da Deformação / Carlos Eduardo
Valladares da Mota. -- Rio de Janeiro, 2026.
37 f.

Orientador: Marcello Goulart Teixeira.
Trabalho de conclusão de curso (graduação) -
Universidade Federal do Rio de Janeiro, Instituto
de Computação, Bacharel em Ciência da Computação,
2026.

1. álgebra de tensores. 2. cisalhamento. 3.
material Mooney-Rivlin. 4. método ALI. 5. método
dos elementos finitos. I. Teixeira, Marcello
Goulart, orient. II. Título.


CARLOS EDUARDO VALLADARES DA MOTA

SIMULAÇÃO NUMÉRICA DE CISALHAMENTO SOBRE UM CORPO ELÁSTICO:
Problema da Deformação


Trabalho de conclusão de curso de graduação
apresentado ao Instituto de Computação da
Universidade Federal do Rio de Janeiro como
parte dos requisitos para obtenção do grau de
Bacharel em Ciência da Computação.

Aprovado em 24 de Fevereiro de 2026


BANCA EXAMINADORA:

Documento assinado digitalmente
 **MARCELLO GOULART TEIXEIRA**
Data: 27/02/2026 10:14:55-0300
Verifique em <https://validar.iti.gov.br>

Marcello Goulart Teixeira
Doutor (Universidade Federal do Rio de
Janeiro)

Documento assinado digitalmente
 **AMAURY ALVAREZ CRUZ**
Data: 27/02/2026 10:47:47-0300
Verifique em <https://validar.iti.gov.br>

Amaury Alvarez Cruz
Doutor (Universidade Federal do Rio de
Janeiro)

Documento assinado digitalmente
 **MAURO ANTONIO RINCON**
Data: 27/02/2026 13:45:19-0300
Verifique em <https://validar.iti.gov.br>

Mauro Antonio Rincon
Doutor (Universidade Federal do Rio de
Janeiro)

RESUMO

Este presente trabalho tem como objetivo apresentar as técnicas matemáticas e computacionais envolvidas na simulação numérica da deformação de um objeto constituído por um material elástico linear, de acordo com modelos da Física Matemática para problemas elípticos (estacionários) de elasticidade linear. As principais ferramentas utilizadas na modelagem e solução teóricas do problema são representação e álgebra de tensores, o método dos elementos finitos e o método das aproximações lineares incrementais (ALI) para simulação de grandes deformações através de incrementos de pequenas deformações. Escolheu-se um caso particular de cisalhamento sob uma malha retangular para fins de validação do resultado numérico gerado pela implementação computacional, dado que a solução analítica do problema é conhecida e que há uma simulação numérica de uma tese de mestrado que trata deste mesmo problema. Os resultados indicam que a solução gerada pela nova implementação está de acordo com o resultado teórico esperado, além de o programa permitir uma maior flexibilidade de alteração de parâmetros e casos de entrada quando comparado a simulação numérica de referência.

Palavras-chave: método dos elementos finitos; material Mooney-Rivlin; álgebra de tensores; método ALI; cisalhamento.

ABSTRACT

The present work aims to present the mathematical and computational techniques involved in the numerical simulation of an object constituted by a linear elastic material, according to Mathematical Physics models for elliptic problems (stationary) on linear elasticity. The main tools used on the problem modeling and solving are tensor representation and algebra, the finite elements method and the Successive Linear Approximations (ALS) method for big deformations simulations through incremental small deformations. The particular case of shear over a rectangular mesh was chosen as the problem to be solved, such that it is possible to validate the numerical result obtained from the computational implementation, given that the analytical solution is known and also that there is a numerical simulation published in a master thesis of such problem. The results indicate that the solution agrees with the expected theoretical ones, and also that the program allows a bigger flexibility on parameter modifications and input data when compared with the numerical simulation of reference.

Keywords: finite elements method; Mooney-Rivlin material; tensor algebra; ALS method; shear.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de deformação	16
Figura 2 – Mapeamento entre elemento finito qualquer e elemento padrão	25
Figura 3 – Ambiente notebook	29
Figura 4 – Trecho do cálculo da K^e	30
Figura 5 – Diagrama de cisalhamento	31
Figura 6 – Solução nos passos 0, 60, 100 e 140	32
Figura 7 – Erro relativo do experimento	33
Figura 8 – Erro relativo para cada incremento $\Delta\tau$, malha 2×2	34
Figura 9 – Erro relativo para cada incremento $\Delta\tau$, malha 10×10	35
Figura 10 – Erro relativo para cada incremento $\Delta\tau$, malha 40×40	35

LISTA DE TABELAS

Tabela 1 – Erro relativo máximo dos experimentos	34
--	----

LISTA DE SIGLAS

ALI	Aproximações Lineares Incrementais
EDP	Equação Diferencial Parcial
i.e.	isto é
MEF	Método dos Elementos Finitos
PVC	Problema de Valor de Contorno
UTF-8	Unicode Transformation Format 8

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVO	9
1.2	REVISÃO BIBLIOGRÁFICA	9
2	BREVE INTRODUÇÃO A TENSORES E TEORIA DA DE- FORMAÇÃO	11
2.1	ÁLGEBRA E CÁLCULO TENSORIAL	11
2.2	CINEMÁTICA DE DEFORMAÇÕES FINITAS	15
3	O PROBLEMA DE DEFORMAÇÃO	18
3.1	O PROBLEMA DE VALOR DE CONTORNO - FORMULAÇÃO FORTE	18
3.2	FORMULAÇÃO FRACA	19
3.3	FORMULAÇÃO MATRICIAL	21
3.4	MONTAGEM DO SISTEMA LINEAR	23
3.5	O MÉTODO DAS APROXIMAÇÕES LINEARES INCREMENTAIS	27
4	IMPLEMENTAÇÃO	29
5	O PROBLEMA DO CISALHAMENTO	31
5.1	EXPERIMENTOS NUMÉRICOS E RESULTADOS	32
6	CONCLUSÃO	36
	REFERÊNCIAS	37

1 INTRODUÇÃO

O método dos elementos finitos (MEF) é atualmente um dos principais métodos para a resolução numérica de equações diferenciais parciais (EDPs). Inúmeros problemas da área de computação geram equações diferenciais para as quais não conhecemos soluções analíticas. Através do MEF é possível, porém, gerar soluções numéricas de grande utilidade para o entendimento de sistemas e fenômenos descritos por estas equações. E para além da computação, muitas outras áreas se deparam com casos parecidos (de EDPs sem solução analítica, mas de suma relevância para a área e em que soluções numéricas são de grande ajuda). Áreas como física, engenharia e astronomia são alguns exemplos em que muitos problemas são modelados por EDPs e frequentemente deparam-se com este cenário.

Dentre estes problemas frequentemente encontram-se fenômenos físicos de deformação com condições de contorno e geometria não-triviais. Simular numericamente este tipo de problema auxilia o processo de busca por soluções, descartando experimentos físicos reais que não se mostram promissores e indicando direções mais prováveis onde pode existir uma solução.

1.1 OBJETIVO

O objetivo deste trabalho é estudar um problema simples de deformação, de forma a desenvolver a teoria necessária para abordá-lo e validar as implementações numéricas feitas. Com isso, é possível fazer simulações numéricas para problemas mais difíceis, uma vez tendo posse de um programa de simulação estável e correto.

1.2 REVISÃO BIBLIOGRÁFICA

Três obras principais embasam este trabalho, sendo elas:

- *Introdução ao Método dos Elementos Finitos* (RINCON; I-SHIH, 2020)
- *Introduction to Continuum Mechanics* (I-SHIH, 2019)
- *Modelagem Elastoplástica Utilizando o Método de Aproximações Lineares Incrementais* (PEREIRA, 2015)

Na primeira obra residem os elementos principais quanto ao método dos elementos finitos, desde as formulações forte, fraca e matricial até alguns aspectos de implementação como estruturas de mapeamento local-global de nós e de numeração e acesso das equações.

Na segunda obra estão os conceitos fundamentais de álgebra e cálculo com tensores, como também de teoria de deformação (Capítulos 1 e 2).

Por fim, na terceira obra estão derivações de resultados importantes para o problema de deformação de materiais Mooney-Rivlin compressíveis e quase incompressíveis, a apresentação do método ALI e a simulação numérica na qual se baseará o presente trabalho.

2 BREVE INTRODUÇÃO A TENSORES E TEORIA DA DEFORMAÇÃO

A equação geral que modela o problema de deformação de um corpo elástico utiliza-se de tensores para poder capturar o comportamento estrutural do objeto sem depender da escolha do sistema de coordenadas. Serão apresentados brevemente nesta seção os conceitos e definições mais importantes para uso posterior. A principal referência deste capítulo será o livro de introdução à mecânica do contínuo, do professor Liu (I-SHIH, 2019).

2.1 ÁLGEBRA E CÁLCULO TENSORIAL

Seja V um espaço vetorial de dimensão finita com $\dim(V) = n$, e admitindo $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ como base de V . Qualquer vetor $\mathbf{v} \in V$ pode ser representado como

$$\mathbf{v} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + \dots + v_n\mathbf{e}_n \quad (2.1)$$

Definição: Um produto interno é um mapa simétrico, positivo definido e bilinear $\cdot : V \times V \rightarrow F$, onde usaremos \mathbb{R} como corpo F .

Definição: A norma $|\cdot| : V \rightarrow F$ é um mapa tal que

$$|\mathbf{v}| = \sqrt{\mathbf{v} \cdot \mathbf{v}} \quad (2.2)$$

Pela desigualdade de Cauchy-Schwarz, $\mathbf{u} \cdot \mathbf{v} \leq |\mathbf{u}||\mathbf{v}|$, e com isso define-se um ângulo orientado entre dois vetores $\theta : V \times V \rightarrow [0, \pi]$ de forma que

$$\cos(\theta(\mathbf{u}, \mathbf{v})) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}||\mathbf{v}|} \quad (2.3)$$

\mathbf{u} e \mathbf{v} serão considerados ortogonais quando $\mathbf{u} \cdot \mathbf{v} = 0$

Definição: Uma base $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ é ortonormal se, e somente se

$$\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}, \quad \forall i, j \in [1, n] \quad (2.4)$$

onde δ é o delta de Kronecker, isto é:

$$\delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

Notação de Einstein: Tendo implícitos o espaço vetorial sendo trabalhado, sua base e sua dimensão, serão omitidos os símbolos de somatório sob estes objetos, deixando apenas os componentes, bases e seus respectivos índices explicitados. Desta forma, as equações

$$\mathbf{v} = \sum_{i=1}^n v_i \mathbf{e}_i ; \mathbf{u} = \sum_{j=1}^n u_j \mathbf{e}_j \quad (2.5)$$

serão expressas na forma mais compacta:

$$\mathbf{v} = v_i \mathbf{e}_i ; \mathbf{u} = u_j \mathbf{e}_j \quad (2.6)$$

As três regras principais para utilização desta notação são:

1. O mesmo índice não pode aparecer mais de duas vezes no produto de 2 ou mais vetores/tensores
2. Índices livres em cada termo da equação precisam estar de acordo uns com os outros.
3. Índices livres e índices *dummy* devem poder ser alterados sem que isso altere o significado da expressão (supondo que itens 1 e 2 não estão sendo violados)

Exemplificação do item 1: $a_i b_j c_i d_j$, $a_i b_i + c_i d_i$ e $A_{ij} b_i$ são expressões válidas, enquanto que $a_i b_i c_i$ e $A_{ii} b_i$ não são.

Exemplificação do item 2: $f_i a_i = g_i b_i$ e $A_{ii} b_j = F_{kk} c_j$ são expressões válidas, enquanto que $f_i a_i = g_j b_j$ e $A_{ii} b_j = F_{kk} c_l$ não são.

Exemplificação do item 3: $A_{ij} b_j$, trocando j por k , $A_{ik} b_k$ mantém a semântica da expressão, enquanto que $A_{ij} b_j$, trocando i por j , $A_{jj} b_j$ muda a semântica da expressão (e está sintaticamente inválida).

Definição: Uma transformação linear A é um mapa $A : V \rightarrow V$ se, e somente se, para $\mathbf{u}, \mathbf{v} \in V$ e $a \in \mathbb{R}$

$$A(a\mathbf{u} + \mathbf{v}) = aA(\mathbf{u}) + A(\mathbf{v}) \quad (2.7)$$

Feitas estas definições, pode-se definir, agora, o objeto central desta seção.

Definição: Seja $\mathcal{L}(V)$ o espaço de transformações lineares em V . Seus elementos são os tensores de ordem 2. A partir deste ponto, todo tensor mencionado será um tensor de ordem 2, salvo quando explicitamente indicado que não.

Escalares são equivalentes a tensores de ordem 0, vetores a tensores de ordem 1 e matrizes a tensores de ordem 2. Um tensor é um objeto que generaliza estruturas com índices associados a cada dimensão. Junto da notação de Einstein, podemos fazer operações entre tensores de ordem maior indicando quais índices estão sendo operados e em qual ordem (a variar da definição do operador).

Definição: O produto tensorial $\otimes : V \times V \rightarrow \mathcal{L}(V)$ é um mapa tal que, $\forall \mathbf{w} \in V$

$$\underbrace{(\mathbf{u} \otimes \mathbf{v})}_{\text{Tensor de ordem 2}} \cdot \mathbf{w} = \underbrace{(\mathbf{v} \cdot \mathbf{w})}_{\text{Escalar}} \mathbf{u} \quad (2.8)$$

Decomposição em componentes: Tendo $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ como base de V , então $\mathbf{e}_i \otimes \mathbf{e}_j, \forall i, j \in [1, n]$ é base de $\mathcal{L}(V)$ e $\forall A \in \mathcal{L}(V)$

$$A = A_{ij}(\mathbf{e}_i \otimes \mathbf{e}_j) = (A\mathbf{e}_j) \cdot \mathbf{e}_i \quad (2.9)$$

Produto tensor-vetor: Aqui pode-se observar vários resultados anteriores sendo utilizados para reduzir o problema de produto entre tensores para produto entre escalares. Esta estratégia será adotada recorrentemente.

$$\begin{aligned} A\mathbf{v} &= \\ (A_{ij}\mathbf{e}_i \otimes \mathbf{e}_j)(v_k\mathbf{e}_k) &= \\ A_{ij}v_k(\mathbf{e}_i \otimes \mathbf{e}_j)\mathbf{e}_k &= \\ A_{ij}v_k(\mathbf{e}_j \cdot \mathbf{e}_k)\mathbf{e}_i &= \\ A_{ij}v_k\delta_{jk}\mathbf{e}_i &= A_{ij}v_j\mathbf{e}_i \end{aligned} \quad (2.10)$$

Note que a expressão acima é equivalente ao produto matriz-vetor.

Produto tensor-tensor: Os passos são análogos ao processo acima, e portanto serão resumidos diretamente ao passo final, e omitindo a base:

$$(AB)_{ij} = A_{ik}B_{kj} \quad (2.11)$$

Novamente, a expressão é equivalente ao produto matriz-matriz. De fato, pode-se pensar o produto tensorial como produto externo entre vetores (vetor-coluna multiplicado por um vetor-linha).

Transposta de um tensor: O transposto A^T de A é um tensor tal que

$$A^T\mathbf{u} \cdot \mathbf{v} = \mathbf{u} \cdot A\mathbf{v} \quad (2.12)$$

Operando com apenas um objeto, em termos de componente:

$$(A^T)_{ij} = A_{ji} \quad (2.13)$$

Tensor ortogonal: Dizemos que um tensor Q é um tensor ortogonal se e somente se

$$Q\mathbf{u} \cdot Q\mathbf{v} = \mathbf{u} \cdot \mathbf{v} \quad (2.14)$$

E da definição anterior de transposta, segue que

$$Q^T Q = I \quad (2.15)$$

Definição: O traço $tr : \mathcal{L}(V) \rightarrow \mathbb{R}$ de um tensor é um mapa tal que

$$tr(\mathbf{u} \otimes \mathbf{v}) = \mathbf{u} \cdot \mathbf{v} \quad (2.16)$$

Com o tensor A em mãos, pode-se calcular diretamente o traço sem precisarmos calcular sua base:

$$\begin{aligned}
 tr(A) &= \\
 tr(A_{ij}\mathbf{e}_i \otimes \mathbf{e}_j) &= \\
 A_{ij} tr(\mathbf{e}_i \otimes \mathbf{e}_j) &= \\
 A_{ij}(\mathbf{e}_i \cdot \mathbf{e}_j) &= \\
 A_{ij}\delta_{ij} &= A_{ii}
 \end{aligned} \tag{2.17}$$

Disso seguem dois corolários:

$$tr(A) = tr(A^T) \tag{2.18}$$

$$tr(AB) = tr(BA) \tag{2.19}$$

Produto interno entre tensores O produto interno $: : \mathcal{L}(V) \times \mathcal{L}(V) \rightarrow \mathbb{R}$ entre dois tensores A e B é definido por

$$A : B = tr(AB^T) = A_{ij}B_{ij} \tag{2.20}$$

Definição: Se $dim(V) = 3$, então define-se o *produto vetorial* $\times : V \times V \rightarrow V$ da seguinte forma:

$$\mathbf{u} \times \mathbf{v} = \mathcal{E}_{ijk}\mathbf{u}_j\mathbf{v}_k \tag{2.21}$$

onde \mathcal{E} é o símbolo de Levi-Civita de dimensão 3, cujo comportamento é:

$$\mathcal{E}_{ijk} = \begin{cases} 1 & \text{se } \{i,j,k\} \text{ é uma permutação par de } \{1, 2, 3\} \\ -1 & \text{se } \{i,j,k\} \text{ é uma permutação ímpar de } \{1, 2, 3\} \\ 0 & \text{caso contrário} \end{cases}$$

Definição: O determinante $det : \mathcal{L}(V) \rightarrow \mathbb{R}$ de um tensor A é um mapa tal que para todo $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$:

$$det(A) = \frac{Au \cdot (Av \times Aw)}{u \cdot (v \times w)} \tag{2.22}$$

Por fim, para a parte de cálculo tensorial, basta a definição de gradiente e de divergente.

Definição: Para $f : \mathbb{R}^n \rightarrow \mathbb{R}$, o gradiente ∇ de f é tal que $\forall \mathbf{v} \in V$:

$$\nabla f[\mathbf{v}] = \underbrace{\nabla f}_{\text{vetor}} \cdot \mathbf{v} \tag{2.23}$$

E a operação à direita da igualdade é um produto interno entre vetores. Em notação indicial:

$$(\nabla f)_i = \frac{\partial f}{\partial x_i} = f_{,i} \quad (2.24)$$

Definição: Para $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, o gradiente ∇ de f é tal que $\forall \mathbf{v} \in V$:

$$\nabla f[\mathbf{v}] = \underbrace{\nabla f}_{\text{tensor}} \mathbf{v} \quad (2.25)$$

E a operação à direita da igualdade é um produto tensor-vetor, como definido em 2.10. Em notação indicial:

$$(\nabla f)_{ij} = \frac{\partial f_i}{\partial x_j} = f_{i,j} \quad (2.26)$$

Definição: Seja a função $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ou $f : \mathbb{R}^n \rightarrow \mathbb{R}$, o divergente $div : \mathcal{L}(V) \rightarrow \mathbb{R}$ é uma mapa tal que

$$div(v) = tr(\nabla v) \quad (2.27)$$

E aplica-se uma das definições anteriores de gradiente, a depender do tipo de função vetorial que é v . Para um tensor A temos que

$$div(A) = (div A)_i \mathbf{e}_i = \frac{\partial A_{,j}}{\partial x_j} \mathbf{e}_i = A_{ij,j} \mathbf{e}_i \quad (2.28)$$

Propriedade do divergente: Uma propriedade futuramente útil, e com que será encerrada esta sessão é a seguinte

$$div(Av) = A : \nabla v + div(A) \cdot v \quad (2.29)$$

2.2 CINEMÁTICA DE DEFORMAÇÕES FINITAS

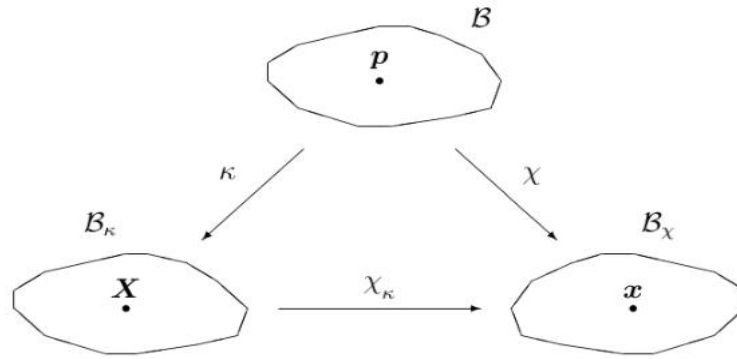
Nesta seção serão feitas as definições das principais estruturas e mostrados os resultados essenciais para as manipulações algébricas do próximo capítulo. Serão utilizados os resultados da seção anterior.

Um corpo \mathcal{B} pode ser definido como uma região do espaço euclidiano tridimensional \mathbb{E} . Chama-se este corpo posicionado no espaço de *configuração* do corpo. Usualmente, define-se uma configuração de referência $\kappa : \mathcal{B} \rightarrow \mathbb{E}$, e um ponto \mathbf{p} abstrato no corpo \mathcal{B} torna-se um ponto \mathbf{X} posicionado no espaço, i.e. $\kappa(\mathbf{p}) = \mathbf{X}$. O corpo posicionado no espaço será denotado por \mathcal{B}_κ .

Seja χ uma configuração arbitrária do corpo \mathcal{B} . Chamaremos de *deformação* o mapa χ_κ tal que

$$\chi_\kappa = \chi \circ \kappa^{-1} \quad (2.30)$$

Figura 1 – Diagrama de deformação



Fonte: I-SHI (2019, p. 11)

De modo que um ponto qualquer \mathbf{x} do corpo deformado é obtido via esse mapa aplicado no respectivo ponto na configuração de referência:

$$\mathbf{x} = \chi(\kappa^{-1}(\mathbf{X})) \quad (2.31)$$

O diagrama da Figura 1 reúne todas estas definições de forma sucinta. Disto, defini-se uma primeira estrutura importantíssima para a simulação numérica da deformação de um corpo: o *gradiente de deformação* de χ relativo a κ , denotado por F_κ é definido por

$$F_\kappa = \nabla_{\mathbf{X}} \chi_\kappa \quad (2.32)$$

e é uma aproximação linear da deformação. Uma representação matricial possível para este gradiente, fazendo escolhas usuais como base canônica e sistema de coordenadas alinhado aos vetores base é

$$[F] = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix} \quad (2.33)$$

E com F não-singular, podemos fazer sua decomposição polar (para tensores de ordem 2, análogo ao caso matricial)

$$F = RU = VR \quad (2.34)$$

Com U, V e R únicas. U e V são simétricas e positiva-definidas, e R é ortogonal. Cada um destes tensores possui um nome associado. U é chamado tensor de esticamento à direita, V é chamado tensor de esticamento à esquerda e R é o tensor de rotação.

A partir dessa decomposição, definiremos 2 outros tensores:

$$C = F^T F = (RU)^T RU = U^T R^T RU = U^T U = UU = U^2 \quad (2.35)$$

chamado de Tensor Cauchy-Green de deformação à direita, e

$$B = FF^T = VR(VR)^T = VRR^T V^T = V^T V = VV = V^2 \quad (2.36)$$

chamado de Tensor Cauchy-Green de deformação à esquerda. Para as duas derivações acima, foram usadas as seguintes propriedades, nesta ordem: Igualdade de F por uma das decomposições em 2.34; aplicação da transposta sob produto; fato de que R é ortogonal e portanto $R^T R = RR^T = I$; fato de que U e V são simétricas, logo $U^T = U$ e $V^T = V$; e por fim apenas uma mudança de representação na notação.

Porém, geralmente não é conhecida a função de deformação explicitamente para que calcule-se F diretamente. Portanto, define-se o vetor de deslocamento

$$\mathbf{u} = \chi_\kappa(\mathbf{X}) - \mathbf{X} \quad (2.37)$$

de forma que pode-se calcular o *gradiente de deslocamento* $H = \nabla_{\mathbf{X}} \mathbf{u}$, com representação matricial

$$[H] = \begin{bmatrix} \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_2} & \frac{\partial u_1}{\partial X_3} \\ \frac{\partial u_2}{\partial X_1} & \frac{\partial u_2}{\partial X_2} & \frac{\partial u_2}{\partial X_3} \\ \frac{\partial u_3}{\partial X_1} & \frac{\partial u_3}{\partial X_2} & \frac{\partial u_3}{\partial X_3} \end{bmatrix} \quad (2.38)$$

Assim, de 2.37, aplica-se o operador $\nabla_{\mathbf{X}}$ dos dois lados e desenvolve-se:

$$\nabla_{\mathbf{X}} \mathbf{u} = \nabla_{\mathbf{X}} \chi_\kappa(\mathbf{X}) - \nabla_{\mathbf{X}} \mathbf{X} \quad (2.39)$$

Usando 2.32 na primeira parcela do lado direito da equação, obtém-se

$$H = F - I \iff F = I + H \quad (2.40)$$

Com I sendo o tensor identidade. Como H é calculável a partir de duas configurações conhecidas, que tem-se acesso, agora é possível calcular F via H .

3 O PROBLEMA DE DEFORMAÇÃO

Feitas as considerações iniciais, tem-se o necessário para começar a analisar o problema de deformação elástica linear bidimensional. A equação que governa o problema particular deste trabalho, onde também supõe-se que o corpo é composto por um material Mooney-Rivlin compressível ou quase incompressível, é derivada através de inúmeras técnicas de linearização e aplicação dos conceitos abordados no capítulo anterior. Como este processo é um pouco longo e foge ao escopo deste trabalho, partiremos do problema de valor de contorno resultante deste processo. Esta derivação pode ser encontrada em (PEREIRA, 2015).

3.1 O PROBLEMA DE VALOR DE CONTORNO - FORMULAÇÃO FORTE

Seja T_κ o tensor que indica o estado de tensão em cada ponto de um corpo (chamado tensor de Piola-Kirchhoff). Consideremos que a região do espaço que o corpo ocupa possa ser subdividida em três conjuntos: Ω , definida por todos os pontos internos da região, Γ_1 definida por todos os pontos da fronteira da região onde é conhecida a tração e Γ_2 a região definida por todos os pontos da fronteira da região onde é conhecida a posição (em coordenadas) destes pontos, deseja-se encontrar uma função $u : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ tal que:

$$\begin{cases} -div(T_\kappa) = 0 \text{ em } \Omega \\ T_\kappa \mathbf{n}_\kappa = f \text{ em } \Gamma_1 \\ u(\mathbf{x}, \tau) = g \text{ em } \Gamma_2 \end{cases} \quad (3.1)$$

onde \mathbf{n}_κ é o vetor normal à superfície, $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ é a função que, dado um ponto do corpo retorna a tensão associada ao ponto em cada uma das direções (define a condição de fluxo, ou condição de Neumann), u é a função deslocamento, que dado um ponto do espaço e uma carga de tensão retorna a nova posição do ponto, e $g : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ é a função que, dado um ponto do espaço, retorna o novo ponto pós-deformação (define a condição prescrita, ou condição de Dirichlet).

Para resolver este problema de valor de contorno, será utilizada a técnica do Método dos Elementos Finitos. Esta técnica consiste de três etapas principais, com 2 técnicas para transicionar entre elas: A primeira etapa é chamada *formulação forte* do problema, onde escrevemos o PVC tal qual foi feito acima. A segunda etapa é chamada *formulação fraca*, onde o problema original é enfraquecido ao introduzir-se uma função auxiliar e realizar uma integração para que o problema torne-se mais fácil de ser resolvido. A terceira etapa é chamada *formulação matricial*, onde o problema de dimensão infinita é aproximado por um problema de dimensão finita, permitindo que um computador seja capaz de resolvê-lo

diretamente. Nesta etapa aplica-se o método de Galerkin para discretização do problema. Para mais detalhes sobre o método, ver (RINCON; I-SHIH, 2020).

3.2 FORMULAÇÃO FRACA

Introduz-se a função auxiliar $v : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ e faz-se o produto com a primeira igualdade de 3.1, nossa equação principal:

$$-div(T_\kappa) \cdot v = 0 \quad (3.2)$$

Integrando os 2 lados da equação:

$$\int_{\Omega} -div(T_\kappa) \cdot v \, d\Omega = 0 \quad (3.3)$$

Usando a propriedade 2.29 e a propriedade de soma para integrais:

$$\int_{\Omega} T_\kappa : \nabla v \, d\Omega - \int_{\Omega} div(T_\kappa v) \, d\Omega = 0 \quad (3.4)$$

Aplicando o Teorema da Divergência na segunda parcela

$$\int_{\Omega} T_\kappa : \nabla v \, d\Omega - \int_{\Gamma_1} v \cdot T_\kappa \mathbf{n}_\kappa \, d\Gamma = 0 \quad (3.5)$$

Usando a definição de produto interno entre tensores 2.20 na primeira parcela

$$\int_{\Omega} tr(T_\kappa \nabla^T v) \, d\Omega - \int_{\Gamma_1} v \cdot T_\kappa \mathbf{n}_\kappa \, d\Gamma = 0 \quad (3.6)$$

Usando a primeira condição de contorno de 3.1 na segunda parcela:

$$\int_{\Omega} tr(T_\kappa \nabla^T v) \, d\Omega - \int_{\Gamma_1} v \cdot f \, d\Gamma = 0 \quad (3.7)$$

Reordenando

$$\int_{\Omega} tr(T_\kappa \nabla^T v) \, d\Omega = \int_{\Gamma_1} v \cdot f \, d\Gamma \quad (3.8)$$

Neste passo, será feito a troca do tensor T_κ pela soma dos tensores T e $\mathcal{L}[H]$. A justificativa desta troca pode ser encontrada em (PEREIRA, 2015), onde as equações principais são: (3.23), (3.20) e (3.17). Além disso, na sessão *Cinemática de Deformações Finitas* do capítulo anterior, as duas estruturas recorrentemente utilizadas nas equações citadas foram apresentadas: os tensores F (equação 2.32), B (equação 2.36) e H (equação 2.38)

Fazendo a troca $T_\kappa = T + \mathcal{L}[H]$

$$\int_{\Omega} tr((T + \mathcal{L}[H]) \nabla^T v) \, d\Omega = \int_{\Gamma_1} v \cdot f \, d\Gamma \quad (3.9)$$

Para evitar alongar muito este trecho, serão aplicadas, de uma só vez e nesta ordem, as seguintes propriedades: distributiva entre a soma de tensores e o gradiente, $tr(A + B) = tr(A) + tr(B)$, propriedade da soma de integrais e uma manipulação algébrica simples para subtrair uma integral dos dois lados. Assim, chega-se em

$$\int_{\Omega} \mathcal{L}[H] \nabla^T v \, d\Omega = \int_{\Gamma_1} v \cdot f \, d\Gamma - \int_{\Omega} tr(T \nabla^T v) \, d\Omega \quad (3.10)$$

E tem-se uma possível formulação fraca do problema. Esta, porém, possui integrais operando sobre termos vetoriais e tensoriais. Para que seja possível reutilizar as ferramentas já conhecidas do cálculo integral e diferencial, esta equação será escrita na sua forma em componentes.

Começando pela integral à esquerda da igualdade, usa-se a igualdade $\mathcal{L}[H] = tr(H)T - TH^T + L[H]$, encontrada na equação (3.24) do trabalho (PEREIRA, 2015):

$$\int_{\Omega} \mathcal{L}[H] \nabla^T v \, d\Omega = \int_{\Omega} (tr(H)T - TH^T + L[H]) \nabla^T v \, d\Omega \quad (3.11)$$

Usando a mesma sequência de propriedades listadas entre as equações 3.9 e 3.10, exceto a última:

$$\int_{\Omega} \mathcal{L}[H] \nabla^T v \, d\Omega = \int_{\Omega} tr(tr(H)T \nabla^T v) \, d\Omega - \int_{\Omega} tr(TH^T \nabla^T v) \, d\Omega + \int_{\Omega} tr(L[H] \nabla^T v) \, d\Omega \quad (3.12)$$

O próximo passo consiste em representar o termo $L[H]$ da última parcela em termos de componentes. Novamente, será apresentado apenas o resultado final para poupar as longas contas com índices. O processo utiliza a equação (3.22) de (PEREIRA, 2015) e todas as propriedades de álgebra de tensores expostas no capítulo anterior. Ao final, tem-se um tensor de quarta ordem:

$$L_{ijkl} = \beta \delta_{kl} \delta_{ij} + s_1 (\delta_{ik} B_{lj} + B_{il} \delta_{lj}) - s_2 (B_{ik}^{-1} \delta_{lj} + \delta_{li} B_{kj}^{-1}) \quad (3.13)$$

Todos os outros termos das integrais das equações 3.10 e 3.12 são mais facilmente expressos em termos de componentes, utilizando as propriedades e definições do capítulo anterior. Será dada uma atenção especial à primeira parcela do lado direito de 3.12, para desmembrar o traço da seguinte forma:

$$\int_{\Omega} tr(tr(H)T \nabla^T v) \, d\Omega = \int_{\Omega} tr(H) tr(T \nabla^T v) \, d\Omega = \int_{\Omega} H_{kk} tr(T \nabla^T v) \, d\Omega \quad (3.14)$$

E juntando, por fim, as equações 3.10, 3.12 e 3.14 e escrevendo em componentes, chega-se em:

$$\int_{\Omega} \frac{\partial u_k}{\partial x_k} T_{ij} \frac{\partial v_i}{\partial x_j} d\Omega - \int_{\Omega} T_{ik} \frac{\partial u_j}{\partial x_k} \frac{\partial v_i}{\partial x_j} d\Omega + \int_{\Omega} L_{ijkl} \frac{\partial u_k}{\partial x_l} \frac{\partial v_i}{\partial x_j} d\Omega = \int_{\Gamma_{1i}} f_i v_i d\Gamma - \int_{\Omega} T_{ij} \frac{\partial v_i}{\partial x_j} d\Omega \quad (3.15)$$

Para facilitar manipulações posteriores, iremos definir dois operadores $a(u, v)$, bilinear, e $b(v)$, linear, tal que o primeiro operador resulta no lado esquerdo da equação e o segundo operador resultado no lado direito da equação (RINCON; I-SHIH, 2020). Assim, podemos reescrever 3.15 como a formulação fraca final:

$$a(u, v) = b(v) \quad (3.16)$$

3.3 FORMULAÇÃO MATRICIAL

Até este momento, trata-se do problema contínuo e de dimensão infinita. Porém, os computadores são máquinas finitas, que não trabalham facilmente com objetos infinitamente suaves. Sendo assim, será feita uma aproximação do problema até o momento para um problema discreto, usando o método de Galerkin. Para isso, se antes eram usadas as funções $u \in \mathcal{H}$ e $v \in \mathcal{V}$, sendo \mathcal{H} e \mathcal{V} espaços de dimensão infinita, agora serão utilizadas as funções $u^h \in \mathcal{H}^h$ e $v^h \in \mathcal{V}^h$ em subespaços de dimensão finita. A formulação do problema, portanto, passa a ser

$$a(u^h, v^h) = b(v^h) \quad (3.17)$$

Será assumido que u^h admite uma decomposição em outras duas funções:

$$u^h = w^h + g^h \quad (3.18)$$

$$\text{onde } g_i^h(x) = \begin{cases} g_i(x), \forall x \in \Gamma_{2i} \\ 0, \forall x \notin \Gamma_{2i} \end{cases} \quad (3.19)$$

De fato, ao final, é necessário apenas que a função seja zero sob os pontos internos da malha que discretizará o espaço. O índice i presente ao lado da identificação da partição de Γ expressa que apenas a direção associada a componente da função deve ser avaliada. Substituindo 3.18 em 3.17:

$$\begin{aligned} a(u^h, v^h) &= b(v^h) \\ a(w^h + g^h, v^h) &= b(v^h) \\ a(w^h, v^h) + a(g^h, v^h) &= b(v^h) \\ a(w^h, v^h) &= b(v^h) - a(g^h, v^h) \quad \forall v^h \in \mathcal{V}^h \end{aligned} \quad (3.20)$$

O objetivo desta manipulação é separar as funções com valores conhecidos (f , g e v) da função u que deseja-se encontrar ao resolver o PVC. Como o operador $b()$ tem em

sua definição a função f , pode-se aproximá-la por f^h com definição análoga a 3.19, onde vale a mesma observação do parágrafo abaixo. E como mencionado nesta observação, o domínio do corpo a ser deformado será discretizado em uma malha de elementos finitos.

Originalmente, u foi definida mapeando $\mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^2$. Porém, se fixarmos a segunda entrada (τ), podemos considerar que u é apenas função de $\mathbb{R}^2 \rightarrow \mathbb{R}^2$, pensada em termos meramente espaciais.

Sejam $\varphi_1, \varphi_2, \dots, \varphi_N$ funções de interpolação da base de \mathcal{V}^h . Como $w^h \in \mathcal{V}^h$, escreve-se w^h como combinação lineares de φ :

$$w^h(x) = \sum_{A=1}^N d_A \varphi_A(x), \quad d_A \in \mathbb{R}^2 \quad (3.21)$$

Em componentes

$$w_j^h(x) = \sum_{B=1}^N d_{jB} \varphi_B(x), \quad j = 1, 2 \quad (3.22)$$

O número 2, ao final destas duas definições, se refere a quantidade de graus de liberdade espacial que a função pode assumir (no caso espacial bidimensional, 2 direções).

Como g^h é conhecida, calculamos w^h apenas nos nós onde não conhecemos g^h (com as direções associadas, já que podemos ter apenas 1 coordenada prescrita - como no caso do deslocamento de um trem sob trilhos). Se N é o conjunto total dos nós da malha, denotemos por N_2 os nós em Γ_2 . E reescreve-se 3.21 como

$$w_j^h(x) = \sum_{B \in N - N_{2j}} d_{jB} \varphi_B(x), \quad j = 1, 2 \quad (3.23)$$

Com um pouco de abuso de notação, já que N e N_{2j} são conjuntos e a diferença normalmente é denotada com \setminus . Repetindo o processo para g^h :

$$g_j^h(x) = \sum_{B \in N_{2j}} g_{jB} \varphi_B(x), \quad j = 1, 2 \quad (3.24)$$

onde $g_{jB} = g_j(x_B)$. Considerando a base canônica do \mathbb{R}^2 , podemos escrever w^h e g^h na forma

$$w^h = w_j^h e_j \quad \text{e} \quad g^h = g_j^h e_j \quad (3.25)$$

Substituindo em 3.23:

$$w^h(x) = \sum_{B \in N - N_{2j}} d_{jB} \varphi_B(x) e_j \quad (3.26)$$

E substituindo esta expressão no lado esquerdo de 3.20:

$$a(w^h, v^h) = a\left(\sum_{B \in N-N_{2j}} d_{jB} \varphi_B(x) e_j, v^h\right) = \sum_{B \in N-N_{2j}} a(d_{jB} \varphi_B(x) e_j, v^h) \quad (3.27)$$

Explicitando o somatório em j e movendo as constantes d_{jB} para fora do operador:

$$a(w^h, v^h) = \sum_{j=1}^2 \sum_{B \in N-N_{2j}} a(\varphi_B(x) e_j, v^h) d_{jB}, \forall v^h \in \mathcal{V}^h \quad (3.28)$$

Como a equação acima é válida para todo $v^h = v_i^h e_i$, escolhe-se $v^h = \varphi_A e_i$ de forma a obter m equações:

$$a(w^h, v^h) = \sum_{j=1}^2 \sum_{B \in N-N_{2j}} a(\varphi_B e_j, \varphi_A e_i) d_{jB}, 1 \leq i \leq m \quad (3.29)$$

finalizando, assim, o lado esquerdo da equação 3.20. O processo é análogo para o lado direito, resultando em

$$b(\varphi_A e_i) - \sum_{j=1}^2 \sum_{B \in N-N_{2j}} a(\varphi_B e_j, \varphi_A e_i) g_{jB}, 1 \leq i \leq m \quad (3.30)$$

E igualando 3.29 e 3.30, agrupa-se e renome-se os objetos da equação de forma a obter o sistema linear

$$K_{AB} d_{jB} = F_A \quad (3.31)$$

Para montar este sistema linear computacionalmente, são necessários mais dois passos: fazer as substituições apropriadas em 3.15, agora com as funções interpoladas, e sair da perspectiva de montagem global das estruturas para a perspectiva de montagem dos elementos finitos (e nestes, fazer a montagem a partir do elemento finito padrão). Estes passos são feitos na seção a seguir.

3.4 MONTAGEM DO SISTEMA LINEAR

Em 3.17, foi feita a definição dos operadores $a(\cdot)$ e $b(\cdot)$ sob entradas genéricas, que nomeamos convenientemente de u^h e v^h . Tendo que estas funções são passíveis de decomposição em produtos de φ 's e dos vetores canônicos, reescreve-se estas entradas como:

$$\begin{aligned} u_r^h &= \langle u^h, e_r \rangle = \langle \varphi_B e_k, e_r \rangle = \varphi_B \delta_{kr} \\ v_i^h &= \langle v^h, e_i \rangle = \langle \varphi_A e_s, e_i \rangle = \varphi_A \delta_{si} \end{aligned} \quad (3.32)$$

E substitui-se estas interpolações em 3.15. Começando pelo lado esquerdo:

$$\int_{\Omega} \frac{\partial \varphi_B \delta_{kr}}{\partial x_k} T_{ij} \frac{\partial \varphi_A \delta_{si}}{\partial x_j} d\Omega - \int_{\Omega} T_{ik} \frac{\partial \varphi_B \delta_{kr}}{\partial x_k} \frac{\partial \varphi_A \delta_{si}}{\partial x_j} d\Omega + \int_{\Omega} L_{ijkl} \frac{\partial \varphi_B \delta_{kr}}{\partial x_l} \frac{\partial \varphi_A \delta_{si}}{\partial x_j} d\Omega \quad (3.33)$$

Será exibido o processo para uma integral, e o processo será análogo para as outras duas:

$$\begin{aligned}
\int_{\Omega} \frac{\partial \varphi_B \delta_{kr}}{\partial x_k} T_{ij} \frac{\partial \varphi_A \delta_{si}}{\partial x_j} d\Omega &= \text{colapsando delta, mantendo o índice r} \\
\int_{\Omega} \frac{\partial \varphi_B}{\partial x_r} T_{ij} \frac{\partial \varphi_A \delta_{si}}{\partial x_j} d\Omega &= \text{colapsando delta, mantendo o índice s} \\
\int_{\Omega} \frac{\partial \varphi_B}{\partial x_r} T_{sj} \frac{\partial \varphi_A}{\partial x_j} d\Omega &= \text{renomeando r para s, s para r, j para i} \\
\int_{\Omega} \frac{\partial \varphi_B}{\partial x_s} T_{ri} \frac{\partial \varphi_A}{\partial x_i} d\Omega &
\end{aligned} \tag{3.34}$$

E ao final do processo para as três parcelas, tem-se a simplificação de 3.33 para:

$$\int_{\Omega} T_{ri} \frac{\partial \varphi_B}{\partial x_s} \frac{\partial \varphi_A}{\partial x_i} d\Omega - \int_{\Omega} T_{ri} \frac{\partial \varphi_B}{\partial x_i} \frac{\partial \varphi_A}{\partial x_s} d\Omega + \int_{\Omega} L_{ijkl} \frac{\partial \varphi_B}{\partial x_j} \frac{\partial \varphi_A}{\partial x_i} d\Omega \tag{3.35}$$

Para o lado direito de 3.15, inicia-se considerando uma interpolação de f^h da mesma forma que foi feita para g^h , considerando os nós apropriados:

$$f_j^h(x) = \sum_{B \in N_{3j}} \varphi_B(x) f_{jB}; \quad j = 1, 2 \tag{3.36}$$

E para a primeira integral:

$$\int_{\Gamma_{1j}} f_i v_i d\Gamma = \int_{\Gamma_{1j}} \left(\sum_{B \in N_{3j}} \varphi_B(x) f_{jB} \right) v_i d\Gamma = \sum_{B \in N_{3j}} \int_{\Gamma_{1j}} \varphi_B(x) f_{jB} \varphi_A(x) d\Gamma \tag{3.37}$$

A segunda integral segue o processo análogo aos dois últimos passos de 3.34. Assim, está totalmente determinado a decomposição da expressão dos operadores $a(\cdot)$ e $b(\cdot)$ em forma de componentes. Ou seja, é possível reescrever o sistema linear 3.31 em forma de componentes.

Seguindo, define-se a partição do domínio Ω em N_e elementos finitos

$$\Omega = \bigcup_{e=1}^{N_e} \bar{\Omega}_e \tag{3.38}$$

E a montagem da matriz e do vetor do sistema linear serão acumulados na iteração em cada elemento finito

$$K = \sum_{e=1}^{N_e} K^e; \quad F = \sum_{e=1}^{N_e} F^e; \tag{3.39}$$

e para definir cada estrutura local, escolhe-se uma função de interpolação local φ_a^e tal que

$$\varphi_a^e(x_b) = \begin{cases} 1 & \text{se } a = b \\ 0 & \text{se } a \neq b \end{cases} \quad (3.40)$$

E, assim, defini-se as estruturas locais

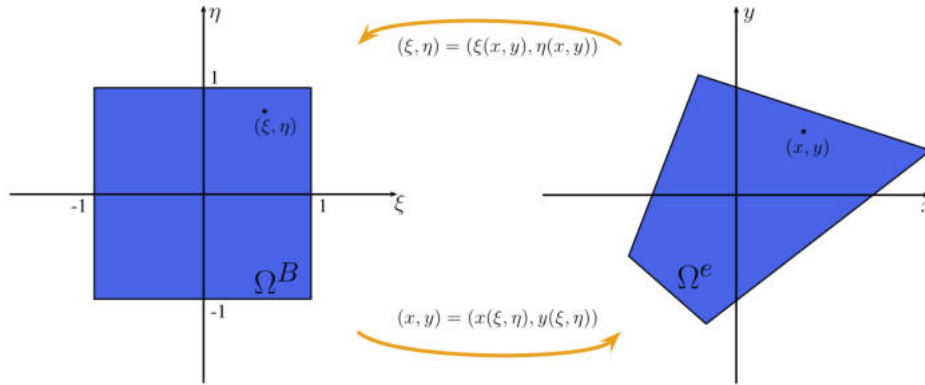
$$K_{ab}^e = \int_{\Omega^e} T_{ri} \frac{\partial \varphi_b^e}{\partial x_s} \frac{\partial \varphi_b^e}{\partial x_i} d\Omega^e - \int_{\Omega^e} T_{ri} \frac{\partial \varphi_b^e}{\partial x_i} \frac{\partial \varphi_a^e}{\partial x_s} d\Omega^e + \int_{\Omega^e} L_{ijkl} \frac{\partial \varphi_b^e}{\partial x_j} \frac{\partial \varphi_a^e}{\partial x_i} d\Omega^e \quad (3.41)$$

$$F_a^e = \sum_{b \in N_{3j}} \int_{\Gamma_{1j}^e} \varphi_b^e(x) f_{jb} \varphi_a^e(x) d\Gamma^e - \int_{\Omega^e} T_{ri} \frac{\partial \varphi_a^e}{\partial x_i} d\Omega^e - \sum_{b \in N_{2j}} K_{ab}^e g_{jb} \quad (3.42)$$

Esta última parcela surge ao aplicar a decomposição de $a(\cdot)$ em componentes na última parcela de 3.20, que foi apenas indicada acima, e percebendo que, quando comparada com o lado esquerdo da equação, a única diferença é do vetor d_{jB} para o vetor g_{jB} . Desta forma, pode-se reaproveitar as contas do cálculo de K^e para o cálculo de F^e .

Por fim, resta trocar os domínios de integração para que seja feita no elemento padrão, e então remapeada para os elementos finitos respectivos, em vez de ser feita no domínio dos próprios elementos. Como o método de integração numérica a ser utilizado para avaliar as integrais será a quadratura gaussiana, o elemento finito padrão é convenientemente tomado como sendo o quadrado biunitário centrado em $(0, 0)$.

Figura 2 – Mapeamento entre elemento finito qualquer e elemento padrão



Fonte: Pereira, Gabriel Thomaz (2015, p. 45)

Como os elementos são convexos, basta mapear os vértices correspondentes entre os elementos, que todo outro ponto será combinação convexa deles. Como o elemento genérico Ω^e não está, necessariamente, alinhado e paralelo aos eixos do sistema de coordenadas, cada reta de fronteira pode depender de ambas as variáveis. Pretende-se encontrar, portanto, um mapeamento

$$\begin{aligned} \Omega^B &\rightarrow \Omega^e \\ (\xi_1, \xi_2) &\rightarrow (x_1(\xi_1, \xi_2), x_2(\xi_1, \xi_2)) \end{aligned} \quad (3.43)$$

onde Ω^B é o domínio do quadrado biunitário. Para auxiliar neste processo, define-se as funções de interpolação local para o elemento finito padrão:

$$\phi_a(\xi_b) = \begin{cases} 1 & \text{se } a = b \\ 0 & \text{se } a \neq b \end{cases} \quad (3.44)$$

Assim, pode-se construir $x_1(\xi_1, \xi_2)$ e $x_2(\xi_1, \xi_2)$ ao interpolar as coordenadas dos pontos do elemento genérico pelas funções ϕ do elemento padrão. A mudança de coordenadas termina sendo

$$\begin{aligned} x_1(\xi_1, \xi_2) &= x_{a1}^e \phi_a(\xi_1, \xi_2) \\ x_2(\xi_1, \xi_2) &= x_{a2}^e \phi_a(\xi_1, \xi_2) \end{aligned} \quad (3.45)$$

Com $a = 1, 2, 3, 4$ e sendo x_{aj}^e a coordenada do a -ésimo ponto de x^e (considerando numeração local) na direção j . Para realizar a troca de domínio de integração em 3.41 e 3.42, também é necessário o jacobiano da transformação.

$$J(x_1, x_2) = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix} \quad (3.46)$$

Substituindo 3.45 na expressão do jacobiano e colocando os escalares fora das derivadas

$$J(x_1(\xi_1, \xi_2), x_2(\xi_1, \xi_2)) = \begin{bmatrix} x_{a1}^e \frac{\partial \phi_a}{\partial \xi_1}(\xi_1, \xi_2) & x_{a1}^e \frac{\partial \phi_a}{\partial \xi_2}(\xi_1, \xi_2) \\ x_{a2}^e \frac{\partial \phi_a}{\partial \xi_1}(\xi_1, \xi_2) & x_{a2}^e \frac{\partial \phi_a}{\partial \xi_2}(\xi_1, \xi_2) \end{bmatrix} \quad (3.47)$$

E resta apenas um detalhe a ser endereçado antes de serem feitas as substituições nas integrais. As funções $\varphi(x)$ e $\phi(\xi)$ são construídas de forma que

$$\varphi(x(\xi)) = \phi(\xi) \quad (3.48)$$

$$\phi(\xi(x)) = \varphi(x) \quad (3.49)$$

Em 3.41 tem-se derivações parciais sendo feitas em $\varphi(x)$ que serão transformadas em derivações parciais feitas em $\phi(\xi)$. Aplicando a regra da cadeia ao derivar ϕ por x em 3.49

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial \xi} \frac{\partial \xi}{\partial x} \quad (3.50)$$

E agora é possível transformar uma derivada de ϕ em x para uma derivada de ϕ em ξ . Sabe-se que a função $\partial \xi / \partial x$ existe, mas não a conhecemos explicitamente (dada a forma como foi construído o mapeamento 3.43). Porém, tendo posse de $J(x_1, x_2)$ (que por sua

vez é calculado via ξ , e por isso é conhecido) realizando a transformação de $x \rightarrow \xi$, ao invertê-lo (e ele é inversível) temos a transformação de $\xi \rightarrow x$

$$J^{-1}(\xi_1, \xi_2) = \frac{1}{\det(J(x_1, x_2))} J^{-1}(x_1, x_2) \quad (3.51)$$

Substituindo na parcela a direita de 3.50

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial \xi} J^{-1}(\xi_1, \xi_2) = \frac{\partial \phi}{\partial \xi} \frac{\partial \xi}{\partial x} \quad (3.52)$$

E tem-se tudo que é necessário para trocar as integrações em 3.41 e 3.42 para o elemento finito padrão. O processo final é:

$$\frac{\partial \varphi}{\partial x}(x) = \frac{\partial \varphi}{\partial x}(x(\xi)) = \frac{\partial \phi}{\partial x}(\xi) = \frac{\partial \phi}{\partial \xi} \left(\frac{\partial \xi}{\partial x}(x) \right) \quad (3.53)$$

Como as expressões ficam muito grandes, as integrais serão exibidas uma por linha, e omitindo as variáveis das funções.

$$\begin{aligned} K_{ab}^e = & \\ & \int_{\Omega^B} T_{ri} \frac{\partial \phi_a^e}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \phi_b^e}{\partial \xi_l} \frac{\partial \xi_l}{\partial x_s} \det(J) d\xi_1 d\xi_2 \\ & - \int_{\Omega^B} T_{ri} \frac{\partial \phi_a^e}{\partial \xi_l} \frac{\partial \xi_l}{\partial x_s} \frac{\partial \phi_b^e}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \det(J) d\xi_1 d\xi_2 \\ & + \int_{\Omega^B} L_{risj} \frac{\partial \phi_a^e}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \phi_b^e}{\partial \xi_l} \frac{\partial \xi_l}{\partial x_j} \det(J) d\xi_1 d\xi_2 \end{aligned} \quad (3.54)$$

$$\begin{aligned} F_a^e = & \\ & \sum_{b=1}^4 \int_{\Gamma_{1j}^B} \phi_a^e \phi_b^e f_{jb} \det(J) d\xi_1 d\xi_2 \\ & - \int_{\Omega^B} T_{ri} \frac{\partial \phi_a^e}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_i} \det(J) d\xi_1 d\xi_2 \\ & - \sum_{b=1}^4 K_{ab}^e g_{jb} \end{aligned} \quad (3.55)$$

Ao resolver o sistema linear, chega-se ao vetor solução d_{jB} constituído dos coeficientes que, ao serem multiplicados pelas funções de interpolação reconstróem a aproximação w^h (vide 3.22) que, somada à g^h reconstrói a u^h de interesse.

3.5 O MÉTODO DAS APROXIMAÇÕES LINEARES INCREMENTAIS

Da maneira como o problema está estabelecido até o momento, o PVC é resolvido em uma única iteração. Porém, o problema de grande deformação possui propriedades não-lineares, e os tensores utilizados nas equações supõem que o problema é linear. De fato, alguns dos tensores apresentados até o momento, como T , L e F construído via equação

2.40, já se encontram em suas formas linearizadas (e o processo de linearização pode ser encontrado em (PEREIRA, 2015)). Além disso, o problema total será subdividido em vários passos de forma que cada passo seja um problema de pequena deformação, com comportamento aproximadamente linear. Assim, será resolvido um PVC para cada passo de carga de tensão τ , as estruturas serão atualizadas e repete-se o processo até que se chegue a tensão original. Este é o método ALI, apresentado no capítulo 3 de (PEREIRA, 2015).

O algoritmo pode ser resumido da seguinte forma

1. Defini-se os valores iniciais de \mathbf{x} (pontos da malha no espaço), τ (carga de tensão no contorno), F e T , além de um $\Delta\tau$ que será o fator pelo qual a tensão do problema sera subdividida.
2. Conhecidas as funções f^h e g^h , calcula-se seus valores sob x .
3. Resolve-se o problema atual para $u(\mathbf{x}, \tau * \Delta\tau)$
4. Soma-se a solução u à \mathbf{x}
5. Com os novos pontos da malha, atualiza-se $F = (I + H)F$ e $T = T + LH$, com L e H calculados sob a solução mais atualizada.
6. Faz-se o incremento de $\Delta\tau$ em τ e recalcula-se f^h e g^h com a solução mais atualizada e nova carga de tensão.
7. Repete-se os itens 2 a 6 até que τ tenha assumido o valor de interesse

4 IMPLEMENTAÇÃO

A implementação do código foi feita na linguagem Julia, usando como paradigma principal a programação procedural. O código foi desenvolvido sendo genérico quando possível (pensando em problemas com diferentes dimensões e graus de liberdade), mas evitando implementações muito complexas. A escolha da linguagem foi feita a partir de pesquisas anteriores feitas sobre aspectos como facilidade de prototipagem, recursos nativos, desempenho, comunidade ativa, dentre outros (MOTA; MESQUITA, 2025). O ambiente de desenvolvimento escolhido foi um Jupyter Notebook, que permite experimentação interativa através de blocos de códigos que podem ser executadas individualmente e possuem saídas próprias. Na Figura 3 há um exemplo de como ficam organizados os códigos e suas respectivas saídas em um ambiente Jupyter Notebook. Esta estrutura é ótima para *debuggar* trechos de código isoladamente.

Figura 3 – Ambiente notebook

```
[1]: using LinearAlgebra

[2]: M = rand(Float64, (5, 5))

[2]: 5×5 Matrix{Float64}:
 0.38002  0.744058  0.71977  0.0163929  0.492911
 0.556422 0.259447  0.364086  0.501191  0.869248
 0.595241 0.6884   0.0288996 0.99255  0.963217
 0.326027 0.503007  0.513283  0.606675  0.0462027
 0.607339 0.682727  0.0886141 0.786185  0.0912293

[3]: lu(M)

[3]: LU{Float64, Matrix{Float64}, Vector{Int64}}
L factor:
5×5 Matrix{Float64}:
 1.0      0.0      0.0      0.0      0.0
 0.916164 1.0      0.0      0.0      0.0
 0.625713 -0.865659 1.0      0.0      0.0
 0.536811 -0.372938 0.628252 1.0      0.0
 0.98008  -0.0526534 -0.0473523 0.343662 1.0
U factor:
5×5 Matrix{Float64}:
 0.607339 0.682727 0.0886141 0.786185 0.0912293
 0.0      -0.366042 0.282901 -0.219082 0.785667
 0.0      0.0      0.909219 -0.665183 1.11595
 0.0      0.0      0.0      0.520841 -0.410861
 0.0      0.0      0.0      0.0      1.10921
```

Fonte: Autoria própria

Quanto a estruturas, para o armazenamento dos valores da matriz K global utilizou-se a estrutura *SparseMatrixCSC*, nativa da linguagem. Desta forma apenas as entradas não nulas são armazenadas, o que é ideal para casos de matrizes grandes porém esparsas. A solução do sistema linear também utilizou o operador nativo da linguagem, o contra barra (\backslash), que analisa brevemente a matriz de entrada e determina automaticamente um

algoritmo apropriado de resolução. O algoritmo escolheu a decomposição LU para o problema. É possível que ele faça uma escolha melhor ao informar características adicionais sobre a matriz, como se a matriz é positiva-definida, triangular, etc, que não foi o caso. O contra barra é nativamente compatível com a estrutura de matriz esparsa.

Já as funções, uma parte considerável das que aparecem ao longo da teoria exposta no capítulo anterior foram construídas dinamicamente e passadas como objetos chamáveis no código, de forma que torna o código mais facilmente adaptável para diferentes funções. Também utilizou-se de alguns operadores nativos de aplicações de função como a sintaxe $f.(X)$, onde f é função definida para receber apenas uma entrada como argumento, mas que o operador $.$ permite aplicar a todo um objeto iterável (equivalente a aplicação da função *map* de programação funcional).

Um aspecto que aumenta bastante a legibilidade do código e sua comparação com as equações é a compatibilidade da linguagem com caracteres UTF-8 em geral. Desta forma, símbolos como ξ , λ , $_{-1}$ (1 subscripto) e e (e sobrescrito) podem ser incluídos em nomes de variáveis e funções, o que permite a escrita de linhas de código que são idênticas as equações que definem seu comportamento. Um exemplo é exibido na Figura 4, da implementação da equação 3.54.

Figura 4 – Trecho do cálculo da K^e

```

termo1 = 0.0
termo2 = 0.0
termo3 = 0.0

for i in 1:2
    for k in 1:2
        for l in 1:2
            termo1 += T*[r, i] * ∂φ(a, k)(ξ1, ξ2) * dξ_dx[k, i] * ∂φ(b, l)(ξ1, ξ2) * dξ_dx[l, s] * det_J_ξ
            termo2 += T*[r, i] * ∂φ(a, k)(ξ1, ξ2) * dξ_dx[k, s] * ∂φ(b, l)(ξ1, ξ2) * dξ_dx[l, i] * det_J_ξ

            for j in 1:2
                termo3 += L*[r, i, s, j] * ∂φ(a, k)(ξ1, ξ2) * dξ_dx[k, i] * ∂φ(b, l)(ξ1, ξ2) * dξ_dx[l, j] * det_J_ξ
            end
        end
    end
end
end

```

Fonte: Autoria própria

Todas as visualizações e gráficos também foram gerados no mesmo código, pois a linguagem possui bibliotecas que permitem tanto o uso das funcionalidades de plotagem padrões como oferece interface para uso de outras bibliotecas de plotagem amplamente conhecidas, como *PythonPlot* e *Plotly.js*. Por questões de preferência, também foi implementado um visualizador próprio de malha de elementos finitos quadriláteros.

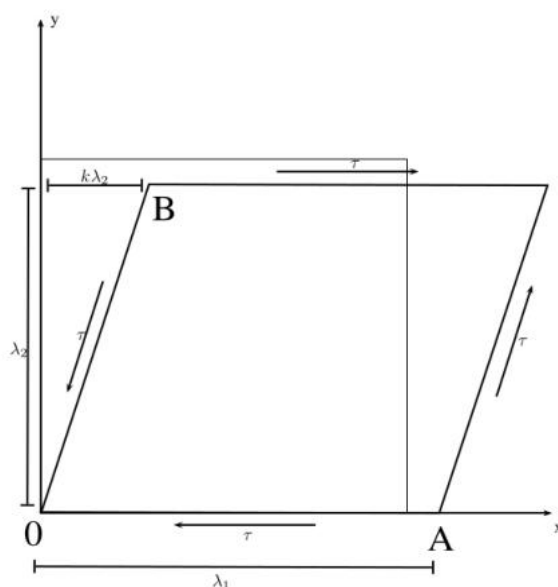
É possível persistir informações no notebook, mas escolheu-se por persisti-los em arquivos separados, usando a biblioteca *DelimitedFiles* para fazer a escrita e posterior leitura para trabalhar com os dados em memória.

5 O PROBLEMA DO CISALHAMENTO

O método construído nos capítulos anteriores é geral para problemas de deformação bidimensional em materiais Mooney-Rivlin compressíveis e quase incompressíveis. Neste capítulo, será trabalhado o problema particular de cisalhamento puro.

O cisalhamento é um fenômeno físico onde as superfícies internas de um objeto deslizam uma sobre a outra. Como o fenômeno tem resultado analiticamente conhecido, é um bom caso de teste para validação da implementação numérica, pois é possível fazer a comparação entre o resultado esperado e o resultado calculado numericamente.

Figura 5 – Diagrama de cisalhamento



Fonte: Pereira, Gabriel Thomaz (2015, p. 46)

Analiticamente, sabe-se que o cisalhamento pode ser descrito da seguinte forma:

$$x_1 = \lambda_1 X_1 + k\lambda_2 X_2 \quad (5.1)$$

$$x_2 = \lambda_2 X_2 \quad (5.2)$$

Para o caso de cisalhamento puro e material elástico, o estado final do corpo é um paralelogramo equilátero. Isto implica em

$$\overline{OA} = \overline{OB} \iff \lambda_1^2 = (1 + k^2)\lambda_2^2 \quad (5.3)$$

Sendo

$$\tau = k\left(s_1\lambda_2^2 - s_2\frac{1}{\lambda_1^2}\right) \quad (5.4)$$

tem-se que

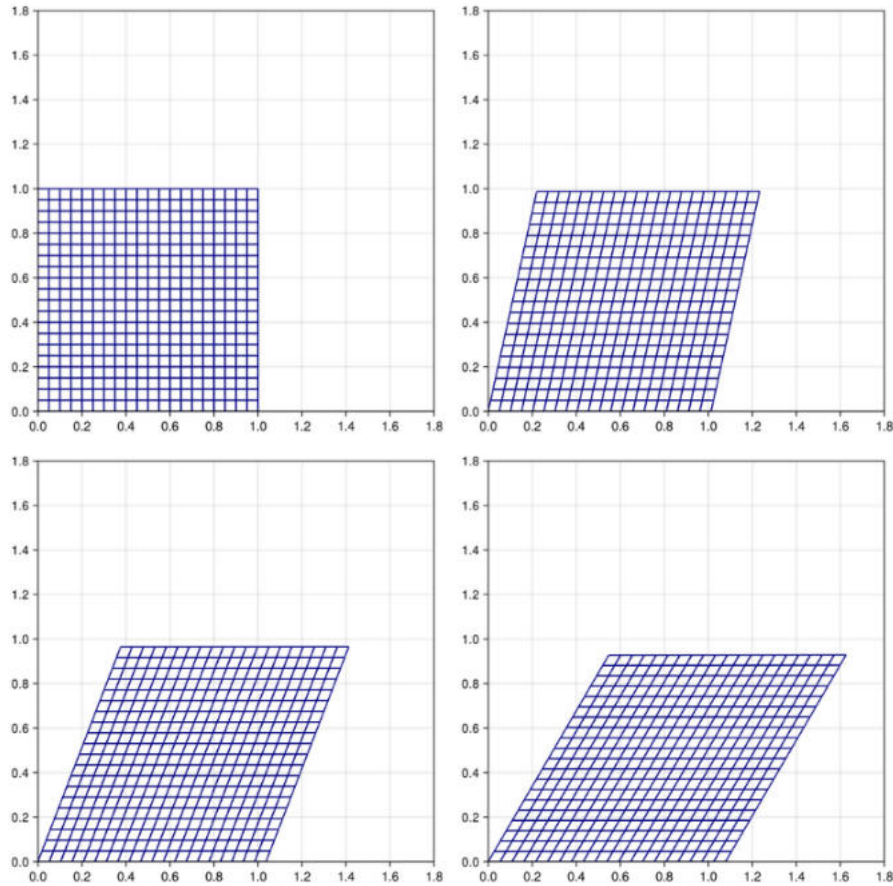
$$\lambda_1 = \left(1 - \tau^2 \frac{1}{(s_1 - s_2)^2} \right)^{-\frac{1}{4}} \quad (5.5)$$

Todas essas propriedades foram retiradas de (PEREIRA, 2015), páginas 46 e 47.

5.1 EXPERIMENTOS NUMÉRICOS E RESULTADOS

Primeiramente, reproduziu-se o experimento de (PEREIRA, 2015) utilizando-se dos mesmos parâmetros: Malha de 20×20 elementos, elementos finitos do tipo Q_4 , domínio $\Omega = (0, 1) \times (0, 1)$, condição de fluxo na fronteira τ , $\Delta\tau = 0.004$, condição prescrita na fronteira sob o eixo x_1 igual a zero na direção x_2 (objeto apoiado sob a superfície), $s_1 = 1$, $s_2 = -0.1$, $\beta = 10^4$, $F_{\text{inicial}} = I$, $T_{\text{inicial}} = 0$.

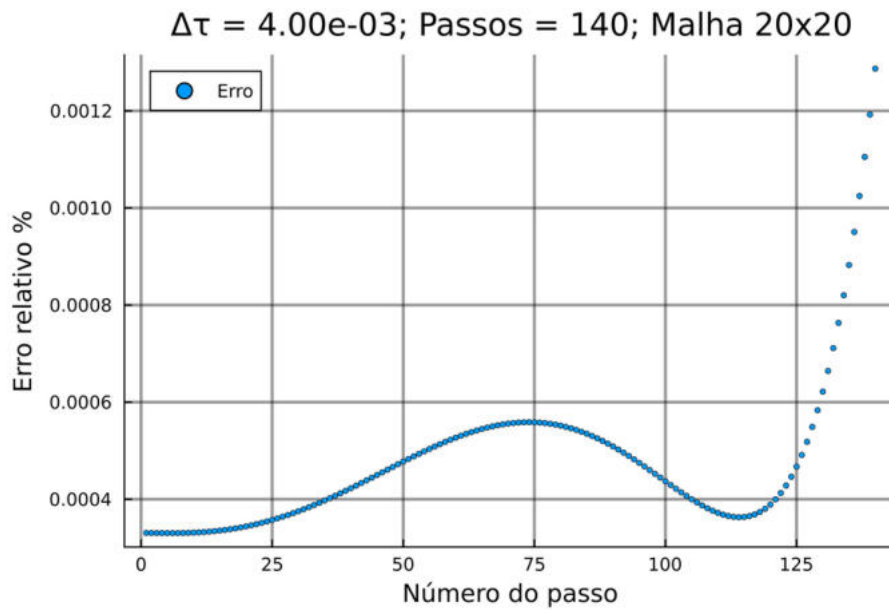
Figura 6 – Solução nos passos 0, 60, 100 e 140



Fonte: Autoria própria

E pode-se observar da Figura 6 que, qualitativamente, o cisalhamento está sendo bem simulado. Para avaliar a precisão da simulação, pode-se observar o gráfico do erro relativo na Figura 7

Figura 7 – Erro relativo do experimento



Tem-se um erro relativo máximo na ordem de 10^{-3} %. O erro relativo está sendo calculado da mesma forma que em (PEREIRA, 2015):

$$E_{\text{rel}} = \left| \frac{\lambda_1^{\text{numérico}} - \lambda_1^{\text{analítico}}}{\lambda_1^{\text{analítico}}} \right| \times 100 \quad (5.6)$$

onde $\lambda_1^{\text{numérico}}$ é calculado pela diferença entre as coordenadas x_1 dos pontos apoiados no eixo x_1 e $\lambda_1^{\text{analítico}}$ é calculado segundo 5.5

Quando comparado ao erro relativo do mesmo experimento em (PEREIRA, 2015), nota-se que houve uma diminuição de uma ordem de grandeza. Porém, para fins de comparação justa, os experimentos originais foram refeitos utilizando o código original da dissertação, gentilmente cedidos pelo autor para este trabalho. O resultado obtido também é da ordem de 10^{-3} %. A hipótese mais plausível levantada sobre esta diferença é que o código original foi executado a aproximadamente 10 anos atrás. Neste meio tempo, os compiladores de C++ (linguagem do código original) devem ter tornado algumas operações com ponto flutuante mais estáveis, bem como possíveis melhores implementações de instruções de máquina nos processadores mais modernos, o que ocasionou o ganho de precisão observado ao diminuir o erro de máquina presente na solução.

Para verificar a estabilidade da implementação, foram realizados 25 experimentos com diferentes combinações de $\Delta\tau$ e refinamento de malha. O que espera-se observar é um menor erro relativo para passos de carregamento menores e malhas mais refinadas (com mais elementos). Na Tabela 1 pode-se verificar o erro relativo máximo para cada experimento, bem como os parâmetros do experimento. É possível notar que, para o problema de cisalhamento e nesta geometria, o refinamento da malha contribui muito pouco para a

diminuição do erro relativo, enquanto que o aumento da quantidade de pequenas cargas possui grande impacto na ordem do erro.

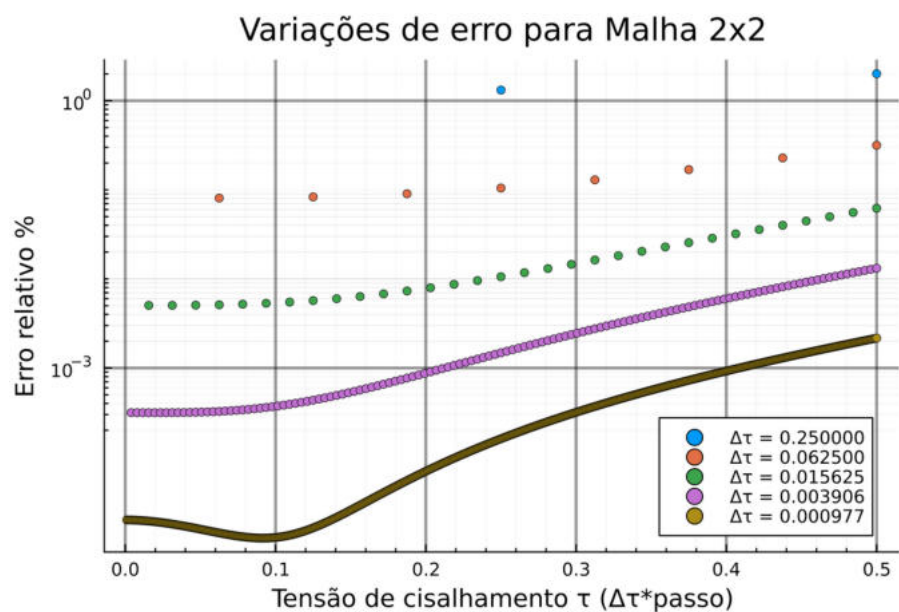
Tabela 1 – Erro relativo máximo dos experimentos

$\Delta\tau \setminus N^{\circ}$ Elementos	2×2	5×5	10×10	20×20	40×40
$0.5/2^1$	2.0115%	2.0100%	2.00468%	1.98605%	1.93680%
$0.5/2^3$	0.3164%	0.3133%	0.30336%	0.27530%	0.23003%
$0.5/2^5$	0.0620%	0.0575%	0.04587%	0.02671%	0.01411%
$0.5/2^7$	0.0131%	0.0093%	0.00387%	0.00038%	0.00056%
$0.5/2^9$	0.0021%	0.0003%	0.00051%	0.00075%	0.00079%

Fonte: Autoria própria

Nas Figuras 8, 9 e 10, são exibidos os gráficos do erro relativo para cada passo de incremento de carga. É possível notar um comportamento parecido entre as malhas para passos de carga grandes enquanto que para as curvas com menor $\Delta\tau$ o erro varia de forma diferente.

Figura 8 – Erro relativo para cada incremento $\Delta\tau$, malha 2×2



O comportamento de queda repentina nos gráficos das malhas 10×10 e 40×40 foi inesperado. Dada a ordem de grandeza do erro relativo e da quantidade de cálculos feitos numericamente, imagina-se que estas flutuações sejam interferência vindas de erros de arredondamento das operações em ponto flutuante.

Figura 9 – Erro relativo para cada incremento $\Delta\tau$, malha 10×10

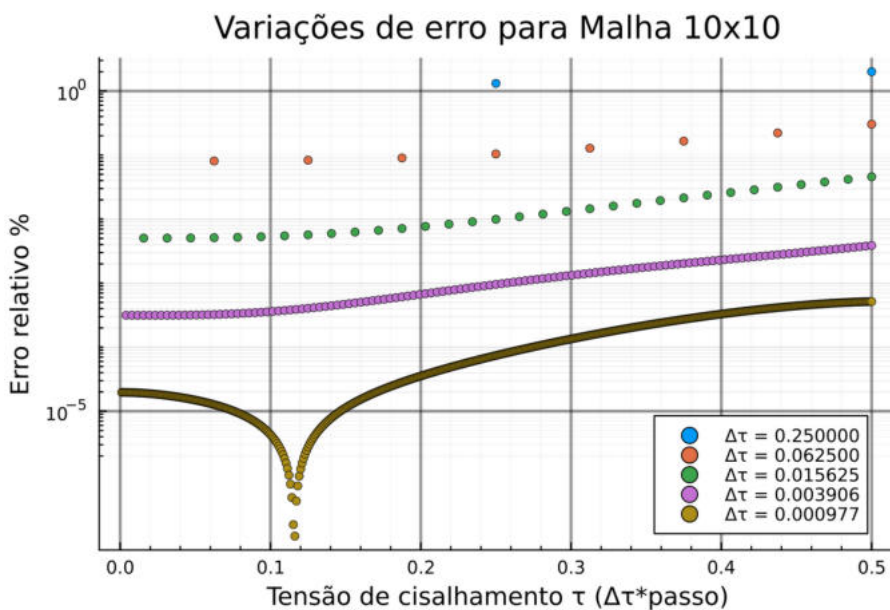
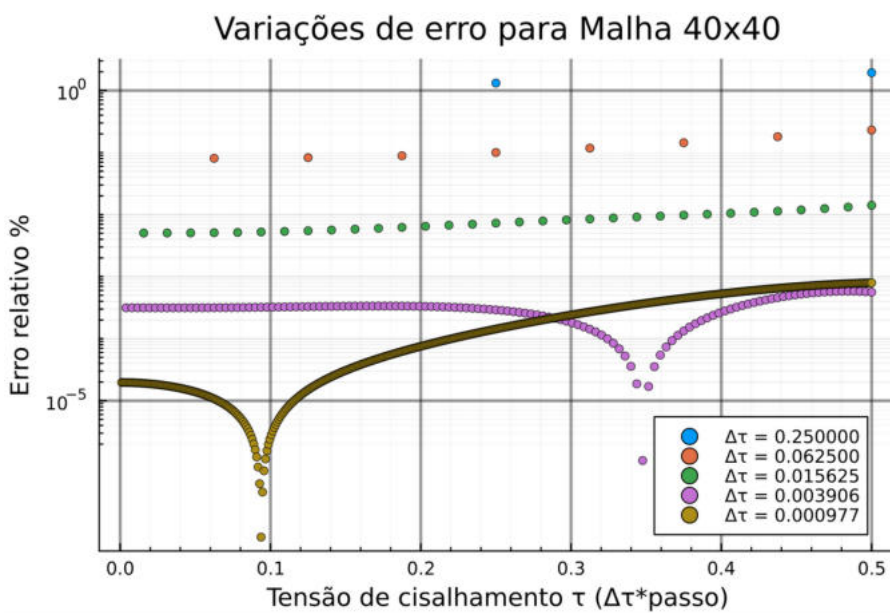


Figura 10 – Erro relativo para cada incremento $\Delta\tau$, malha 40×40



6 CONCLUSÃO

Neste trabalho foi possível verificar as escolhas do método dos elementos finitos e do método das Aproximações Lineares Incrementais se mostraram adequadas para a simulação do problema de cisalhamento sobre um corpo elástico quadrado, com os experimentos mostrando que com passos suficientemente pequenos é possível simular o fenômeno com precisão numérica na ordem de 10^{-4} . Foi possível também concluir que, para este problema, a diminuição do passo de carga possui mais impacto no erro do que o refinamento da malha de elementos finitos. Supõe-se que este fato se deve a natureza relativamente simples da geometria da figura.

As técnicas empregadas para a transição da formulação forte do problema até a formulação final mostraram que o trabalho inicial de aprender a lidar e decompor tensores em componentes vale o esforço, de forma que foi possível simular um problema complexo utilizando as ferramentas amplamente conhecidas do cálculo integral e diferencial após a conversão do problema em componentes.

Tendo sido validado, o programa desenvolvido pode ser utilizado para outros testes, com diferentes parâmetros do material elástico, condições de contorno e malhas. Ainda é necessário, porém, um trabalho de otimização do código, visto que a maior simulação dentre os experimentos apresentados levou cerca de 2 horas para ser realizada.

REFERÊNCIAS

I-SHIH, L. **Introduction to Continuum Mechanics**. Rio de Janeiro: UFRJ/IM, 2019.

MOTA, C. E. V.; MESQUITA, R. F. C. Democratização da pesquisa em elementos finitos: uma revisão bibliográfica de bibliotecas gratuitas de código aberto. Este trabalho foi um projeto de Iniciação Científica apresentado na Semana de Integração Acadêmica da UFRJ. 2025.

PEREIRA, G. T. A. **Modelagem Elastoplástica Utilizando o Método de Aproximações Lineares Incrementais**. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática - Universidade Federal do Rio de Janeiro, 2015.

RINCON, M. A.; I-SHIH, L. **Introdução ao Método dos Elementos Finitos**. Rio de Janeiro: UFRJ/IM, 2020.