UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA
CÂMPUS DE ILHA SOLTEIRA

ANDRÉ DE MORAES VINHA

ANÁLISE DO MÉTODO PERIDINÂMICO COMO POTENCIAL FERRAMENTA DE MANUTENÇÃO PREDITIVA.

Ilha Solteira
2022

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de Ilha Solteira

ANDRÉ DE MORAES VINHA

ANÁLISE DO MÉTODO PERIDINÂMICO COMO POTENCIAL FERRAMENTA DE MANUTENÇÃO PREDITIVA.

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia de Ilha Solteira – Unesp como parte dos requisitos para obtenção do título de Bacharel em Engenharia Mecânica.

Nome do orientador
Prof. Dr. Márcio Antônio Bazani

Ilha Solteira
2022

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

V784a

Vinha, André de Moraes.
Análise do método peridinâmico como potencial ferramenta de manutenção preditiva / André de Moraes Vinha. -- Ilha Solteira: [s.n.], 2022
68 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Mecânica) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2022

Orientador: Márcio Antônio Bazani
Inclui bibliografia

1. Peridinâmica. 2. Implementação numérica. 3. Propagação de trincas.

Raiane da Silva Santos
Supervisora Técnica de Seção
Seção Técnica de Referência, Atendimento ao usuário e Documentação
Diretoria Técnica de Biblioteca e Documentação
CRB/8 - 9999

MODELO DE ATA

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE ENGENHARIA - CAMPUS DE ILHA SOLTEIRA

CURSO DE ENGENHARIA MECÂNICA

ATA DA DEFESA – TRABALHO DE GRADUAÇÃO

TÍTULO: ANÁLISE DO MÉTODO PERIDINÂMICO COMO POTENCIAL FERRAMENTA DE MANUTENÇAO PREDITIVA

ALUNO: André de Moraes Vinha RA: 142053457

ORIENTADOR: Prof. Dr. Márcio Antônio Bazani

Aprovado (x) - Reprovado ( ) pela Comissão Examinadora

Comissão Examinadora:

Prof. Márcio Antônio Bazani *Presidente (Orientador)*

Prof. Rafael da Silva Raqueti

Rafael da Silva Raqueti

Prof. João Angelo Ferres Brogin

João Angelo Ferres Brogin

Ilha Solteira(SP): 16 de Maio de 2022

DEDICATÓRIA

Dedico este trabalho de graduação ao meu pai José Vinha Filho (in memorian), que sempre me apoiou na escolha de caminhos mais ousados na vida.

AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas que de alguma forma me ajudaram na jornada da graduação.

Aos psicólogos, que ajudaram a manter minha saúde emocional nos momentos mais difíceis.

Aos professores da Unesp de Ilha Solteira que fizeram parte da minha graduação em engenharia mecânica, em especial ao meu orientador Prof. Dr. Márcio Antônio Bazani por toda a paciência e dedicação comigo e com meus colegas do grupo de estudos sobre peridinâmica.

A todos os bons amigos e colegas que fiz durante o período que residi na cidade de Ilha Solteira.

À minha família por ter me apoiado de diversas formas nesta jornada.

“*Let everything happen to you: beauty and terror. Just keep going. No feeling is final.”*

Rainer Maria Rilke

RESUMO

Este trabalho faz uma avaliação do método peridinâmico como uma potencial ferramenta de prognóstico para utilização na indústria, em específico no campo da manutenção preditiva. O trabalho apresenta inicialmente uma introdução contendo as considerações gerais sobre os tipos de manutenção existentes na indústria e o auxílio dos métodos numéricos neste campo, seguido pelos objetivos com o escopo do trabalho. Na sequência, há uma introdução teórica com os princípios básicos da teoria peridinâmica, incluindo questões importantes como sua abordagem não-local, horizonte, energia crítica de fratura e inclusão de dano (propagação e ramificação de trincas). Na metodologia encontra-se uma visão geral sobre como foi o processo de realização do trabalho, seguida de uma breve explicação sobre a implementação numérica do programa computacional utilizado e que foi escrito no software MATLAB. Advindos desse programa, estão vários resultados gráficos pertinentes e interessantes, juntamente com uma extensa análise e discussão dos mesmos. Finalizando, há uma conclusão discorrendo sobre os resultados obtidos, as dificuldades encontradas e atestando a eficácia do método peridinâmico como potencial ferramenta de prognóstico na manutenção preditiva.

Palavras-chave: peridinâmica; prognóstico; propagação de trincas; manutenção preditiva; implementação numérica; MATLAB.

ABSTRACT

This work brings an evaluation of the peridynamic method as a potential prognostic tool for industry purposes, specifically in the predictive maintenance field. The work initially presents an introduction with general considerations about the existing types of maintenance in the industry and the aid of numerical methods in this field of study, followed by the objectives with the work's scope. Afterwards, there is a theoretical introduction with the peridynamic theory basic principles, including important subjects like the non-local approach, horizon, critical fracture energy and damage inclusion (crack propagation and branching). In the methodology, an overview about how this work was carried out can be found, followed by a brief explanation about the used numerical implementation of the computer program written in the MATLAB software. As results, there are many relevant and interesting graphic results, together with a extensive analysis and discussion of them. At last, there is a conclusion talking about the obtained results, the difficulties found and attesting the peridynamic method efficiency as a potential prognostic tool in the predictive maintenance field.

Keywords: peridynamics; prognostic; crack propagation; predictive maintenance; numerical implementation; MATLAB.

SUMÁRIO

1. INTRODUÇÃO

1.1 CONSIDERAÇÕES GERAIS

Em um ambiente industrial, é de extrema importância o monitoramente constante da qualidade e da continuidade da produção. Para que máquinas e sistemas mecânicos dos mais diversos tipos continuem a desempenhar seus processos com eficiência, é necessário que ocorram periodicamente atividades de manutenção (Gonçalves e Bazani, 2017).

Em equipamentos, danos que ocorrem de maneira imprevista, desencadeados geralmente por falta de ações ou atividades de manutenção, podem gerar como conseqüência quebras inesperadas, paradas bruscas, perda de tempo na produção, prejuízos financeiros e até acidentes com perdas humanas. Nesse contexto, a manutenção necessária para evitar os problemas citados pode ser entendida como um conjunto de ações. Algumas destas ações são reparos, ajustes, desmontagens, testes, alinhamentos, inspeções, lubrificação, calibração, rotinas preventivas programadas e adequadas, dentre outras (Gonçalves e Bazani, 2017).

De acordo com a norma técnica NBR 5462, que aborda os conceitos e terminologias principais de Confiabilidade e Mantenabilidade, há três tipos de manutenção: a Manutenção Corretiva, a Manutenção Preventiva e a Manutenção Preditiva (Marinelli, 2019). A Figura 1 a seguir mostra representações simples com figuras de equipamentos que diferenciam os tipos de manutenção:

Figura 1 – Representação dos tipos de manutenção.

Fonte: Marinelli, 2019.

A Manutenção Corretiva, de forma simples, é a ação realizada após acontecer a falha no equipamento. Pode ser a recolocação de uma peça nova, de modo que o equipamento volte a exercer a função requerida. A depender da situação, esta manutenção pode ser planejada ou não planejada ou, em outras palavras, emergencial ou programada. No caso em que não é planejada, não há dúvida de que é uma manutenção corretiva. Não existe um plano, a equipe de manutenção é acionada quando a falha acontece. O custo com peças e materiais até a ocorrência é zero, mas após acontecer, geralmente é muito acima do que é gasto com os outros tipos de manutenção, pois também envolve perdas de produção e compras de emergência de novas peças. Finalmente, caso a Manutenção Corretiva seja planejada, provavelmente houve uma decisão gerencial baseada em estudos técnico-financeiros para o mesmo. A intervenção no equipamento ocorre previamente decidida após a falha, provavelmente baseado na premissa de que nesse caso o custo dessa manutenção será menor que em outros métodos (Teles, 2018a).

Na Manutenção Preventiva, o objetivo de reduzir ou evitar falhas e queda no desempenho de equipamentos é baseado em ações realizadas em intervalos predeterminados ou de acordo com critérios prescritos, obedecendo a todo um planejamento. É uma manutenção sistemática realizada de forma prévia que inclui ações como: inspeções, reapertos, substituição de peças desgastadas, limpezas, lubrificações, ajustes, etc. Os critérios que norteiam um plano de Manutenção Preventiva são geralmente estatísticos, independentemente do equipamento realmente precisar ou não de reparos e é muito utilizada pelo seu baixo custo e simplicidade, permitindo aumentar e garantir os índices de confiabilidade e disponibilidade dos equipamentos. A freqüência com que esta manutenção ocorre pode estar ligada ao tempo, quilometragem, produtividade ou outros fatores (Abecom, 2021).

Finalmente, a Manutenção Preditiva é um método de manutenção que se baseia na condição do equipamento ou componentes da máquina. É a mais complexa, exigindo um conhecimento técnico avançado e instrumentos específicos para obtenção e análise de dados. Exemplos comuns de manutenção preditiva são: termografia, ultrassonografica, análise de vibrações, medição de corrente, ferrografia, etc. Os dados em questão são relacionados a fenômenos como desgaste, temperatura, vibração, etc. Quando estes atingem valores próximos a

limites previamente estabelecidos, inicia-se a manutenção, de forma a corrigir o motivo da variação do fenômeno. Seu objetivo é predizer o comportamento e encontrar falhas em estágio inicial no equipamento, em um momento em que ainda não sejam danosas ao próprio equipamento e processo de produção (Teles, 2017b).

A seguir, na Figura 2, estão presentes algumas representações simples dos possíveis dados que podem ser utilizados para auxílio na manutenção preditiva:

Figura 2 – Exemplos de possíveis dados para auxílio na manutenção preditiva.

Fonte: IClass, 2021.

Na engenharia, relativo a prognósticos, existe uma abordagem chamada *Prognostics and Health Monitoring* (PHM) que basicamente tem como objetivo o monitoramento da capacidade de componentes, sistemas e estruturas de resistir a carregamentos estruturais, térmicos e químicos. Baseado em informações relacionadas à aplicação anterior e diagnóstico momentâneo, o PHM é uma ferramenta ou sistema muito útil para predizer a condição futura de um componente ou sistema. A partir disso, o prognóstico revela a vida útil remanescente do componente, ou *Remaining Useful Life* (RUL), que pode ser considerada como o período de tempo depois do qual a performance do componente ou equipamento não é suficiente para atender os requisitos operacionais mínimos (Gonçalves e Bazani, 2017).

A estimativa de uma RUL é de extrema importância para o planejamento e agendamento de atividades envolvendo manutenção por parte de engenheiros. Estas ações por sua vez refletem não só em segurança, como também no lucro, reduzindo custos de manutenção e operação e diminuindo tempo de inatividade das máquinas. Neste contexto, um fator importante para a predição de uma RUL

adequada é o conhecimento preliminar sobre o comportamento de degradação do sistema, este por sua vez podendo ser representado tanto por modelos matemáticos baseados nas descrições físicas do sistema, quanto por modelos baseados em dados de medição, os *data driven models* (Gonçalves e Bazani, 2017).

Dos modelos matemáticos existentes e que são utilizados dentro do domínio da Mecânica do Contínuo, a grande maioria é baseada no método numérico dos elementos finitos, ou *finite element method,* FEM (Azevedo, 2003). Este método é predominante tanto na industria quanto na academia no que diz respeito à modelagem de problemas envolvendo danos estruturais. A partir dele é possível resolver equações diferenciais parciais governantes, levando em conta as condições de fronteira do sistema a ser analisado e discretizando-o em diversos elementos. Cada elemento possui nós ou pontos materiais que, sob a ação de forças ou carregamentos, podem se deslocar. Por meio das equações constituintes do método numérico utilizado, pode-se descobrir a interação que ocorre entre esses pontos materiais e com isso o comportamento do sistema como um todo (Azevedo, 2003).

Um comportamento de interesse para se prever é a quebra ou ruputura de um componente ou sistema. Como mencionado anteriormente, a quebra geralmente surge da nucleação e propagação de trincas, e estas dependem de certas condições no carregamento do componente, como amplitude, freqüência, razão de tensão e também fatores ambientais, como mudanças de temperatura e processos químicos. Importante ressaltar que essas condições e fatores são contestáveis ou incertos, tornando difícil o estudo e predição do crescimento de trincas (Gonçalves e Bazani, 2017).O fenômeno envolvendo a trinca é bastante complexo e, apesar de vários progressos, ainda apresenta muitas limitações quando abordado pelo FEM, principalmente pela formulação matemática e esforço computacional necessário (Gonçalves e Bazani, 2017).

Com relação à formulação matemática, a desvantagem está no fato de que com o FEM padrão, existe a hipótese de que o corpo sempre permanece contínuo após se deformar, o que não é a realidade quando ocorre algum dano ou propagação de trincas. Dessa maneira torna-se difícil realizar modelos e formulações matemáticas sobre uma região de descontinuidade, como uma fissura ou falha. Em torno das pontas da fissura, existem singularidades de tensões, e o FEM tradicional não define estas singularidades (Gonçalves e Bazani, 2017).

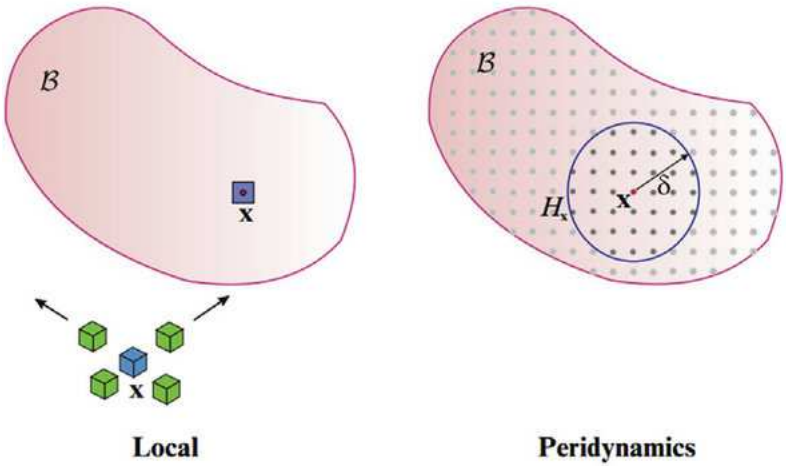
Também é de conhecimento que no FEM não podem ser realizadas em torno de uma superfície de trinca as derivadas espaciais para as equações diferenciais parciais governantes. Quando acontece ao acaso que a borda de uma malha de FEM coincida com a superfície de uma falha, existem meios para se lidar com as tensões das trincas, mas geralmente não acontece, pois uma trinca propaga-se aleatoriamente pelo material (Gonçalves e Bazani, 2017).

As limitações mencionadas anteriormente justificam a questão do grande esforço computacional necessário para se modelar um problema envolvendo trincas com o FEM padrão. A cada iteração da programação a malha deve se ajustar ao contorno da falha à medida que ela cresce. Por fim, foi observado experimentalmente que é devido a irregularidades microestruturais do material que ocorre a nucleação e a propagação de trincas. Modelos contínuos não conseguem abranger este nível de escala (A. Gonçalves e M. Bazani, 2017).

Para agregar no campo da Manutenção Preditiva, uma nova ferramenta de prognóstico vem se destacando para detecção de formação de trincas, desde os seus estágios iniciais, com nucleação e propagação, até a quebra de um componente. A nova ferramente se chama Peridinâmica (PD), um método apresentado por Stewart Silling (Silling, 2000) com expectativas de suprir as limitações do FEM em lidar com nucleação e propagação de trincas em elementos. A PD é uma reformulação não-local da Mecânica do Contínuo e deixa de lado o pressuposto ou hipótese de que um corpo continua contínuo após se deformar (Silling, 2000).

A Figura 3 apresenta uma representação prática da diferença entre modelos contínuos locais e não-locais. Nesta figura, no domínio à esquerda com a abordagem local, é possível observar que só são levadas em consideração as interações do ponto “x” com os pontos que ele tem contato direto, ao seu redor. Já no domínio da esquerda, com a abordagem não-local da PD, é possível observar que são levadas em consideração todas as interações do mesmo ponto “x” com todos os pontos contidos dentro de uma “vizinhança” H_x delimitada por um raio δ. Essa é uma das diferenças fundamentais entre o FEM e a PD e fator importante para estudo da propagação de trincas na teoria peridinâmica.

Figura 3 – Relação entre modelos contínuos locais e não-locais.



Fonte: Madenci and Oterkus, 2014.

A Peridinâmica baseia-se na formulação integral das equações constitutivas de movimento, mas não inclui derivadas espaciais de deslocamentos, pois estas não são apropriadas para as descontinuidades geradas por trincas. No lugar das derivadas de deslocamentos, a PD utiliza deslocamentos (Gonçalves e Bazani, 2017).

A utilização do Método Peridinâmico como uma nova ferramenta é uma ação muito importante para contribuir com o campo da manutenção preditiva e com a indústria como um todo. Implementações numéricas baseadas na PD e realizadas em diferentes softwares, tanto livres quanto de licença comercial, são uma forma eficiente para difundir a potencialidade da ferramenta em analisar e predizer o comportamento do material, em especial a propagação de trincas.

1.2 OBJETIVOS

O principal objetivo deste trabalho é realizar uma análise do método peridinâmico como potencial ferramenta de prognóstico no campo da manutenção preditiva. Para tal, serão estudados os principais pontos da teoria da peridinâmica, juntamente com a análise qualitativa de resultados obtidos com uma implementação numérica realizada no software MATLAB.

O objetivo secundário será realizar uma comparação entre os dois “solucionadores” ou *solvers* da implementação numérica utilizando simulações com

uma mesma configuração e analisar os resultados progressivos de simulações com adição de trincas.

2. FUNDAMENTOS TEÓRICOS

Tomando como base um corpo qualquer em um determinado instante de tempo. O corpo pode ser representado discretizado ou dividido em infinitos pontos materiais. Em um instante inicial, onde não existe deformação, cada ponto material pode ser indicado em um sistema como $\mathbf{x}_{(k)}$, sendo $(k = 1, 2, \dots, \infty)$, e a eles também estão atrelados um volume incremental $\boldsymbol{V}_{(k)}$, e uma densidade de massa $\boldsymbol{\rho}(\mathbf{x}_{(k)})$ (Madenci and Oterkus, 2014).

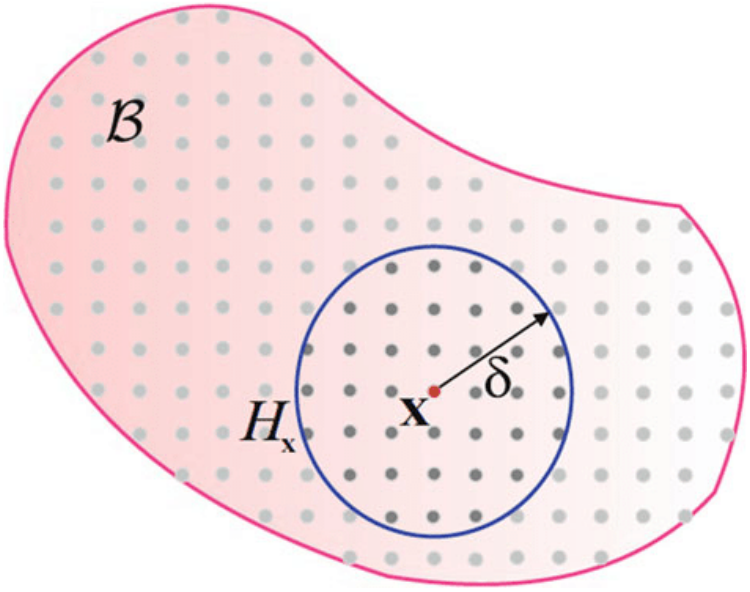
Para que haja movimento e deformação, os pontos materiais devem sofrer a ação de carregamentos, deslocamentos ou velocidades. Sob essas ações prescritas, o ponto material $\mathbf{x}_{(k)}$ passa por um deslocamento $\mathbf{u}_{(k)}$, e sua localização no sistema de coordenadas é expressa pelo vetor posição $\mathbf{y}_{(k)}$ em um novo estado deformado. Nesse caso, as respectivas representações para carregamento externo e deslocamento são $\boldsymbol{b}_{(k)}(\mathbf{x}_{(k)}, \boldsymbol{t})$ e $\boldsymbol{u}_{(k)}(\mathbf{x}_{(k)}, \boldsymbol{t})$ (Madenci and Oterkus, 2014).

Como mencionado anteriormente, uma das características chaves da teoria Peridinâmica é o fato de que analisa-se não apenas a interação do ponto material $\mathbf{x}_{(k)}$ com seus vizinhos adjacentes, mas também sua interação com outros pontos dentro de uma região maior denominada vizinhança ou família, $\boldsymbol{H}_{\boldsymbol{x}_{(k)}}$. O estado de cada ponto é determinado pela sua interação entre pares com esses outros pontos $\mathbf{x}_{(j)}$, sendo $(\boldsymbol{j} = \mathbf{1}, \mathbf{2}, \dots, \infty)$, dentro da região de $\boldsymbol{H}_{\boldsymbol{x}_{(k)}}$. Da mesma forma, o ponto material $\mathbf{x}_{(j)}$ tem sua interação com outros pontos em uma vizinhança $\boldsymbol{H}_{\boldsymbol{x}_{(j)}}$. É muito importante ressaltar que a interação de um par de pontos materiais só é formada quando a distância entre eles é menor ou igual à um raio de tamanho $\boldsymbol{\delta}$, chamado de "horizonte" (E. Madenci and E. Oterkus, 2014).

Na Figura 4 é possível observar alguns dos elementos principais da teoria Peridinâmica mencionados anteriormente, como os pontos materiais, a vizinhança ou família do ponto $\mathbf{x}$ delimitada por $\boldsymbol{H}_{\boldsymbol{x}}$ e de raio $\boldsymbol{\delta}$. Outro fato interessante e relevante sobre a teoria da PD é que, à medida que o horizonte δ diminui, mais locais se tornam as interações entre os pontos materias. A partir disso pode-se

afirmar que, com o horizonte tendendo a zero, a teoria clássica da elasticidade pode ser considerada um caso limitante da PD (Patriota, 2018).

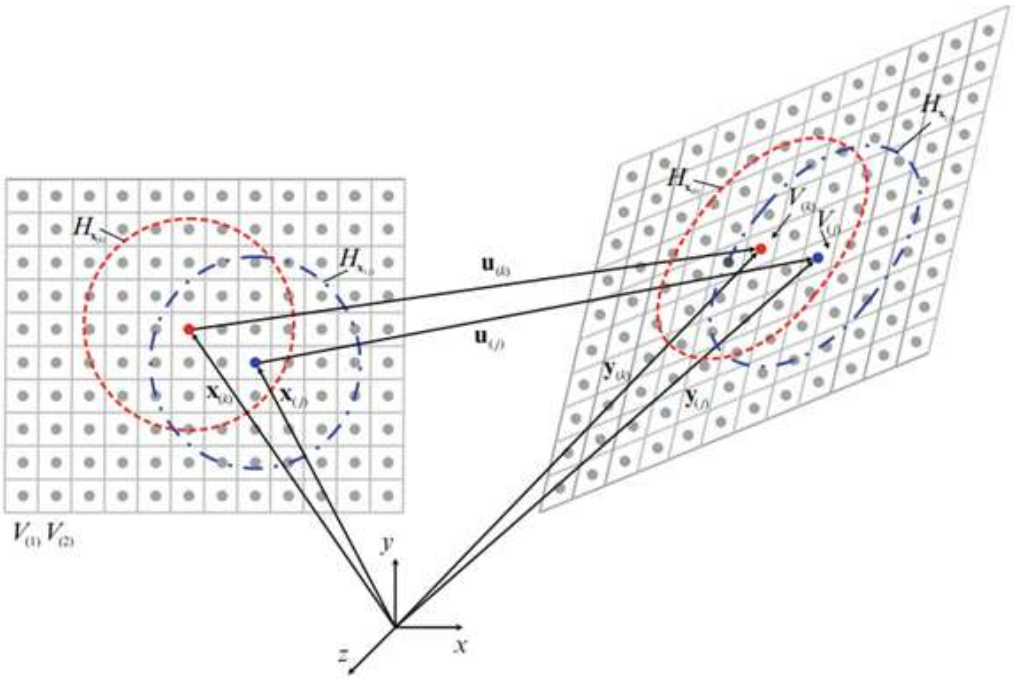
Figura 4 – Esquema para conceito de Horizonte na Peridinâmica.



Fonte: E. Madenci and E, Oterkus, 2014.

Como ilustrado na Figura 5, todo ponto material, seja $\mathbf{x}_{(k)}$, $\mathbf{x}_{(j)}$ ou outro qualquer interage e é influenciado pela deformação conjunta de outros pontos materiais presentes em suas respectivas famílias, $\boldsymbol{H}_{\boldsymbol{x}_{(k)}}$, $\boldsymbol{H}_{\boldsymbol{x}_{(j)}}$ ou outra qualquer (Madenci and Oterkus, 2014).

Figura 5 – Cinemática de pontos materiais na Peridinâmica.



Fonte: E. Madenci, E. Oterkus, (2014).

No estado inicial, sem deformação, seu vetor de posição relativa inicial (posição relativa entre dois pontos de diferentes horizontes) pode ser definido como:

$$\xi_{(k)(j)} = \mathrm{x}_{(j)} - \mathrm{x}_{(k)} \tag{1}$$

Os deslocamentos $\mathrm{u}_{(k)}$ e $\mathrm{u}_{(j)}$ também podem ser expressos em função das posições como:

$$\mathrm{u}_{(k)} = \mathrm{y}_{(k)} - \mathrm{x}_{(k)} \text{ e } \mathrm{u}_{(j)} = \mathrm{y}_{(j)} - \mathrm{x}_{(j)} \tag{2}$$

E por sua vez os deslocamentos expressam um deslocamento relativo entre dois pontos pertencentes a dois horizontes distintos:

$$\eta_{(k)(j)} = \mathrm{u}_{(j)} - \mathrm{u}_{(k)} \tag{3}$$

Para relacionar a posição de dois pontos distintos nos estados deformado e não-deformado, têm-se o deslocamento relativo S:

$$S_{(k)(j)} = \frac{\left|\mathrm{y}_{(j)} - \mathrm{y}_{(k)}\right| - \left|\mathrm{x}_{(j)} - \mathrm{x}_{(k)}\right|}{\left|\mathrm{x}_{(j)} - \mathrm{x}_{(k)}\right|} \tag{4}$$

Fazendo-se substituições, utilizando as Equações (1) e (3), a Equação (4) pode também ser representada na forma:

$$S_{(k)(j)} = \frac{\left|\xi_{(k)(j)} + \eta_{(k)(j)}\right| - \left|\xi_{(k)(j)}\right|}{\left|\xi_{(k)(j)}\right|} \tag{5}$$

Assumindo que dentro do domínio ou vizinhança, $H_{x_{(k)}}$, as interações entre os pares de pontos materiais seguem a Segunda Lei de Newton, a equação de movimento para a Peridinâmica pode ser apresentada de forma mais genérica como (Monelli, 2020):

$$\rho(\mathrm{x})\ddot{u}(\mathrm{x},t) = \int_{H_x} f(\eta,\xi)\, dV_x + b(\mathrm{x},t) \tag{6}$$

Na Equação (6), dentre os termos que ainda não foram apresentados, no lado esquerdo está a aceleração, $\ddot{u}(\mathrm{x},t)$, que basicamente é a derivada de segunda ordem do campo vetor deslocamento. Enquanto isso, no lado direito está o termo $f(\eta,\xi)$, relacionado às forças internas por unidade de volume exercidas entre os pontos materiais. Tais forças internas podem possibilitar a deformação e, por conseguinte, o desenvolvimento de trinca no material, e o termo é integrado sobre a vizinhança H_x . Como citado anteriormente, o termo $b(\mathrm{x},t)$ representa um carregamento externo, ou forças aplicadas sobre o material, como por exemplo cisalhamento, compressão, tração, etc.

Baseado no proposto por Stewart Silling (Silling, 2000), que envolve a teoria da microelasticidade linear, a função $f(\eta,\xi)$ pode ser modelada a partir da função de energia potencial microelástica, $\omega(\eta,\xi)$:

$$f(\eta,\xi) = \frac{\partial}{\partial\eta}\omega(\eta,\xi) \tag{7}$$

Levando em consideração a suposição de que o alongamento do material modifica-se linearmente com a força entre um par de pontos, a energia potencial microelástica fica na forma (Dias, Bazani, Paschoalini, Barbanti, 2017):

$$\omega(\eta, \xi) = \frac{C(\xi) S^2 \xi}{2} \tag{8}$$

Na Equação (8), o termo novo é o micromodulo de elasticidade ou constante de ligação, $C(\xi)$ (Dias, Bazani, Paschoalini, Barbanti, 2017). Este representa a rigidez da ligação entre pontos materiais e é um fator de proporcionalidade dependente das características do material (Dias, Bazani, Paschoalini, Barbanti, 2017). Considerando que este fator tenha um valor constante, $C(\xi) = C$, ele pode ser calculado da seguinte maneira:

$$C = \frac{6E}{\pi \delta^3 (1 - \nu)} \tag{9}$$

A Equação (9), do micro-módulo de elasticidade, depende tanto do módulo de Young ou módulo de elasticidade, E, quanto do coeficiente de Poison, ν, ambos características do material em questão (Dias, Bazani, Paschoalini, Barbanti, 2017). Sua obtenção se dá igualando-se a função da energia de deformação comum com a da energia de deformação própria para a Peridinâmica. Esta última é obtida integrando-se o micropotencial elástico sob a região do horizonte (Dias, Bazani, Paschoalini, Barbanti, 2017):

$$W(x) = \frac{1}{2} \int_{H_x} \omega(\eta, \xi)\, dx' = \frac{1}{2} \int_{-\delta}^{\delta} \left[\frac{C(\xi) S^2 r}{2} \right] 2\pi r\, dr = \frac{\pi}{6} C(\xi) S^2 \delta^3 \tag{10}$$

Na Peridinâmica, falha ou dano ocorrem quando são anuladas as interações ou micropotenciais entre os pares de pontos materiais inseridos em um horizonte (Dias, Bazani, Paschoalini, Barbanti, 2017). Se o deslocamento ou estiramento, $S_{(k)(j)}$, for maior que um valor crítico, S_c, os pares de força de contato desaparecem, os pontos materiais não interagem mais entre si, a trinca se formou (Dias, Bazani, Paschoalini, Barbanti, 2017).

O dano (trinca) é expressado nas equações de movimento com a remoção dos vetores de densidade de força entre os pontos materiais de forma irreversível. Uma das conseqüências é a redistribuição do carregamento entre os pontos materiais do corpo, o que conduz a um crescimento progressivo da trinca de forma autônoma. Informações importantes como a direção e a velocidade de propagação da trinca recém iniciada são adquiridas na medida em que vão se quebrando ou anulando as interações entre os outros pontos do domínio (Dias, Bazani, Paschoalini, Barbanti, 2017).

O valor crítico para o estiramento que causa a falha nas interações, S_c, é uma função dependente do módulo de Young do material, E, do raio do horizonte, δ, e de outro importante fator denominado energia crítica de fratura, G_0. Em problemas possíveis de serem abordados em 2D, a expressão para esse valor crítico é (Dias, Bazani, Paschoalini, Barbanti, 2017):

$$S_c = \sqrt{\frac{4\pi G_0}{9E\delta}} \tag{10}$$

Finalmente, fazendo manipulações com as Equações (5) e (8) na Equação (7), e considerando o valor crítico da Equação (10), a função relacionada às forças internas exercidas entre os pares de pontos materiais pode ser expressa da seguinte forma (Dias, Bazani, Paschoalini, Barbanti, 2017):

$$f(\eta,\xi) = \begin{cases} c(\xi) S \dfrac{\xi+\eta}{\|\xi+\eta\|}, & S_{(k)(j)} < S_c \\ 0, & S_{(k)(j)} \geq S_c \end{cases} \tag{11}$$

3. METODOLOGIA

3.1 VISÃO GERAL

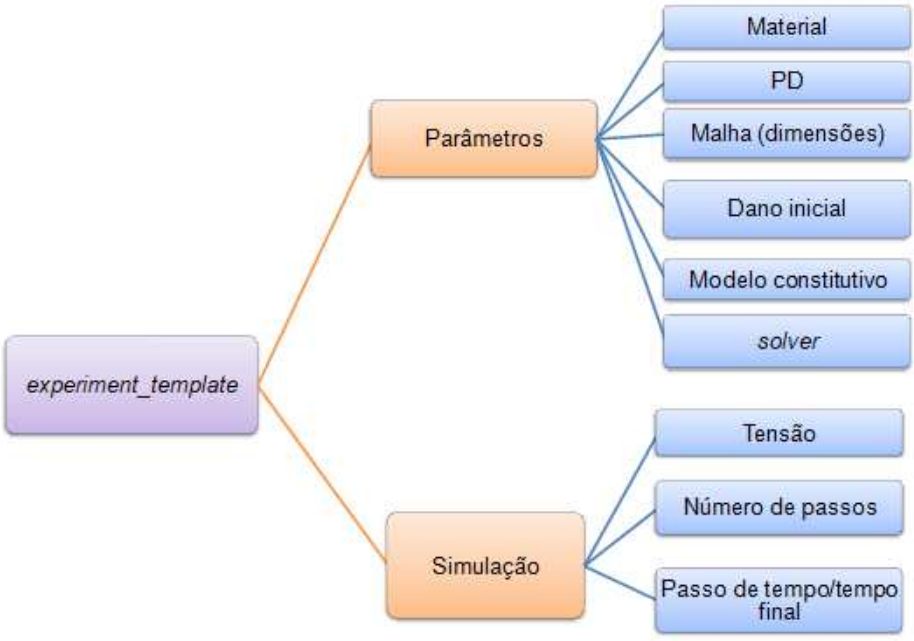
A realização deste Trabalho de Graduação teve como base o estudo da teoria Peridinâmica por meio de vários trabalhos acadêmicos como *papers*, teses, dissertações e livros contendo o assunto, assim como o estudo e análise de um

complexo programa de computador escrito no software Matlab por Túlio V. B. Patriota (Patriota, 2018). A tese de doutorado do autor do programa computacional em questão também foi base para o estudo, tanto para questões teóricas, quanto para aprender uma forma eficaz de implementação numérica da Peridinâmica, e por conseqüência o entendimento do funcionamento do código.

A implementação numérica no Matlab em si é bastante ramificada, sendo diversas funções interligadas. Inicialmente, o usuário acessa três *templates* (modelos) ou funções principais, que são a *experiment_template*, a *generateMesh* e a *prescribedBC*.

A função *experiment_template*, representada de forma simples por um organograma presente na Figura 6 a seguir, é a principal das três mencionadas. Com ela o usuário escolhe os parâmetros importantes para a simulação, como as características físicas do material (módulo de elasticidade, coeficiente de Poisson, densidade, energia crítica de deformação), geometria, parâmetros peridinâmicos (raio do horizonte, razão de malha), dano ou entalhe inicial, tipo de modelo constitutivo para cálculo de forças internas e a energia de deformação, o tipo de *solver* (solucionador), tensão aplicada, número de passos, passo de tempo e tempo final, etc.

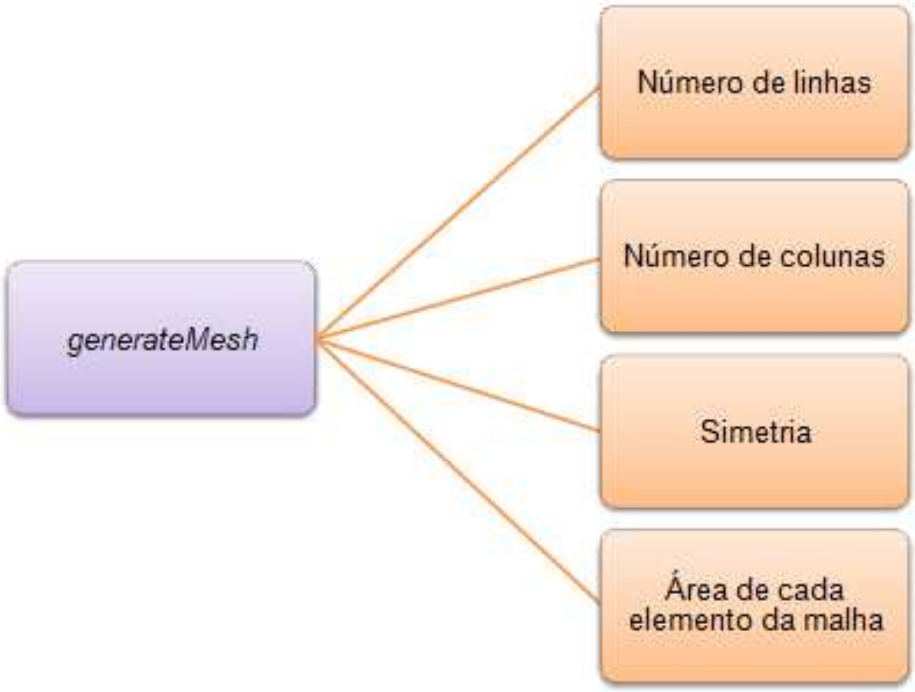
Figura 6 – Organograma da função *experiment_template.*



Fonte: Elaborado pelo prório autor.

A *generateMesh* é uma função mais simples, onde basicamente manipula-se alguns detalhes da malha, influenciando tanto a simulação quanto o pós-processamento, este último sendo os resultados em representação gráfica. É importante destacar que, neste programa, o domínio ou malhas limitam-se a geometrias quadradas ou retangulares. A Figura 7 a seguir aresenta um organograma da função *generatMesh*, onde o usuário pode modificar número de linhas, número de colunas, simetria e área de cada elemento de malha.

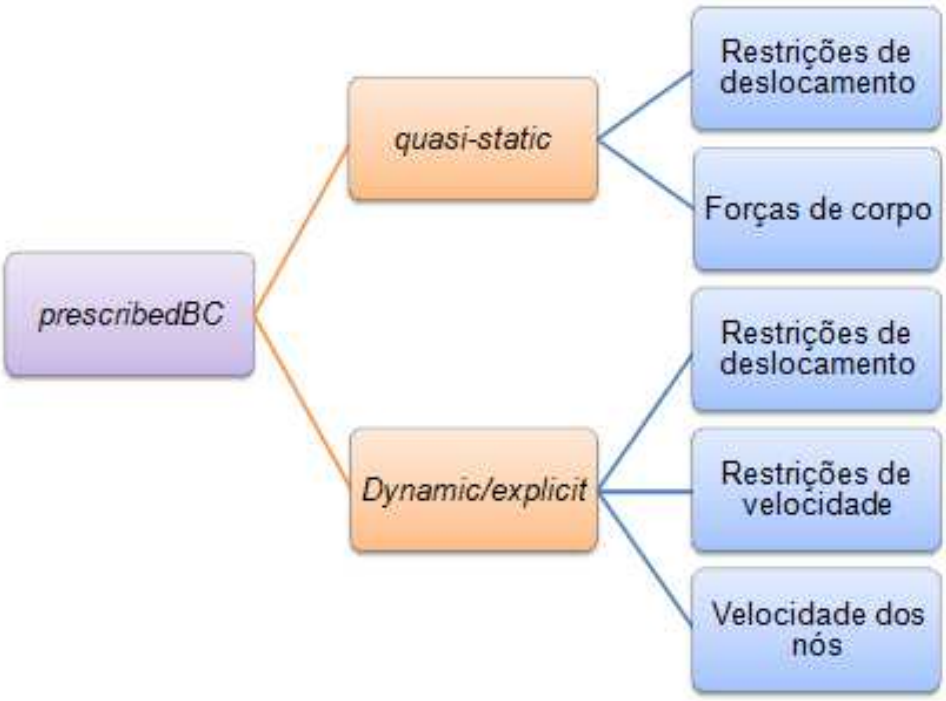
Figura 7 Organograma da função *generateMesh.*



Fonte: Elaborado pelo próprio autor.

A última das três funções principais citadas anteriormente, a *precribedBC*, também muito importante, serve para manipular as condições de fronteira da simulação. A escolha das condições de fronteira dependerá do *solver* a ser utilizado. Caso seja o *quasi-static solver*, o usuário deve inserir as restrições de deslocamento e as forças de corpo, mas caso seja o *dynamic/exlicit solver,* deve-se inserir as restrições de deslocamento, restrições de velocidade e velocidade prescrita de cada nó ou onto material. A Figura 8 apresenta um organograma sobre a função *prescribedBC.*

Figura 8 – Organograma da função *prescribedBC.*



Fonte: Elaborado pelo próprio autor.

A simulação em si é rodada ou acionada pela *experiment_template,* que é a principal. As outras funções mencionadas, *generateMesh* e *prescribedBC,* são ligadas na primeira, ou como se descreve no jargão de programação, são “chamadas” pela primeira. Existem diversas outras funções no programa que estão armazenadas em pastas secundárias. Estas funções são responsáveis por todo processamento e pós-processamento da simulação, e são chamadas diretamente na função principal (*experiment_template*) com suas opções.

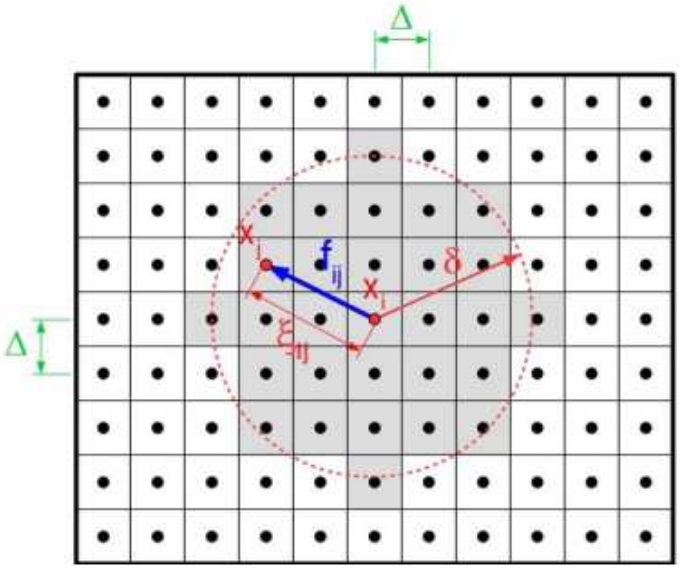
3.2 IMPLEMENTAÇÃO NUMÉRICA

É importante ressaltar que, devido às dificuldades em se obter soluções para problemas envolvendo deformações e falhas com a Peridinâmica, simulações numéricas são uma das melhores opções para entender um problema relacionado e até para uso em aplicações práticas (Patriota, 2018).

Neste tópico serão apresentados de forma geral os pontos mais importantes da implementação numérica realizada pelo autor do programa, Túlio V. B. Patriota (Patriota, 2018).

Tomando como base a Figura 9, a malha total em questão pode ser representada como um domínio bidimensional discretizado $\mathcal{B}_D$, enquanto que $\mathcal{B}$ representa o domínio bidimensional de referência. $\mathcal{B}_D$ pode ser aproximado a $\mathcal{B}$ por uma malha homogênea. Neste esquema, cada ponto i possui uma posição descrita por x_i ,situada no centro de uma célula quadrada τ_i (parte discretizada do domínio total) de lado $\Delta = h$, densidade ρ_i e área h^2. Também estão presentes na Figura 6 o termo das forças internas (f_{ij}), exercidas entre os pontos materiais i e j, o raio δ, denotando o tamanho do horizonte, a posição relativa (ξ_{ij}) entre os dois pontos e Δ, que é a distância entre dois pontos materiais vizinhos (Patriota, 2018).

Figura 9 – Esquema de discretização de um domínio na Peridinâmica.



Fonte: J. P. Dias, M. A. Bazani, A. T. Paschoalini, L. Barbanti, 2017.

As variáveis físicas do problema são ponderadas nestes pontos, e a Equação (6), ou Equação da Segunda Lei de Newton, é espacialmente integrada usando a regra de quadratura do ponto médio, *midpoint quadrature rule.* Em vista disso, considerando o ponto $x_i \epsilon \mathcal{B}_D$, a equação governante de movimento da Peridinâmica no formato discretizado fica na forma (Patriota, 2018):

$$\rho_i \ddot{\mathrm{u}}_i = \sum_{j \epsilon \mathcal{F}_i} \mathrm{f}_{ij} \Delta V_{ij} + b_i \qquad (12)$$

Nesta equação, $\ddot{u}_i = \ddot{u}(x_i)$ é a aceleração, $b_i = b(x_i)$ é uma força ou carregamento externo, $\mathcal{F}_i$ é uma família de índices correlacionadas com o nó i e ΔV_{ij} é o volume do ponto material j dentro do horizonte do ponto material i.

Outra questão muito importante para a teoria Peridinâmica, e que é bem destacada no trabalho de Túlio V. B. Patriota (Patriota, 2018), é o tamanho do horizonte, δ. Sendo um parâmetro do material, como mencionado anteriormente na Seção 2, um tamanho ínfimo de horizonte deixa as interações entre os pontos materiais cada vez mais locais, retomando a abordagem da teoria clássica da elasticidade. Em outros casos, para macroestruturas onde pode-se trabalhar com um δ maior, é aconselhado que o tamanho deste seja suficiente para proporcionar resultados conclusivos sem que haja custo computacional desnecessário (Patriota, 2018).

Para se ter uma medida melhor de que tamanho de horizonte usar, principalmente no caso de uma simulação numérica, existe a taxa ou proporção de malha, ou *mesh ratio* (Patriota, 2018):

$$d = \frac{\delta}{h} \tag{13}$$

De acordo com o autor, quanto maior o valor dessa taxa, “mais precisa é a quantidade integral mensurada na vizinhança com raio δ de um ponto material”. Também de acordo com Túlio V. B. Patriota (Patriota, 2018), uma boa prática para garantir a precisão da integração numérica é a escolha de uma taxa d adequada, sendo usual:

$$\begin{cases} 4 < d < 6\,, \text{ para simulações em 2D, e} \\ 6 < d < 9, \text{ para simulações em 3D.} \end{cases}$$

Após esta escolha, para assumir um tamanho de horizonte adequado, realiza-se uma verificação da convergência de δ. Sobre esta convergência, aparentemente existem dois tipos citados em trabalhos numéricos: δ*-convergence*, em que d é mantido fixo e δ se aproxima de zero e com isso a solução aproximada converge para a solução clássica da abordagem local (relembrando a questão de ser um caso limitante da PD, citada na Seção 2), e a d*-convergence*, em que δ é mantido fixo e d

é aumentado, e a partir disso a solução aproximada é convergida para a solução analítica peridinâmica. Resultados em outros trabalhos indicam que a combinação de valores pequenos tanto para horizonte quanto para taxa de malha podem culminar em soluções errôneas (Patriota, 2018).

3.2.1 *SOLVER: QUASI-STATIC OR DYNAMIC/EXPLICIT*

Para realizações de testes e simulações com o programa, existem duas opções de solucionadores ou *solvers*, o quase-estático e o dinâmico-explícito. Para validação do primeiro, realizou-se simulações de experimentos quase estáticos, enquanto que para a validação do segundo *solver*, simulações de experimentos dinâmicos.

3.2.1.1 SOLUCIONADOR DINÂMICO EXPLÍCITO

Para realização de simulações dinâmicas, há uma integração no tempo exercida por um método de diferença central utilizando um esquema de tempo explícito denominado *Velocity Verlet*. Tal esquema é representado no Algoritmo 1 (Patriota, 2018), presente no Anexo A.

Pelo programa do autor Túlio V. B. Patriota (Patriota, 2018) levar em conta um método explícito, o incremento de tempo Δt deve ser pequeno o bastante para garantir uma estabilidade. No caso, é definido um incremento crítico de tempo:

$$\Delta t_{crit} = \min\left(\sqrt{\frac{2\rho}{\sum_{j\in\mathcal{F}_i} C_{ij}\Delta V_{ij}}}\right) \tag{14}$$

Na Equação (14), C_{ij} é o micro-módulo de elasticidade calculado em relação ao vetor de posição relativa inicial (ligação), $\xi_{ij} = \mathrm{x}_j - \mathrm{x}_i$. Em casos 2D, a expressão do micro-módulo pode ser derivado de:

$$C_{ij} = |\mathbf{C}_{ij}| \qquad \mathbf{C}_{ij} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}}\Big|_{(\boldsymbol{\eta}=0,\boldsymbol{\xi}=\boldsymbol{\xi}_{ij})} \tag{15}$$

A Equação (14) acima, $\mathbf{f}$ é uma função da ligação (forças internas exercidas entre os pares de pontos materiais) e do deslocamento relativo entre dois pontos $\boldsymbol{\eta}$. Realizando alguma manipulação matemática entre esta última Equação (15) e a Equação (11), tem-se:

$$\mathbf{C}(\boldsymbol{\xi}) = \frac{C\omega_\delta(|\boldsymbol{\xi}|)}{|\boldsymbol{\xi}|^3}\begin{bmatrix} \xi_1^2 & \xi_2\xi_1 \\ \xi_1\xi_2 & \xi_2^2 \end{bmatrix} \tag{16}$$

Na expressão acima, os subscritos de ξ são os componentes cartesianos do vetor de ligação (posição relativa inicial) $\boldsymbol{\xi}$. C continua sendo uma constante de ligação ou micro-módulo de elasticidade, podendo ser calculado pela Equação (9).

3.2.1.2 SOLUCIONADOR QUASE ESTÁTICO EM TEMPO IMPLÍCITO

A Equação (12), (equação de movimento da peridinâmica), tem uma forma dinâmica, o que é indicado pelo termo de inércia $\rho_i \ddot{u}_i$. Por conta disso, ela não seria diretamente aplicável para casos estáticos ou quase-estáticos. Para contornar a situação, existem alguns tratamentos especiais ou técnicas que podem ser aplicados ao sistema para convergi-lo a uma condição estática em um período curto de tempo. Geralmente tais técnicas são baseadas no fato de que a solução estática é parte do estado estacionário da resposta transitória da solução (Patriota, 2018).

Partindo dessas informações, com intuito de realizar simulações com soluções no estado estacionário, pode-se zerar o termo de inércia $\rho_i \ddot{u}_i$, o que fará a equação governante discretizada ficar na forma (Patriota, 2018):

$$\sum_{j \in \mathcal{F}_i} \mathrm{f}_{ij} \Delta V_{ij} + b_i = 0 \tag{17}$$

Com o objetivo de reescrever a equação anterior em forma vetorial, assume-se que l e k são respectivamente os graus de liberdade do primeiro e segundo componentes cartesianos do campo de deslocamento do nó i. Desse modo a equação fica na forma (Patriota, 2018):

$$\bar{\mathrm{f}}_{int} + \bar{\mathrm{b}} = 0 \tag{18}$$

Na Equação (18), as componentes l e k de $\bar{\mathrm{f}}_{int}$ ficam na forma:

$$\bar{f}_{int,l} = \sum_{j \in \mathcal{F}_i} f_{ij,1} \Delta V_{ij}, \qquad \bar{f}_{int,k} = \sum_{j \in \mathcal{F}_i} f_{ij,2} \Delta V_{ij} \tag{19}$$

com $f_{ij,1}$ e $f_{ij,2}$ sendo os componentes cartesianos da força de interação f_{ij}. Para dar clareza a notação utilizada, cada variável de um vetor com uma sobrelinha refere-se ao índice de um nó, enquanto que as que não possuem sobrelinha referem-se ao índice do grau de liberdade. De acordo com o autor Túlio V. B. Patriota (Patriota, 2018), a formulação por vetor traz mais vantagens que a matricial, pois permite uma abordagem mais simples das restrições cinemáticas.

Para abordar a condição de simulação no estado estacionário, utiliza-se no programa o *quasi-static implicit time solver*, que é baseado no método de Newton (método de análise numérica que tem como objetivo estimar as raízes de uma função). Em vez dos passos de tempo, são as condições de carregamento que são atualizadas a cada passo, o que leva a uma configuração desbalanceada que implica em uma solução de campo de deslocamento atualizada. Assim como em métodos de cálculo numérico, o algoritmos para este *solver* estima a solução do campo de deslocamento que minimiza um residual escalar r para um passo ou etapa de carregamento.

Nessa abordagem, para cada etapa de carregamento o valor de r, que tem unidade de força, deve ser nulo para uma solução numérica da Equação (17). De outra forma, utiliza-se um esquema iterativo para alcançar uma solução aproximada da Equação (17) que aproxima r para um valor menor que um limite r_ϵ. O Algoritmo 2, presente no Anexo B deste trabalho, apresenta o procedimento citado acima.

3.2.2 CONDIÇÕES DE CONTORNO

Este tópico é muito importante para o funcionamento do programa e realização das simulações. Existem três tipos de condições de contorno: a restrição de deslocamento, a restrição de velocidade e o carregamento externo.

As condições de contorno estão diretamente relacionadas com os solucionadores ou *solvers.* Tanto o dinâmico quanto o estático utilizam a restrição de deslocamento, mas o primeiro só funciona com a restrição de velocidade e o segundo com o carregamento externo.

3.2.2.1 RESTRIÇÃO DE DESLOCAMENTO

No código de Túlio V. B. Patriota (Patriota, 2018), as restrições são aplicadas diretamente nos nós ou pontos materiais e apenas deslocamentos constantes podem ser prescritos. Para este caso, há um vetor contendo os componentes de deslocamentos prescritos em um determinado tempo. Ele pode ser representado como:

$$\bar{u}^{n}(\Gamma_{d}) = u_{p} \tag{20}$$

Nessa expressão ou vetor, o termo Γ_{d} representa o conjunto de graus de liberdade de restrição, onde o deslocamento é estabelecido, n é um determinado instante discreto de tempo e u_{p} é o vetor que carrega os deslocamentos prescritos.

3.2.2.2 RESTRIÇÃO DE VELOCIDADE

Esse caso é similar ao anterior, com a diferença de que Γ_{v} é o conjunto de graus de liberdade de restrição onde velocidade constante é estabelecida, v_{p} é um vetor contendo as velocidades prescritas neste conjunto e Δt é o incremento de tempo (Patriota, 2018).

$$\bar{u}^{n}(\Gamma_{v}) = \bar{u}^{n-1}(\Gamma_{v}) + v_{p}\Delta t \tag{21}$$

A expressão acima corresponde ao vetor que contem as componentes de deslocamento em Γ_{v} em um determinado instante discreto de tempo n. Essa restrição é a que rege o *Dynamic/Explicit Solver*, assim como a possibilidade de incluir trincas na simulação.

3.2.2.3 CARREGAMENTO EXTERNO

No teoria podem ser aplicadas forças externas de contato, tanto de tração quanto de compressão, por unidade de área. Neste caso as rotinas de programação são baseadas nos modelos clássicos do contínuo.

Na abordagem do programa, as forças de tração/compressão são aproximadas como forças de corpo por unidade de volume, agindo em uma camada fina próxima à fronteira onde as forças de tração/compressão são aplicadas. Na Equação (21) a seguir, Ω_t representa o conjunto de nós na fronteira do domínio discretizado $\mathcal{B}_D$ onde as forças são prescritas, Δx é o comprimento da célula normal à força que se encontra na fronteira ou condição de contorno, d é a razão de malha e A_{Ω_t} é a área de uma única célula ou nó (Patriota, 2018). Lembrando que, para uma malha homogênea, $A_{\Omega_t} = h^2$.

$$\mathrm{b}\left(\mathrm{x}_{\Omega_t}\right) = \frac{\mathrm{T}\Delta x}{dA_{\Omega_t}} \tag{22}$$

Também presente na Equação (22) está a força de tração T. Esta contém o tensor de tensão σ e o vetor unitário normal em relação a superfície N:

$$\mathrm{T} = \sigma\mathrm{N} \tag{23}$$

3.2.3 MODELOS CONSTITUTIVOS IMPLEMENTADOS

No trabalho do autor Túlio V. B. Patriota (Patriota, 2018), existem várias opções para implementação de modelos constitutivos para simulações. Tais modelos têm a função de calcular a densidade das forças internas $\mathrm{f}_{\mathrm{int}}$ e a densidade de energia de deformação W.

Para os resultados obtidos neste trabalho, apenas dois modelos foram estudados e utilizados, o *Linearized Bond-based model* (LBB), e o *Prototype micro brittle model* (PMB).

3.2.3.1 *LINEARIZED BOND-BASED MODEL* (LBB)

Este é o modelo mais simples utilizado no trabalho do autor Túlio V. B. Patriota (Patriota, 2018), sendo uma versão linearizada do modelo seguinte (*rototype Micro Brittle Model*) mas sem dano incluído. A questão da relação entre modelo de dano aplicado na simulação será discutida em outra seção.

Neste modelo, o cálculo da densidade das forças internas é dado pela Equação (24):

$$\mathrm{f}_{int}(\mathrm{x}_i,\mathrm{u}) = \sum_{j\in\mathcal{F}_i} C_{LBB}\,\omega(\xi_{ij})\,\frac{\xi_{ij}\otimes\xi_{ij}}{|\xi_{ij}|^2}\,\eta_{ij}\Delta V_{ij}. \tag{24}$$

Nesta equação, o símbolo $\otimes$ indica o produto dos tensores ξ_{ij} e o termo C_{LBB} é o micro-modulo de elasticidade, que para o modelo é calculado na forma:

$$C_{LBB} = \frac{6E}{m} \tag{25}$$

Na Equação (25), E é o módulo de Young do material e m é um termo chamado volume ponderado.

A densidade de energia de deformação para este caso será dada por:

$$W_{LBB}(\mathrm{x}_i) = \frac{1}{4}\sum_{j\in\mathcal{F}_i} C_{LBB}\,\frac{\omega_\delta}{|\xi_{ij}|}\left(|\xi_{ij}|\right)\left(\eta_{ij}.\,\xi_{ij}\right)^2\Delta V_{ij} \tag{26}$$

3.2.3.2 *PROTOTYPE MICRO BRITTLE MODEL* (PMB)

Com este modelo já é possível realizar simulações com a adição de dano ou trinca inicial. Isso ocorre graças à adição de uma função binária que também é chamada de fator de dano μ (Patriota, 2018):

$$\mu(x) = \begin{cases} 1, & x < S_c, \\ 0, & x \geq S_c \end{cases}. \tag{27}$$

Nesta equação aparece de novo um termo muito importante, que é o valor crítico de deslocamento entre dois pontos, S_c. É possível notar uma semelhança entra a Equação (27) e a Equação (11), presente nos Fundamentos Teóricos, que é uma função relacionada às forças internas exercidas entre os pares de pontos materiais. No modelo PMB, a função da densidade das forças internas é calculada na forma:

$$\mathrm{f}_{int}(\mathrm{x}_i) = \sum_{j \in \mathcal{F}_i} C_{PMB}\omega_\delta(|\xi_{ij}|)\mu(S_{ij,max})\frac{|\xi_{ij}+\eta_{ij}|-|\xi_{ij}|}{|\xi_{ij}|}\mathrm{e}_{ij}\Delta V_{ij} \tag{28}$$

Na Equação (28), o micro-módulo de elasticidade do modelo, C_{PMB}, é calculado da mesma forma que o do modelo anterior na Equação (25). Os outros dois termos novos da expressão são o vetor unitário na direção da ligação deformada,

$$\mathrm{e}_{ij} = \frac{(\xi_{ij}+\eta_{ij})}{|\xi_{ij}+\eta_{ij}|},$$

e $S_{ij,max}$ é o estiramento ou alongamento máximo das ligações entre os nós i e j durante uma simulação. A densidade de energia de deformação para este modelo é:

$$W_{PMB}(\mathrm{x}_i) = \frac{1}{4}\sum_{j \in \mathcal{F}_i} C_{PMB}\omega_\delta(|\xi_{ij}|)\mu(S_{ij,max})\left(\frac{|\xi_{ij}+\eta_{ij}|-|\xi_{ij}|}{|\xi_{ij}|}\right)^2|\xi_{ij}|\Delta V_{ij}. \tag{29}$$

3.2.4 BALANÇO DE ENERGIA

Outro ponto importante para análise de resultados, tanto no que diz respeito à teoria quanto à implementação numérica, é o balanço de energia. A evolução desta no tempo auxilia na percepção da estabilidade das simulações no programa. A energia total do sistema não pode aumentar, e no caso de não se dissipar, deve ser constante (Patriota, 2018).

A energia total $E(t^n)$ do sistema ou corpo deformado é o somatório das energias cinética $K(t^n)$, potencial $P(t^n)$ e do trabalho externo $W^E(t^n)$:

$$E(t^n) = K(t^n) + P(t^n) - W^E(t^n) \tag{30}$$

Um fato comum para os membros desse somatório é a validação no tempo $t^n = n\Delta t$, lembrando que n é o tempo e Δt um incremento de tempo.

As energias cinética e potencial discretas são calculadas respectivamente:

$$K(t^n) = \frac{1}{2}\sum_{i\in\mathcal{B}_D} \rho_i |\mathrm{v}_i^n|^2 V_i \tag{31}$$

e

$$P(t^n) = \sum_{i\in\mathcal{B}_D} W(\mathrm{x}_i, t^n) V_i \tag{32}$$

Relembrando alguns termos presentes nas Equações (31) e (32), V_i é o volume da célula i, $\mathcal{B}_D$ é o domínio bidimensional discretizado, v_i é o vetor de velocidade para cada ponto e W é a função da densidade de energia de deformação, que depende do modelo a ser usado (LBB ou PMB).

Por fim, o trabalho externo $W^E(t^n)$ dependerá da condição de contorno aplicada na simulação, o que por sua vez influenciará na escolha do *solver.* Se a condição de contorno for a de carregamento externo e que seja constante no tempo, a expressão pode ser, de forma simplificada:

$$W^E(t^n) = \sum_{i\in\mathcal{B}_D} \mathrm{b}_i . \mathrm{u}_i^n V_i \tag{33}$$

Caso a condição de contorno seja restrição de velocidade, o equacionamento torna-se maior. Com a prescrição de uma velocidade, implicitamente há a prescrição de uma força externa, que pode ser expressada e calculada de forma semelhante à Equação (12), colocando-se em evidência o carregamente externo. O trabalho do autor do programa, Túlio V. B. Patriota (Patriota, 2018), apresenta mais detalhes sobre essa questão.

A idéia principal envolvendo o balanço de energia é que, caso não haja dissipação no sistema, todo o trabalho externo deve acelerar os pontos e “deformá-

los", o que fará com que o balanço de energia total fique nulo (T. V. B. Patriota, 2018):

$$E(t) = 0, \ \ \forall t \in [0, t_{\text{final}}]. \tag{34}$$

Porém, caso haja a dissipação,

$$\frac{dE}{dt} < 0, \tag{35}$$

e isso indicará que a fonte de dissipação, para o caso das simulações desse programa, são as nucleações e propragações de trincas para o caso do modelo que inclui dano prévio (PMB).

Como mencionado anteriormente no começo dessa Seção (3.2), optou-se por mostrar apenas o que considerei essencial na implementação numérica do programa. No total, existem diversas funções intrincadas no programa do autor Túlio V. B. Patriota (Patriota, 2018), sendo que todas são necessárias para o funcionamento correto da simulação, tanto no processamento quanto no pós-processamento.

4. RESULTADOS E DISCUSSÕES

Os resultados abordados nesse trabalho são focados em dois pontos: a comparação entre simulações utilizando os *solvers quasi-static* e *dynamic/explicit* e a comparação entre simulações e resultados experimentais com a inclusão de trincas.

Em todos os resultados um único material foi utilizado, o *soda-lime glass*, que é um tipo de vidro mais grosseiro que é amplamente utilizado em garrafas de refrigerante. As propriedades do material são encontradas na Tabela 1 a seguir:

Tabela 1: Propriedades materiais do *soda-lime glass*.

Propriedade	Valor
Módulo de Young $\boldsymbol{E}$ (GPa)	72
Densidade $\boldsymbol{\rho}$ (kgm^{-3})	2440
Coeficiente de Poisson	0.3
Energia Crítica de Fratura (Jm^{-2})	3.8

Fonte: Patriota, 2018.

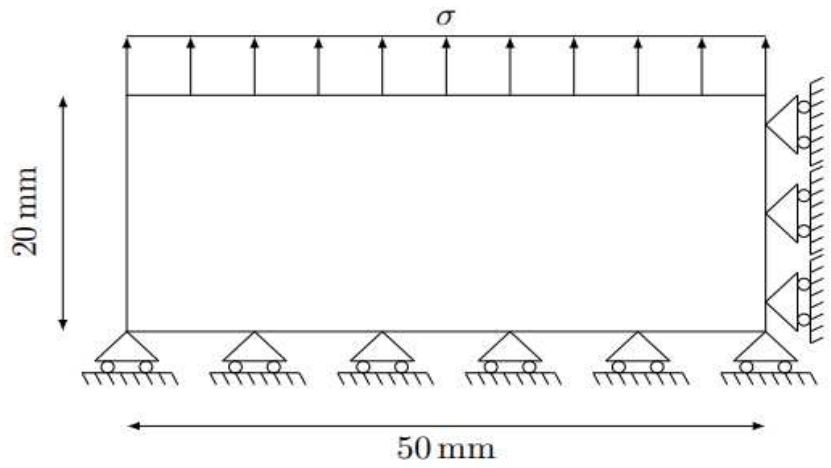
Relacionado ao estudo de propagação de trincas, o *soda-lime glass* é um dos mais utilizados, tanto na abordagem da peridinâmica quanto nas do MEF.

4.1 *QUASI-STATIC* X *DYNAMIC/EXPLICIT*

Neste subtópico, a análise e validação de ambos os *solvers* foi realizada utilizando um único modelo, o *Linearized Bond-based model* (LBB). Obviamente, a configuração para as simulações foi a mesma. A configuração em questão relaciona-se às propriedades do material, geometria, malha, restrições e os parâmetros peridinâmicos (horizonte e razão de malha). Essas primeiras simulações foram baseadas nas primeiras que o autor do programa, Túlio V. B. Patriota (Patriota, 2018), apresenta em sua tese, com intuito de validar a implementação correta dos modelos numéricos e rotinas.

As simulações a seguir são testes de tração em uma placa de vidro retangular em 2D de 20 mm de altura por 50 mm de comprimento. Suas bordas inferior e lateral direita estão restritas por apoios móveis, enquanto que a borda superior é tracionada no sentido vertical. Um esquema da configuração pode ser obeservado na Figura 10 a seguir:

Figura 10 - Configuração para validação dos *solvers.*



Fonte: Patriota, 2018.

Na simulação, as restrições são aplicadas em uma camada de nós ou pontos, enquanto que as tensões σ são aplicadas em três camadas. Isso fica visivelmente melhor nas figuras referentes às condições de contorno para cada simulação.

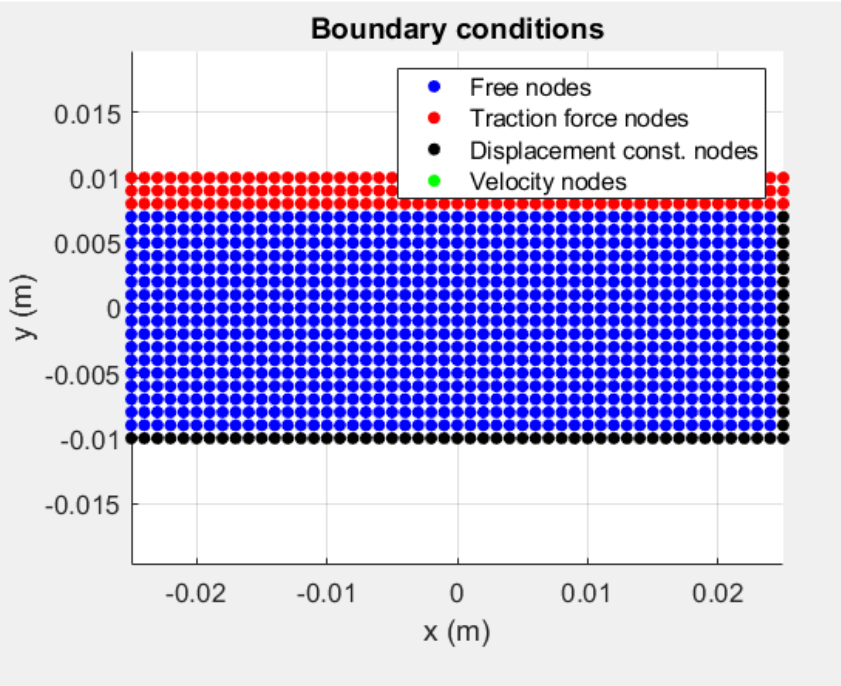
No que se refere aos termos relacionados à peridinâmica, para ambas as simulações optou-se um raio do horizonte $\delta = 5$ mm e a razão de malha $d = 5$, o que implica em um tamanho de lado de célula para cada nó $h = 1$ mm. Como pode ser observado, o tamanho do raio do horizonte é grande em relação às dimensões do retângulo, então pode-se considerar que a malha é grosseira.

O pós-processamento, que são os resultados gráficos do programa, é diferente para cada *solver.* No caso do *quasi-static,* são apresentados ao final da simulação oito gráficos, enquanto que no *dynamic/explicit*, sete gráficos. Ambos os casos possuem gráficos em comum, seis no total para comparação, enquanto os restantes são diferentes, representando resultlados particulares para cada caso.

4.1.1 *QUASI-STATIC*

O primeiro tipo de gráfico, apresentado na Figura 11, aparece em qualquer tipo de simulação do programa e mostra as condições de contorno escolhidas pelo usuário.

Figura 11 - Condições de contorno da simulação utilizando *quasi-static solver.*



Fonte: Elaborada pelo próprio autor.

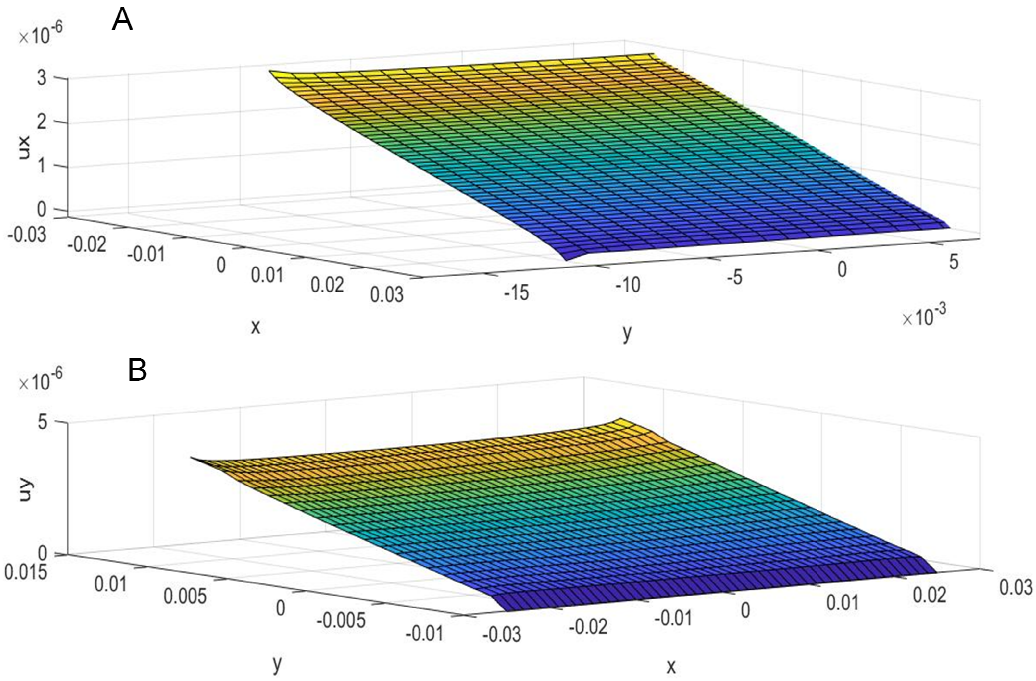
O gráfico referente às condições de contorno apresenta nas legendas os quatro tipos de possibilidade para os nós da malha. Em azul é representado os nós livres, em vermelho os que sobrem carregamento externo, em preto os nós restritos e em verde os nós com velocidade.

É importante lembrar da relação entre as condições de contorno e o *solver* usado. Na implementação do programa, quando se faz uma simulação com o *quasi-static solver*, obviamente só será possível escolher ou preencher a restrição de deslocamento e o carregamento externo, como foi discorrido na subseção 3.2.2. A inclusão da restrição de velocidade causará erro no programa, não rodando o mesmo.

Nessa simulação em particular foi utilizado uma tensão $\sigma = 4$ MPa, valor que pode ser mudado livremente na função *experiment_template*, com direção vertical e sentido para cima, detalhes que são implementados na função *prescribedBC*.

O próximo gráfico, presente na Figura 12, apresenta o campo de deslocamento por eixo cartesiano.

Figura 12 - Campo de deslocamento por eixo cartesiano para simulação *quasi-static solver.*



Fonte: Elaborada pelo próprio autor.

Como mencionado no próprio título, ambos os gráficos acima mostram o deslocamento que a malha sofre em ambas as direções do eixo cartesiano em decorrência da tração aplicada. Para simulações com este *solver*, é possível observar que o deslocamento não é tão uniforme em certas regiões, principalmente nos extremos ou contorno da malha. Isso ocorre devido ao *surface effects*, ou efeitos de superfície, que levam à redução da rigidez do material perto das fronteiras ou contorno. Tal efeito acontece normalmente na vizinhança ou horizonte da região da fronteira e é causado pela impossibilidade da existência de um horizonte circular (ou esférico) na região da fronteira, o que afeta a forma de como quantidades materiais são avaliadas no local (Patriota, 2018). De forma simples, nas regiões fronteiriças da malha onde não se pode delimitar um horizonte or completo, as interações entre os pontos materiais ficam incompletas, deficitárias, levando à pequenas falhas ou oscilações nos deslocamentos.

É importante destacar que os efeitos de superfície são maximizados quando as forças ou tensões são aplicadas em camadas menores e mais externas dos nós da malha. No entanto, eles podem ser minimizados com a confecção de uma malha mais refinada, escolhendo razão de malha e raio do horizonte adequados, e

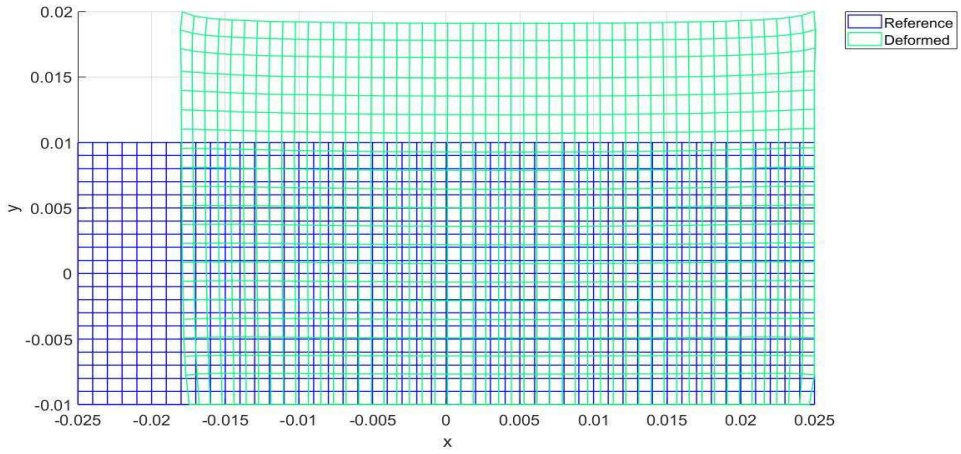
utilização de uma função de influência (Patriota, 2018). Este tipo de função não é abordada neste trabalho por questões práticas, mas está presente na tese e programa do autor Túlio V. B. Patriota (Patriota, 2018).

Como mencionado anteriormente, esta simulação possui uma malha grosseira, o tamanho do raio do horizonte δ é grande em relação ao próprio domínio como um todo. A malha grosseira e a falta de utilização de alguma função de influência não são tão relevantes para os resultados deejados neste trabalho, não fazem parte do escopo principal, que é mostrar a diferenças entre os *solvers*, modelos constitutivos e funcionamento do programa.

Para fins mais práticos, as médias de valores de deslocamento máximo em cada direção são $\bar{u}_x \cong 2{,}8592.10^{-6}$m e $\bar{u}_y \cong 3{,}7143.10^{-6}$m. A escolha do tamanho de casas decimais para esses resultados foi igual a exibida no pós-processamento (gráficos) do programa.

O próximo gráfico, referente à Figura 13, é uma representação da comparação entre o domínio antes de sofrer a ação da tensão (*reference*) e depois (*deformed*). Neste caso, o autor do programa inseriu no código um fator de escala (*scale factor*) de aproximadamente $8{,}37.10^3$ com intuito de aumentar e ressaltar o formato deformado, o que faz com que o gráfico seja apenas ilustrativo. Apesar disso, é possível observar a ação dos efeitos de superfície na representação deformada, agindo principalmente nas fronteiras do domínio onde ao aplicadas restrições e tensões.

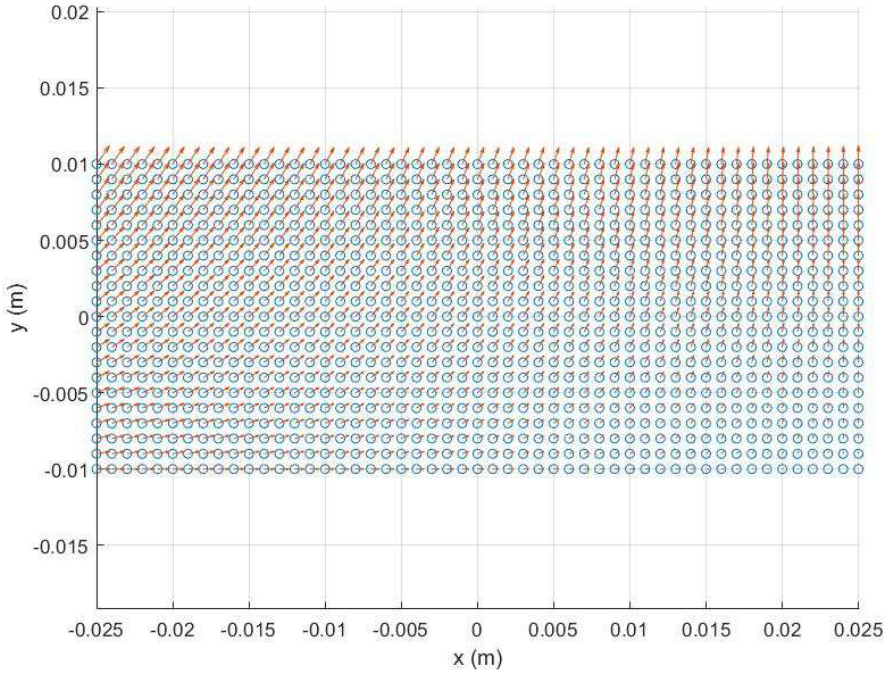
Figura 13 - Comparação da malha antes de depois de sofrer deformação (*scale factor* = $8{,}37.10^3$).



Fonte: Elaborado elo próprio autor.

O gráfico seguinte, da Figura 14, apresenta a malha discretizada em pequenas circunferências de cor azul, com a direção e sentido do deslocamento de cada ponto representada como somatório vetorial na forma de setas vermelhas.

Figura 14 - Representação vetorial da direção dos deslocamentos.

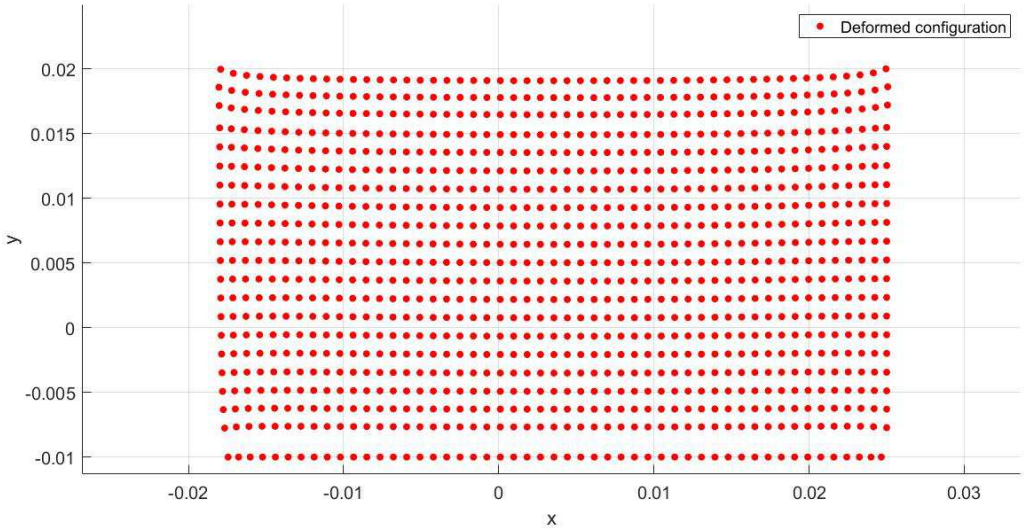


Fonte: Elaborado pelo próprio autor.

É possível observar que as setas vermelhas, que representam a direção dos deslocamentos após a aplicação das tensões, condizem com as condições de fronteira e graus de liberdade impostos para a simulação.

A Figura 15 a seguir, como o próprio nome informa, reresenta a configuração deformada da malha após sofrer as tensões de tração. A plotagem é feita por pontos vermelhos dispersos e novamente foi utilizado pelo autor original do programa (Patriota, 2018) um fator de escala de aproximadamente $8{,}37.10^3$ para destacar a deformação. Neste gráfico também é possível observar a ação dos efeitos de superfície, principamente na região superior, com as duas “pontas” nos extremos do eixo x.

Figura 15 - Representação em pontos da deformação para o *quasi-static solver* (*scale factor* = 8,37.10^3).

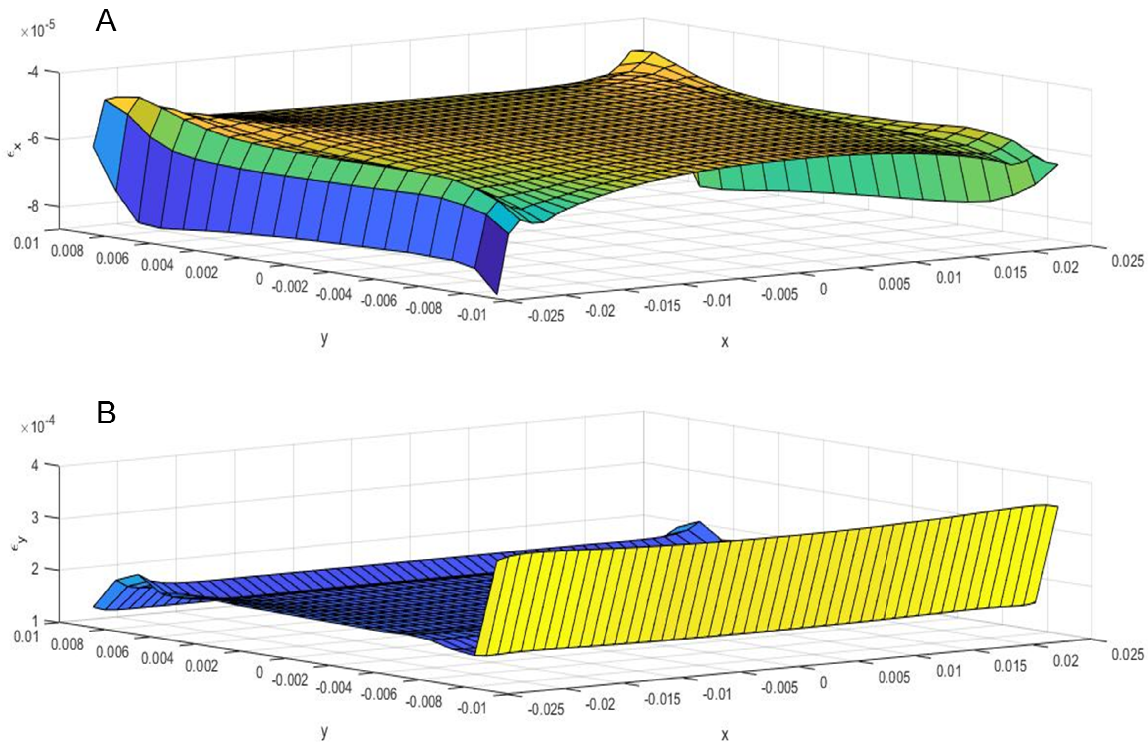


Fonte: Elaborado pelo próprio autor.

Os próximos gráficos (A e B), contidos na Figura 16, mostram a deformação adimensional da malha em relação aos eixos cartesianos. No gráfico superior A, referente a deformação em x (ϵ_x), é possível observar que os maiores valores de deformação nesta direção (em módulo), estão perto das extremidades laterais da malha. Neste mesmo gráfico, os menores valores de deformação em módulo estão na região mais central como um todo, quase de forma constante.

No outro gráfico da Figura 16 (B), observa-se que a deformação na direção y (ϵ_y) começa com seus maiores valores na extremidade inferior da malha, na região negativa do eixo y, vai diminuindo de forma bem acentuada até por volta de y≅ -7 mm e então se mantém constante. Em y ≅ 5 mm ele volta a diminuir e em seguida aumenta, fazendo uma pequena curva e diminuindo novamente até o final da malha.

Figura 16 - Deformação em relação aos eixos para simulação *quasi-static solver.*

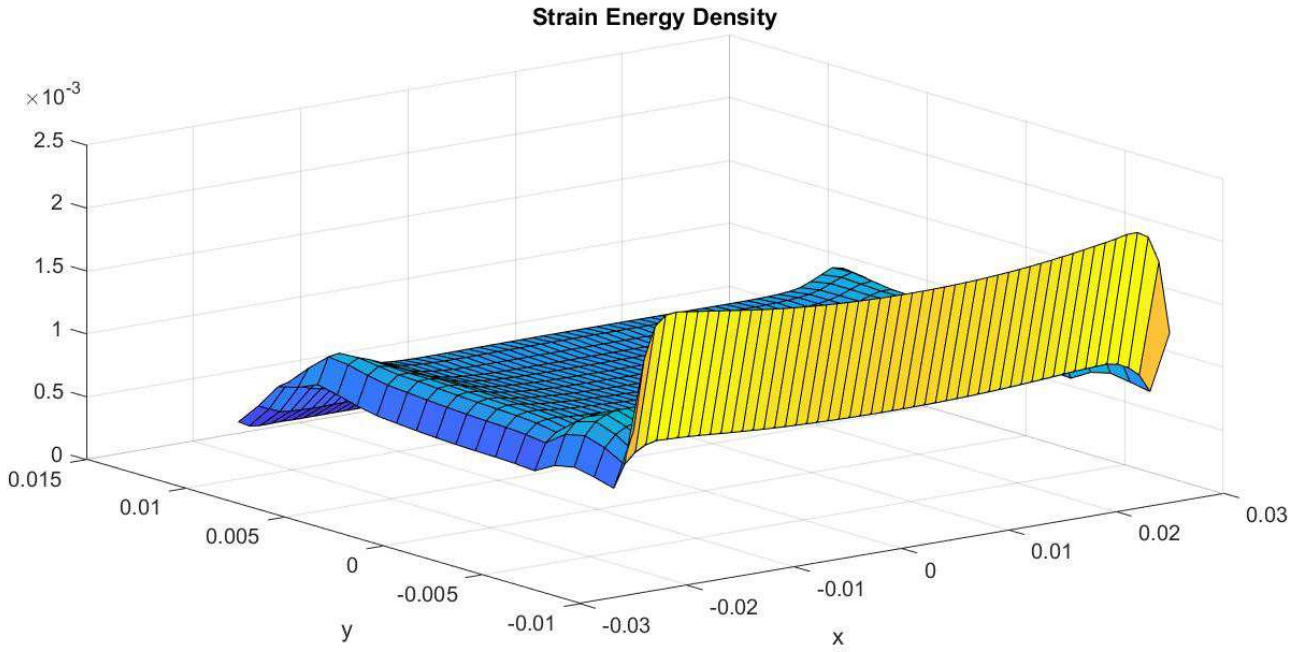


Fonte: Elaborado pelo próprio autor.

Na sequência, com a Figura 17, tem-se a representação gráfica da densidade de energia de deformação da malha inteira em relação aos eixos cartesianos. A primeira coisa interessante a se notar é que este gráfico assemelha-se à uma junção dos dois gráficos anteriores, presentes na Figura 16 e que indicam a deformação por eixo cartesiano.

Observa-se neste gráfico que os maiores valores da densidade de energia de deformação estão na parte inferior da malha, seguido por uma queda acentuada. O resultado é condizente pois, como pode-se observar por outros resultados em outros gráficos, as duas primeiras fileiras (horizontal) de pontos discretizados do domínio na parte inferior apresentam os maiores deslocamentos entre si, e o cálculo da energia de deformação é diretamente proporcional aos deslocamentos.

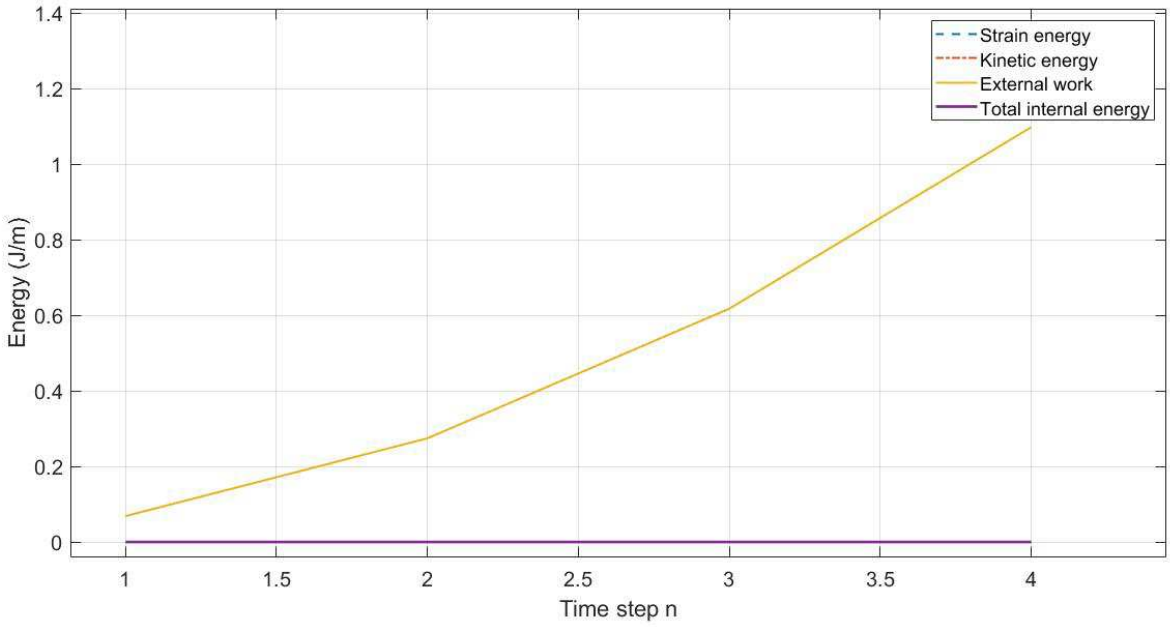
Figura 17 - Densidade de energia de deformação para *quasi-static solver.*



Fonte: Elaborado pelo próprio autor.

O último gráfico dessa simulação simples utilizando o *quasi-static solver* está presente na Figura 18 e apresenta o balanço de energia por passo de tempo.

Figura 18 – Curvas de energia para o balanço de energia com simulação utilizando *quasi-static solver.*



Fonte: Elaborado pelo próprio autor.

É importante ressaltar que o *time step,* ou passo de tempo, é escolhido pelo próprio usuário do programa na função *experiment_template*, na seção final denomiada *SOLVER,* obviamente no caso *Quasi-Static.* Outro modo de se denominar o eixo x é por número de etapas de carregamento. Como pode ser observado, o valor escolhido para o passo de tempo foi 4.

Um detalhe importante para este resultado é que a energia interna total se mantém nula durante a simulação ou etapas de carregamento. Isso condiz com a teoria, pois indica que não há dissipação de energia $\left(\frac{dE}{dt} < 0\right)$, o que também implica na inexistência de nucleação e propagação de trincas.

A curva da energia cinética, que esta sobreposta pela linha da energia interna total, também permanece constante e nula, pois a simulação com este *solver* utiliza como condições de contorno o carregamento externo (ao invés de restrição de velocidade) e tem um passo de tempo constante para este mesmo carregamento.

A curva da energia de deformação está sobreposta pela curva de trabalho externo, tendo os mesmos valores. Este resultado faz sentido, pois no balanço de energia total, Equação (26), o trabalho externo tem sinal negativo. Como a energia interna total e a energia cinética são nulas, a energia de deformação e o trabalho externo devem se anular. No passo de tempo n=4, os valores de trabalho externo e energia de deformação são aproximadamente 1,0978 J/m.

4.1.2 *DYNAMIC/EXPLICIT*

Os resultados para a simulação com o *dynamic/explicit solver* foram por escolha própria guiados para alcançar um valor aproximado da energia de deformação pelos gráficos de balanço de energia da simulação com *quasi-static solver.* A escolha de igualar esse resultado foi uma maneira encontrada para chegar a outros resultados semelhantes para ambos os *solvers.*

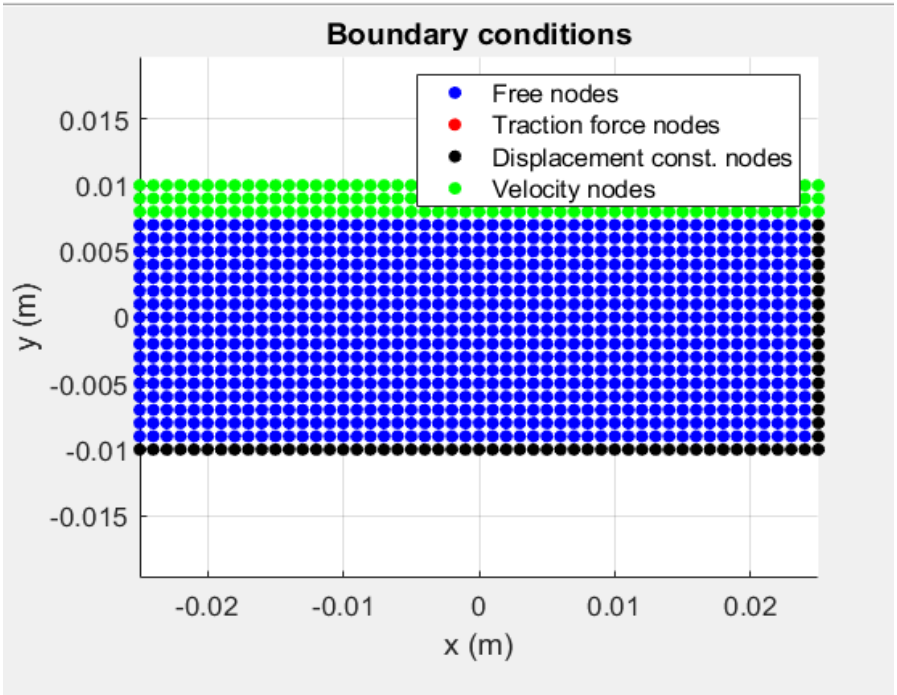
Três pontos muito importantes para uma simulação com este *solver* são o incremento de tempo, o tempo final e a velocidade dos pontos. Para estes resultados um incremento de tempo de 2μs, um tempo final de 30μs e uma velocidade de 117,67mm/s. No programa, os valores de tempo devem ser inseridos pelo usuário na função *experiment_template*, na área referente ao *solver*, enquanto que a velocidade deve ser inserida na função *prescribedBC*, em um vetor

relacionado às restrições de velocidade. O valor de velocidade dos pontos de 117,67mm/s foi escolhido por tentativa e erro para se chegar a um resultado aproximado de balanço de energia à simulação com *quasi-static solver.*

O primeiro gráfico desta simulação (Figura 19), assim como na anterior, mostra as condições de contorno prescritas na malha. Novamente os pontos azuis indicam nós da malha que estão livres, os pontos pretos indicam os que estão restritos e os verdes indicam os nós com velocidade prescrita.

As simulações com este tipo de *solver* levam maior tempo para rodar, no entanto este gráfico aparece bem no início, ajudando na verificação das condições de contorno.

Figura 19 - Condições de contorno para simulação com *dynamic/explicit solver.*

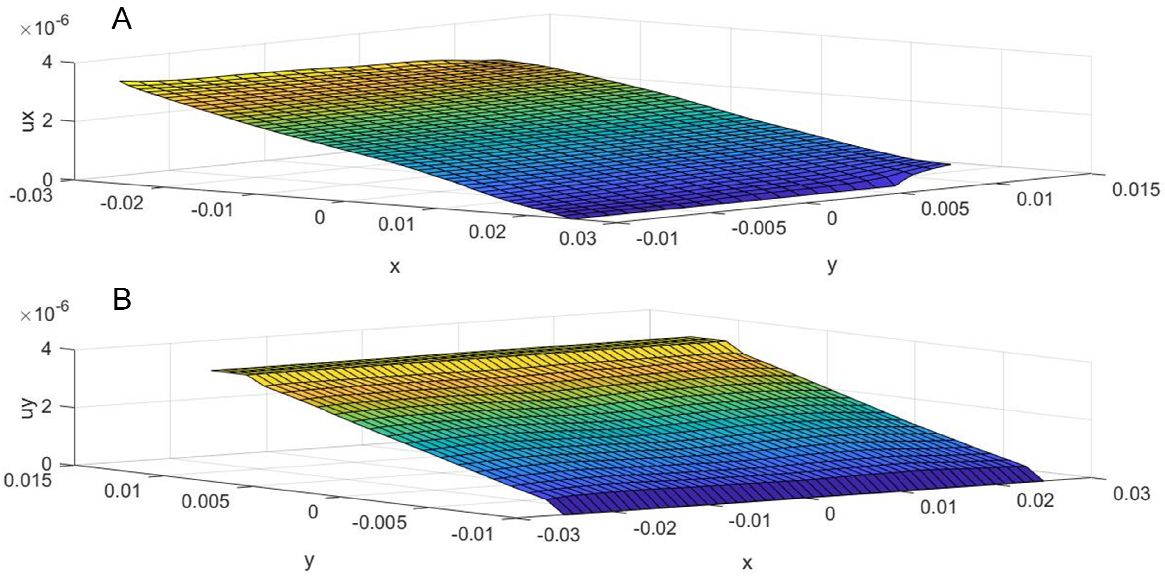


Fonte: Elaborado pelo próprio autor.

Os próximos gráficos para esta simulação, presentes na Figura 20, mostram os campos de deslocamento em relação aos eixos coordenados no domínio da malha. Inicialmente é possível observar que a simulação com o *dynamic/explicit solver* é menos afetada por efeitos de superfície. Para fins práticos, a média do deslocamento máximo no eixo x é de $\bar{u}_x = 3{,}2326.10^{-6}\mathrm{m}$, sendo que o valor diminui gradativamente no sentido y positivo para y negativo. No eixo y, o valor de

deslocamente máximo é constante para as três fileiras mais superiores de nós e seu valor é de $\bar{u}_y = 3{,}5301.10^{-6}\mathrm{m}$. Estas últimas três fileiras de nós são justamente as que estão sofrendo a “ação das forças”, ou de modo mais preciso para o *solver,* o deslocamento pela velocidade prescrita.

Figura 20 - Campo de deslocamento por eixo cartesiano para simulação *dynamic/explicit solver.*

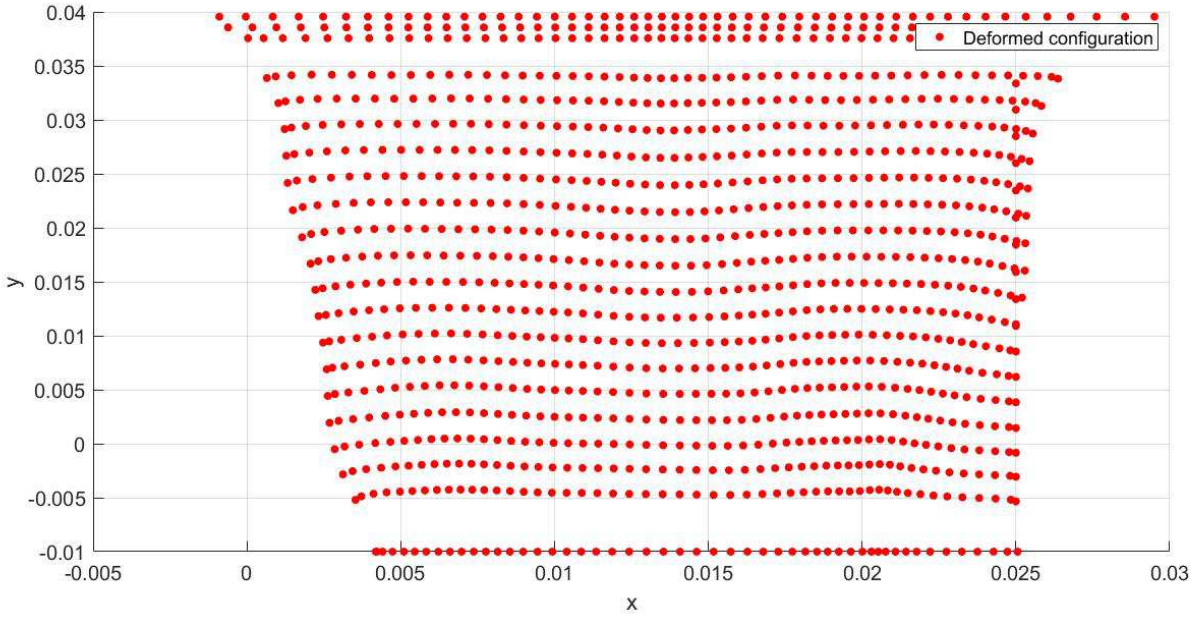


Fonte: Elaborado pelo próprio autor.

O próximo gráfico, Figura 21, assim como na simulação anterior, aprensenta a configuração deformada da malha, mas neste caso após sofrer os deslocamentos ocasionados pelas velocidades prescritas dos nós. A malha também é representada por pontos vermelhos dispersos.

Com esta representação já é possível observar melhor a diferença dos *solvers* para uma mesma configuração de simulação. Mesmo com idênticos graus de liberdade para as restrições de velocidade, forças de corpo e restrições de velocidade, as formas deformadas de ambas as simulações apresentam grandes diferenças.

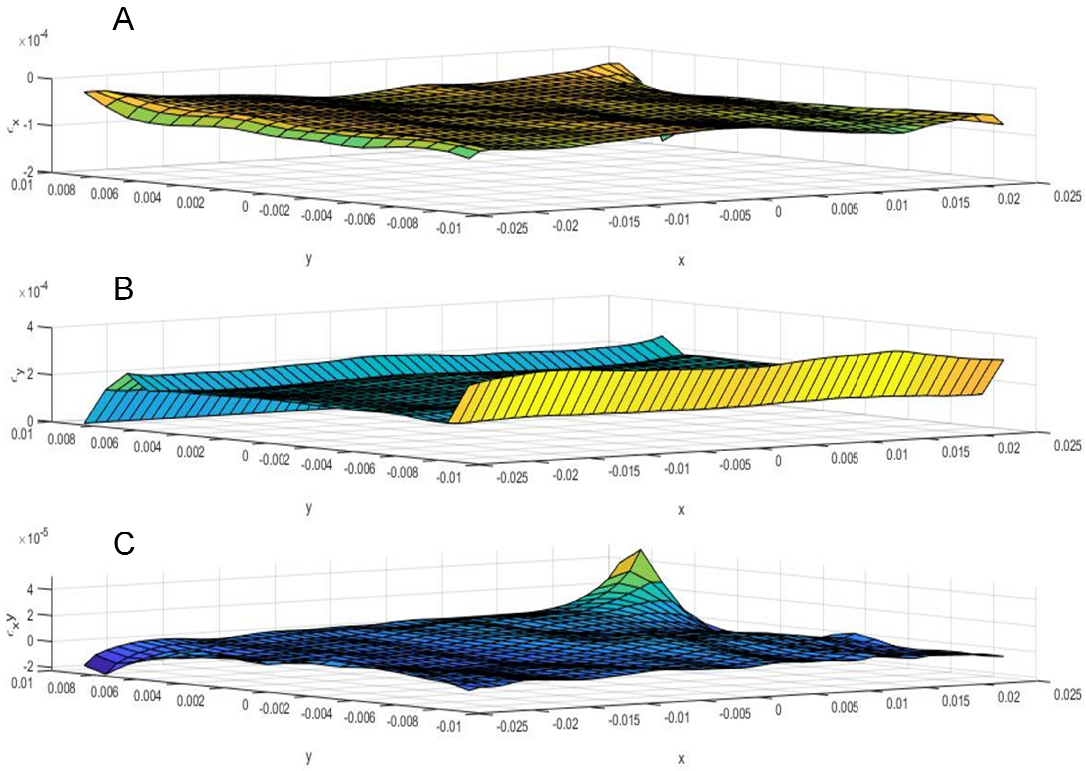
Figura 21 - Representação em pontos da deformação para o *dynamic/explicit solver* (*scale factor* = 8,37.10^3).



Fonte: Elaborada pelo próprio autor.

Na sequência, com a Figura 22, tem-se os gráficos de deformação adimensional (A, B e C) em relação aos eixos cartesianos para esta simulação. Diferente do mesmo conjunto de gráficos para a simulação anterior, com o *dynamic/explicit solver* têm-se um terceiro gráfico C, da deformação de y em relação a x. A adição deste gráfico C foi provavelmente devido ao fato de que simulações com o *dynamic//exlicit solver* apresenta mais deformações e mais “oscilações” nas próprias deformações e/ou deslocamentos. Isto pode ser conseqüência também dos efeitos de superfície.

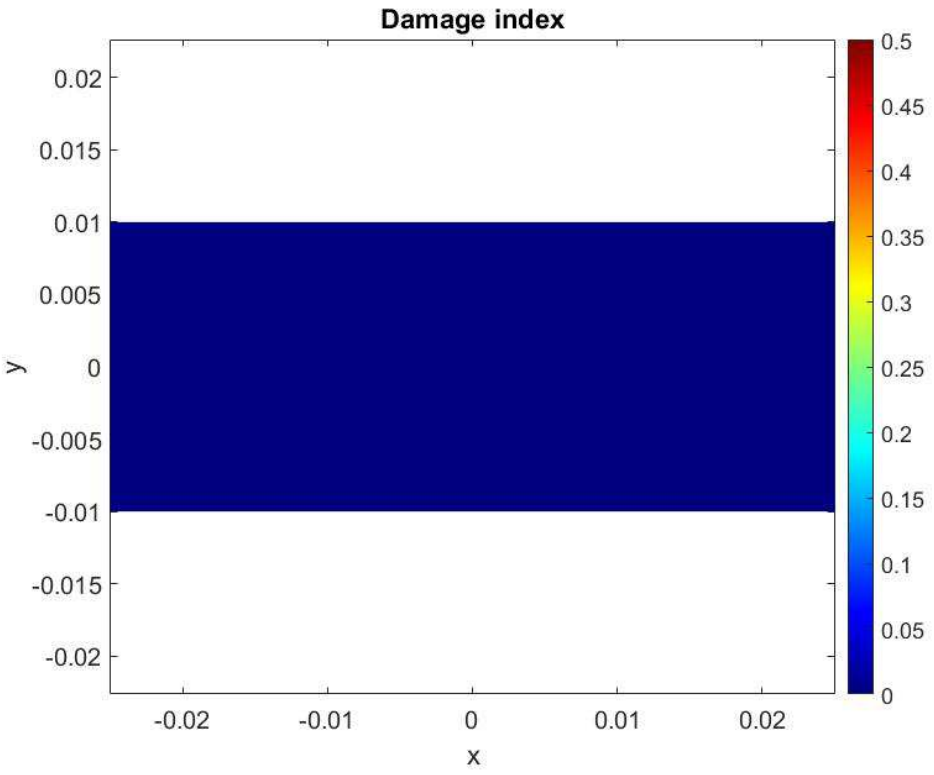
Figura 22 - Deformação em relação aos eixos para simulação *dynamic/explicit solver.*



Fonte: Elaborado pelo próprio autor.

Continuando, a Figura 23 mostra um resultado muito interessante, mas que será melhor explorado na próxima sub-seção, com simulação envolvendo dano inicial. O gráfico desta figura mostra um índice de dano representado por cores e ele só é apresentado em simulações utilizando *dynamic/explicit solver.* Como pode-se observar, nesta simulação não há dano inicial nem nucleação e propagação de trincas.

Figura 23 - Índice de dano.

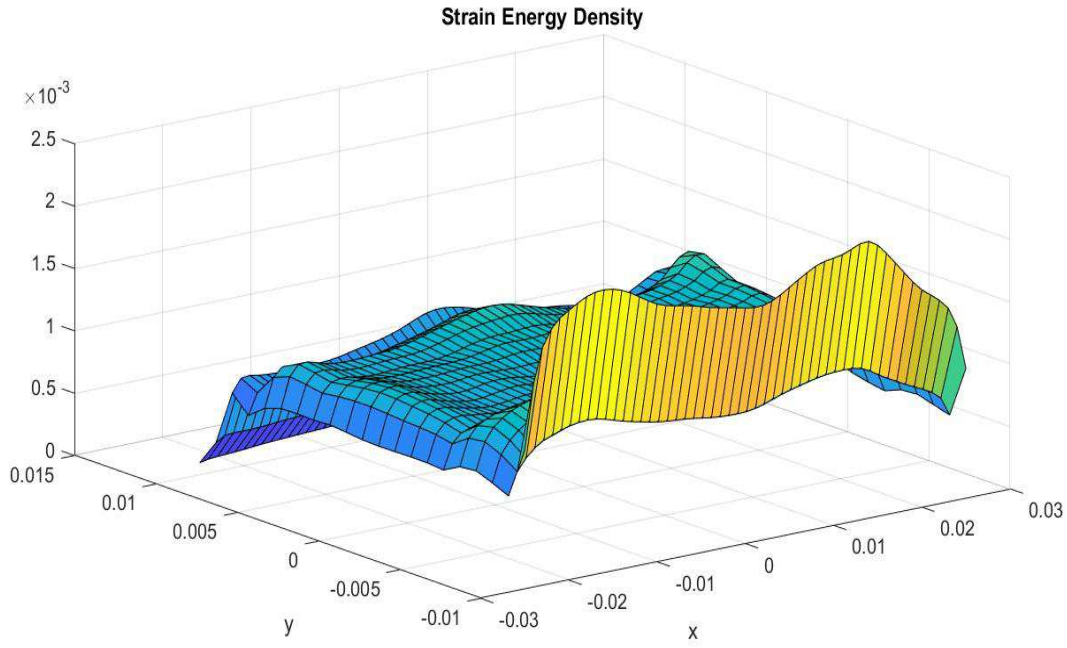


Fonte: Elaborado pelo próprio autor.

Essa simulação também possui um gráfico da densidade de energia de deformação em relação ao domínio ou malha e está na Figura 24 a seguir. Inicialmente é possível observar que a densidade de energia de tensão na malha segue um padrão similar à simulação anterior, utilizando *quasi-static solver,* porém apresentando maior irredularidade no domínio.

Ao analisar-se rapidamente a configuração deformada para esta simulação, no quesito de posição relativa entre os nós, fica nítido que este resultado é compatível com o desejado. Esta afirmação é baseada nos mesmos fundamentos do resultado gráfico da densidade de energia de deformação para o *quasi-static solver,* o cálculo da energia de deformação é diretamente proporcional ao deslocamento relativo entre os pontos.

Figura 24 - Densidade de energia de deformação para *dynamic/explicit solver.*



Fonte: Elaborado pelo próprio autor.

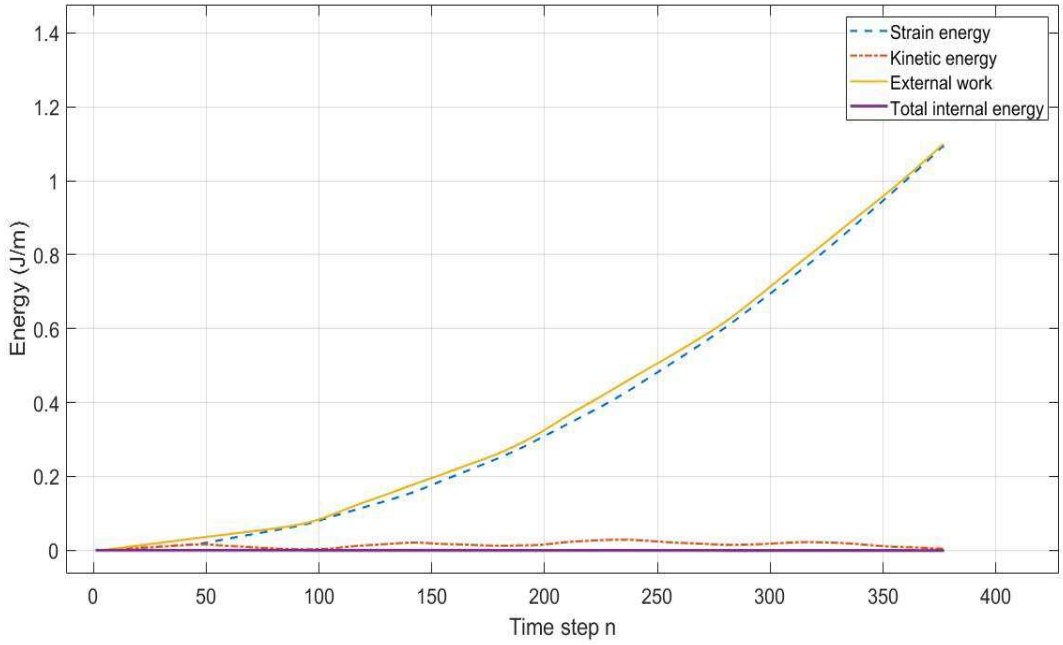
Para fechar os resultados desta simulação, a Figura 25 também apresenta, assim como na simulação anterior, um balanço de energia por passo de tempo. Na utilização do *dynamic/explicit solver,* o incremento de tempo e tempo final escolhidos pelo usuário são transformados no passo de tempo, que para esta simulação foi n = 377. Este passo de tempo é bem maior que o realizado para a simulação com o *solver* anterior, por isso o maior tempo para uma simulação com *dynamic/explicit solver* rodar.

De início oberva-se algumas oscilações nas curvas de energia, em comparação com a simulação com o outro *solver.* Tais oscilações são referentes ao trabalho externo e a energia cinética. Por haver a velocidade prescrita em determinados pontos da malha, a energia cinética não é anulada, mas tende a zero. Como o trabalho externo também é calculado de forma diferente com as restrições de velocidade, também há uma certa oscilação. Sobre este último é válido informar que as velocidades prescritas são transformadas em forças de corpo apenas para o cálculo deste trabalho externo, não para solução do problema numérico.

Um último fator importante é que, assim como na simulação anterior, esta não tem dissipação de energia $\left(\frac{dE}{dt} = 0\right)$, indicando que não há nucleação e propagação

de trinca. Como mencionado no começo desta sub-seção, os valores de energia de deformação e trabalho externo são aproximadamente 1,0978 J/m.

Figura 25 - Balanço de energia para *dynamic/explicit solver.*



Fonte: Elaborado pelo próprio autor.

4.2 PROPAGAÇÃO DE TRINCAS

Com o programa do autor Túlio V. B. Patriota (Patriota, 2018) também é possível realizar simulações com propagação de trincas. O objetivo nesta parte do trabalho foi de analisar os resultados da implementação numérica do programa em relação à propagação de trincas e compará-los com poucos exemplos de experimentos práticos com o mesmo material (*soda-lime glass*).

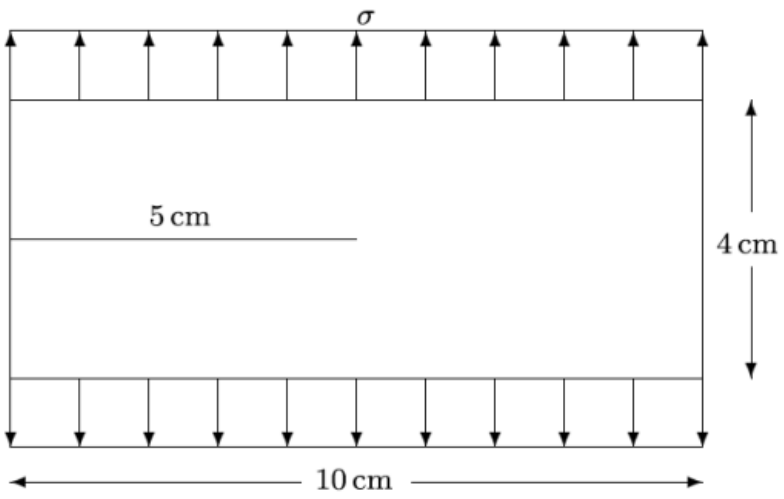
Quanto às comparações, é importante ressaltar que a intenção não foi buscar ou comparar valores numéricos, dada a dificuldade de representar de forma semelhante às condições do experimento. A intenção foi de analisar e comparar superficialmente certos detalhes do crescimento ou propagação de trincas em um material quebradiço, como por exemplo a bifurcação da trinca, simetria, etc.

Para se realizar uma simulação com propagação de trincas com esse programa, são necessários ao usuário se atentar a quatro pontos importantes. O

primeiro é a escolha do *solver*; a propagação de trincas ou rachaduras é algo dinâmico, então a abordagem deve ser com o *dynamic//explicit solver,* caso contrário a solução particular para o caso é igual a zero, aparece uma mensagem de erro. O segundo é acionar o dano inicial; para realizar uma simulação com propagação de trincas, deve-se colocar um dano inicial que precisa ser “acionado” como *true* na variável *damage.damageOn,* na seção *MODEL,* na função *experiment_template.* A terceira é inserir as coordenadas do dano inicial; na função *experiment_template,* dentro da seção *Initial Damage,* deve-se inserir no vetor *damage.crackIn* dois pontos como coordenadas inicial e final no plano cartesiano da malha. Por fim, o quarto ponto é a escolha do modelo constitutivo correto para a simulação; apenas modelos que tenham em suas funções o fator de dano é que vão funcionar para esse tipo de simulação. Relembrando, o modelo é inserido na seção *MODEL*, na variável *model.name* com as próprias iniciais (exemplo: “PMB” = *Prototype micro brittle model*).

Para estas primeiras simulações com trincas, foi utilizada a configuração da Figura 26, que também foi utilizada pelo autor do programa, Túlio V. B. Patriota (Patriota, 2018), para validação da propagação dinâmica de trincas. A representação na Figura 23 possui 4 cm de altura, 10 cm de comprimento e um dano inicial linear de 5 cm que é perpendicular à altura, paralelo ao comprimento e esta situada sobre a linha central do objeto. As tensões são aplicadas como trações em uma camada de nós, tanto na parte superior quanto na inferior.

Figura 26 - Configuração para análise de simulações com trincas.

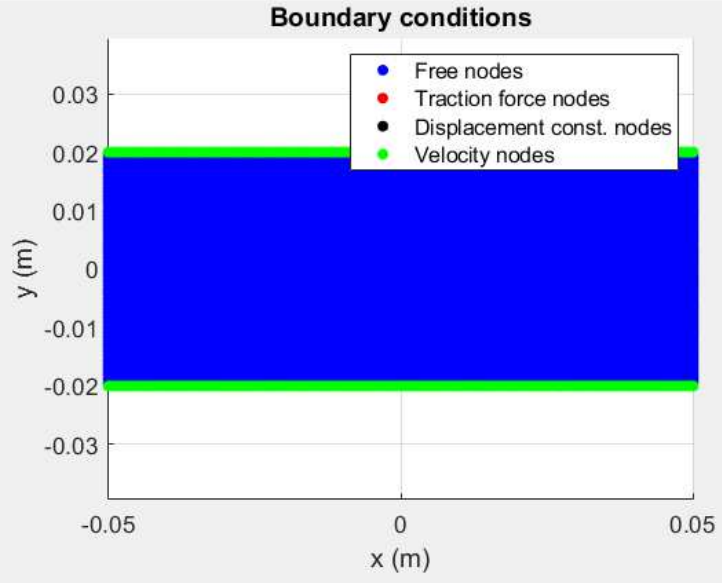


Fonte: T. V. B. Patriota, 2018.

Para essas simulações, o raio do horizonte (δ) é de 2 mm, a razão de malha (d) é 5,5 e o tamanho de lado da célula para cada nó (h) é de aproximadamente $3{,}64.10^{-4}$ m. A partir desses dados é possível afirmar que a malha desse domínio é mais refinada, o que faz inclusive a simulação levar mais tempo para ser completada. Além disso, o incremento de tempo utilizado foi de 2μs e o tempo final de 30μs.

A Figura 27 a seguir mostra as condições de contorno e a malha mais refinada, como mencionado anteriormente. Novamente, como na simulação anterior utilizando o *dynamic/explicit solver,* os pontos azuis indicam nós livres e os pontos verdes indicam os nós com velocidade prescrita. Para esta simulação não foi necessário adicionar restrição de deslocamento nos nós. A uniformidade da representação da configuração em relação as cores já é um indicativo de uma malha mais refinada.

Figura 27 - Condições de contorno para simulação com trinca.



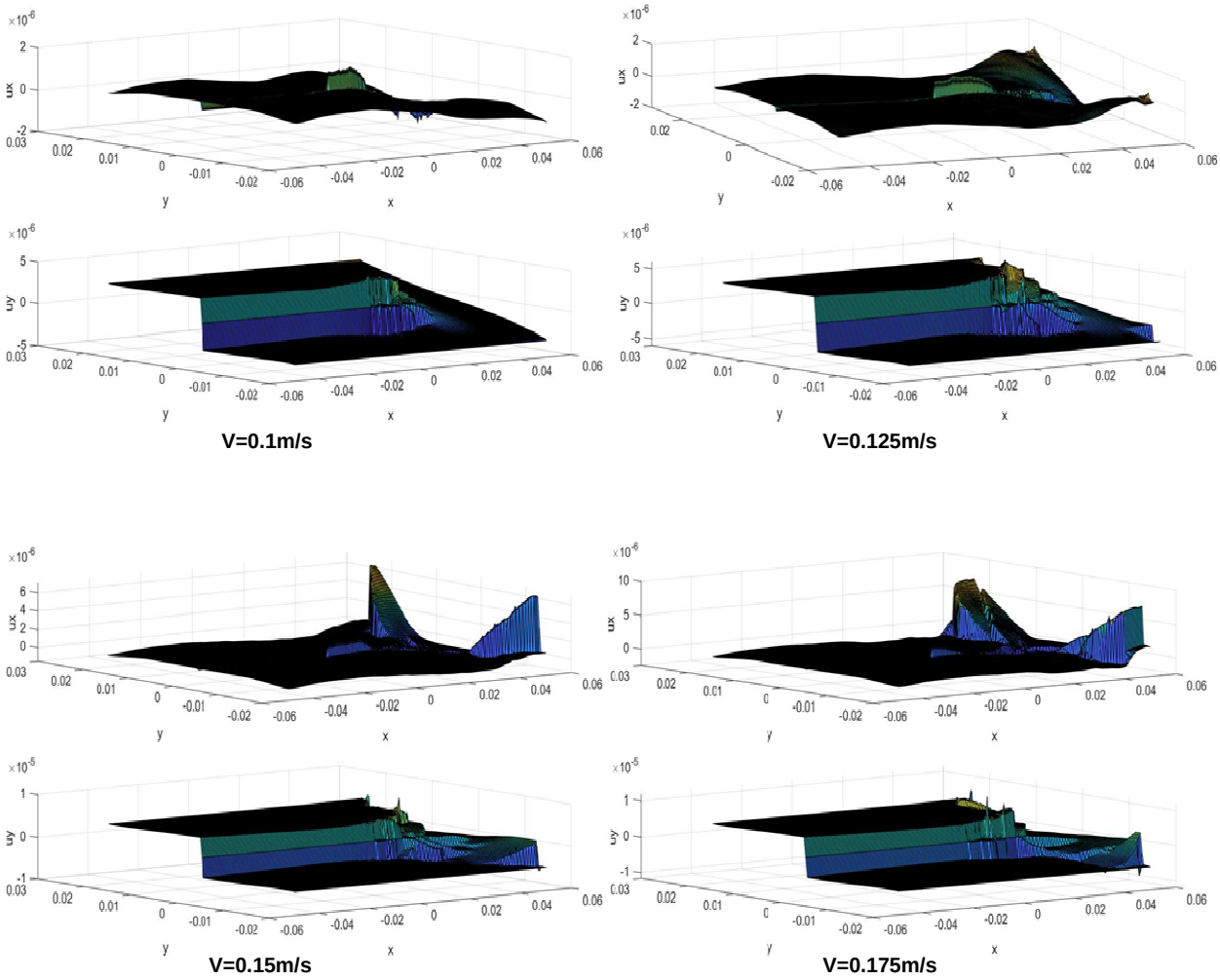
Fonte: Elaborado pelo próprio autor.

Para essas simulações, o único valor diferente utilizado foi a velocidade prescrita nos nós. Foram quatro velocidades diferentes, portanto quatro simulações, com acréscimos de 0,1m/s até 0,175m/s, sendo acréscimos de 0,025m/s. A Figura 28 a seguir mostra, assim como nas simulações anteriores, o campo de deslocamento por eixo cartesiano para simulações com propagação de trinca.

Nestes gráficos de deslocamento, com as diferentes velocidades em crescente, é possível observar uma evolução visual curiosa do campo de deslocamento, em especial em x (u_x). Tal evolução, para ambos os eixos, não é totalmente contínua ou uniforme, justamente pela propagação e ramificação das trincas, que será mostrado nas figuras seguintes.

Em relação ao ao eixo y (u_y), nota-se a partir da linha central horizontal da malha (paralela ao eixo x), um deslocamento em forma de elevação, como se fosse um degrau. No deslocamento em relação ao eixo x (u_x), nota-se o crescimento nas "pontas" para os valores máximos de y positivo e negativo de v = 0,1m/s até v = 0.15 m/s. Na velocidade seguinte, v = 0,175 m/s, nota-se uma distribuição diferente do deslocamento, justamente causado pela propagação e ramificação das trincas.

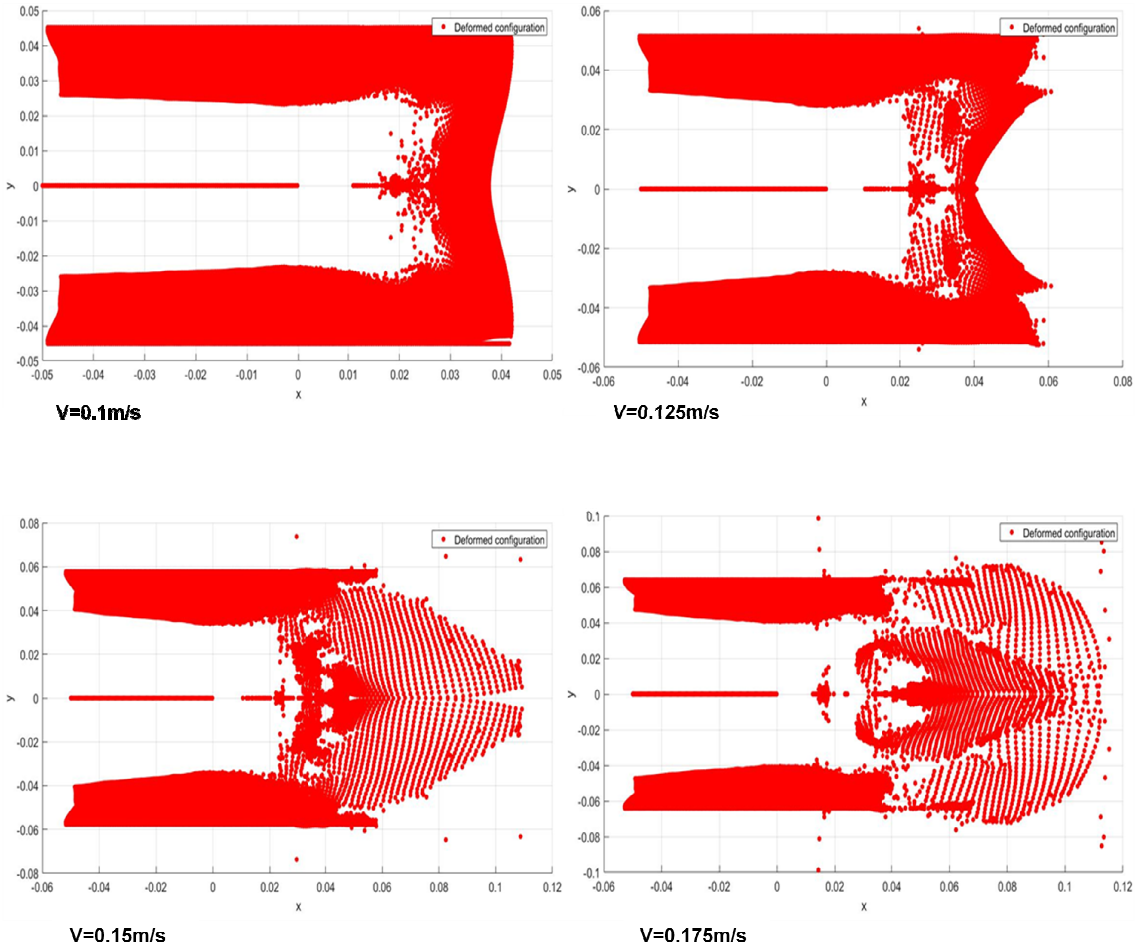
Figura 28 - Campo de deslocamento por eixo cartesiano para simulações com propagação de trinca.



Fonte: Elaborado pelo próprio autor.

Os resultados gráficos da Figura 29 a seguir, referente à configuração deformada após a aplicação das “tensões” (velocidades prescritas), ficam um pouco complexos devido a já mencionada aplicação de um fator de escala para ressaltar a deformação. Apesar da aparência em forma de “mosaico”, é possível notar a maior deformação da malha ou domínio, juntamente com as maiores propagação e ramificação do dano inicial.

Figura 29 - Configuração deformada para simulação com proagaçao de trinca (*scale factor* = 8,37.10^3).

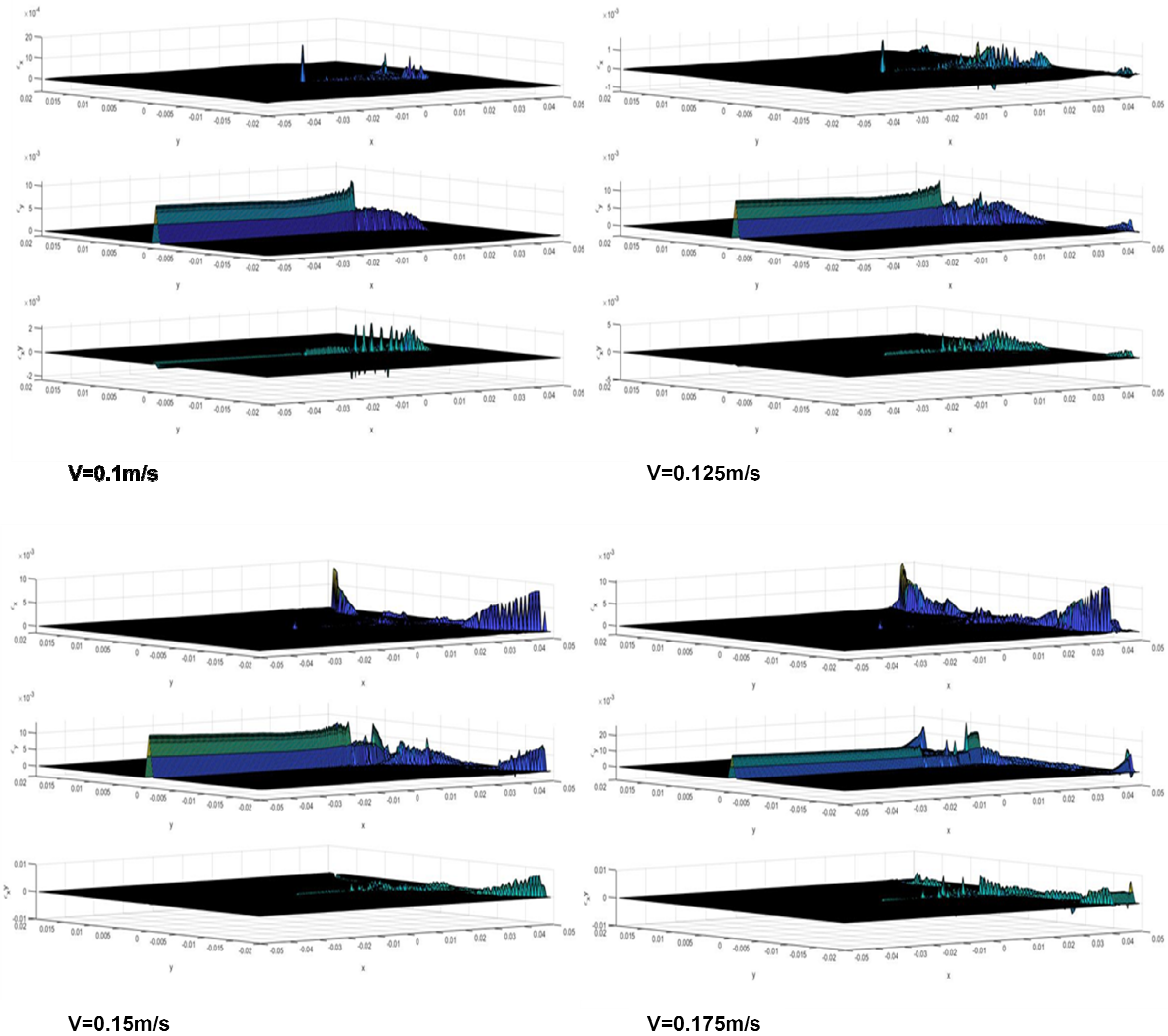


Fonte: Elaborado pelo próprio autor.

Os próximos gráficos, presentes na Figura 30, apresentam as deformações em relação aos eixos cartesianos. Novamente, como apresentado na simulação anterior com *dynamic/explicit solver*, o pós-processamento faz a plotagem de três gráficos relativos a ϵ_x, ϵ_y e ϵ_{xy}.

É interessante notar, principalmente no referente a ϵ_y, a elevação crescente e uniforme em forma de barbatana de x < 0 e y = 0 até (x,y) = (0,0). Estas coordenadas são as do dano inicial, inserido antes da simulação começar, e que são deslocados após a aplicação das trações. A partir de x > 0 essa deformação continua mas não é mais uniforme, pois a partir daí começa a propagação da trinca.

Figura 30 - Deformação em relação aos eixos cartesianos para simulação com trincas.

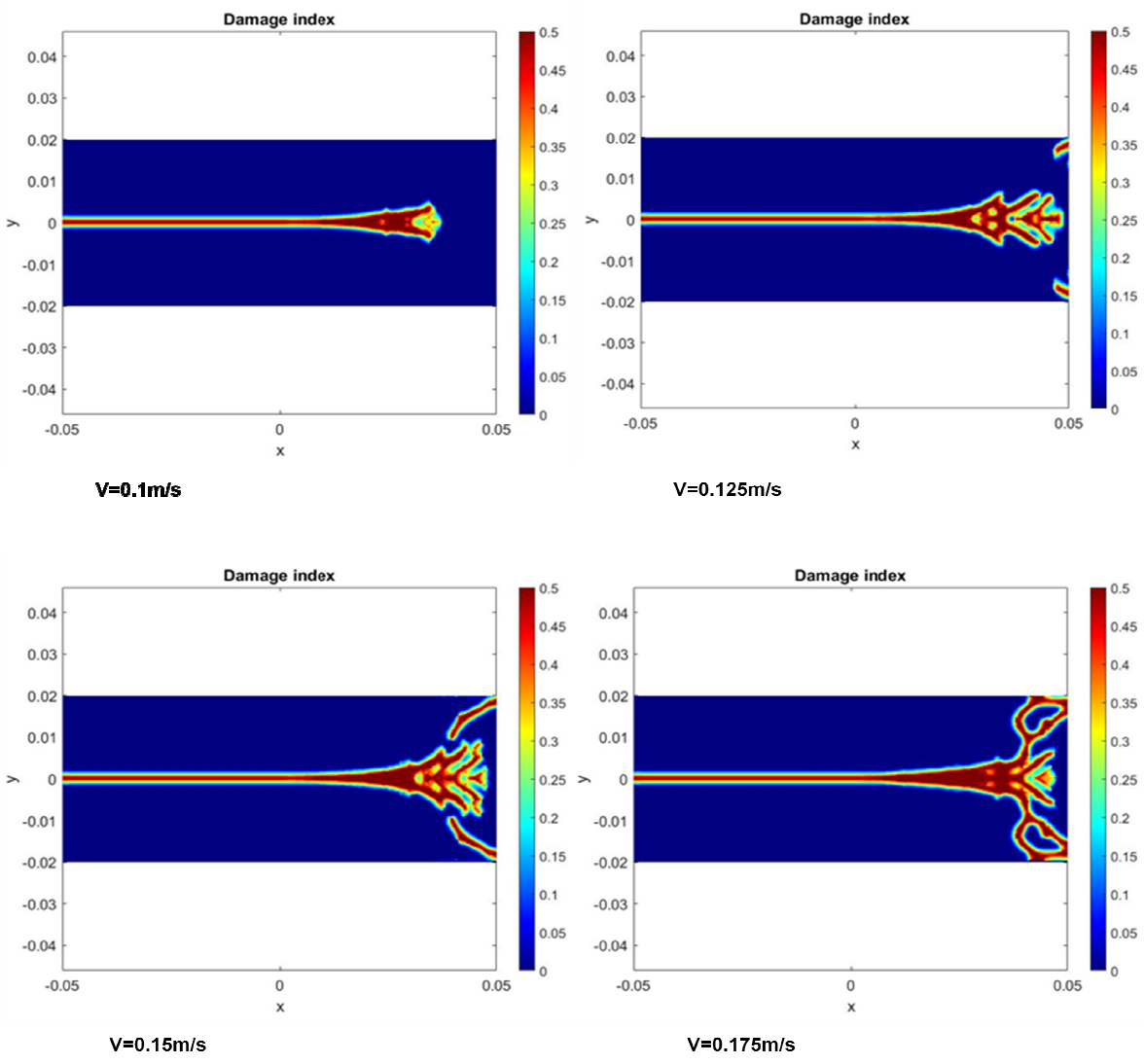


Fonte: Elaborado pelo próprio autor.

A próxima figura tem os resultados mais importantes e interessantes para esta subseção, que são as propagações e ramificações de trincas. Na Figura 31 é possível observar a evolução do dano inicial com o aumento da velocidade prescrita.

O dano inicial, prescrito antes da simulação começar na função *experiment_template*, começa em $(x_i,y_i) = (-0.05,0)$ e vai até $(x_f,y_f) = (0,0)$. A partir do último ponto, com o começo da simulação e deslocamento dos nós, a trinca começa a se propagar até chegar num ponto em que começa a se ramificar também. É possível também observar que as ramificações não partem apenas do dano inicial, e que quanto maior as velocidades prescritas para os nós em um determinado intervalo de tempo, maior será seu deslocamento, portanto maiores serão sua propagação e ramificações.

Figura 31 - Índice de dano ara simulação com trinca.



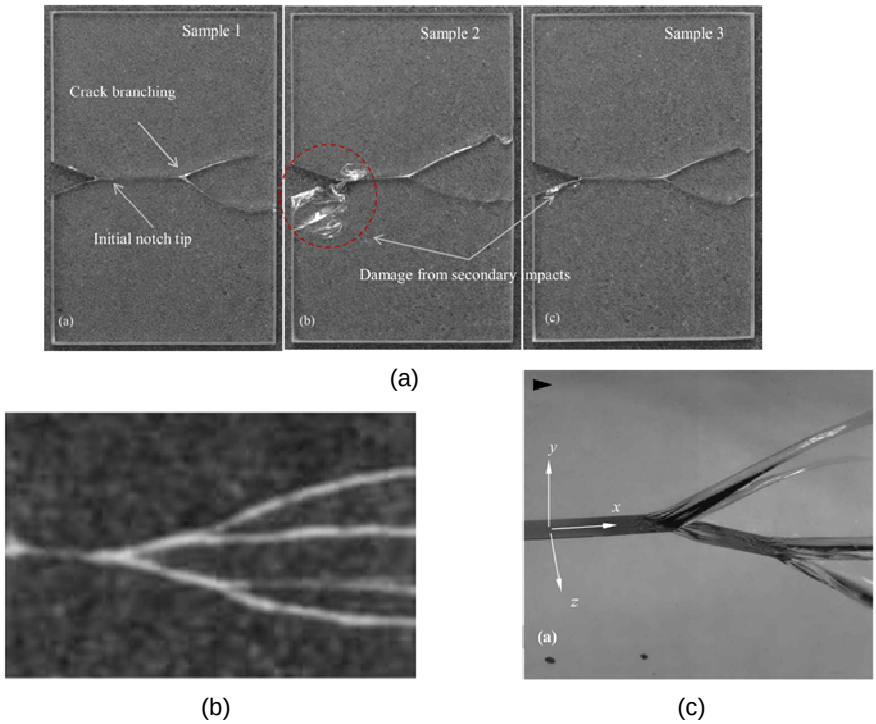
Fonte: Elaborado pelo próprio autor.

Outro resultado importante obtido nesta simulação é a simetria em relação a propagação e ramificação do dano. Observa-se que mesmo ramificações que não partem do dano principal possuem simetria em relação ao eixo x.

A Figura 32 a seguir apresenta fotos de experimentos práticos realizados com placas do material *soda-lime glass* e foram em sua maioria realizados com intuito de analisar a propagação dinâmica de trincas. Como mencionado anteriormente, as simulações com trincas realizadas neste trabalho não foram com intenção de replicar de forma idêntica os experimentos, a adição das fotos de resultados experimentais práticos foi com intuito de uma comparação visual qualitativa.

Os ensaios dos resultados presentes nas fotos da Figura 32 são de autores diferentes mas realizados de forma semelhante. Uma placa de *soda-lime glass* com um entalhe em uma das laterais é fixada em pé (perpendicular à uma superfície). A ação consiste em soltar um pendulo com um peso de formato cônico na ponta diretamente onde está o entalhe na placa. Após o impacto, ocorre a propagação da trinca. Este tipo de ensaio lembra muito o ensaio de impacto charpy.

Figura 32 - Resultados de experimentos práticos.



(a)

(b)

(c)

Fonte: (a) e (c) B. M. Sundaram and H. V. Tippur (2018); (b) T. V. B. Patriota, 2018.

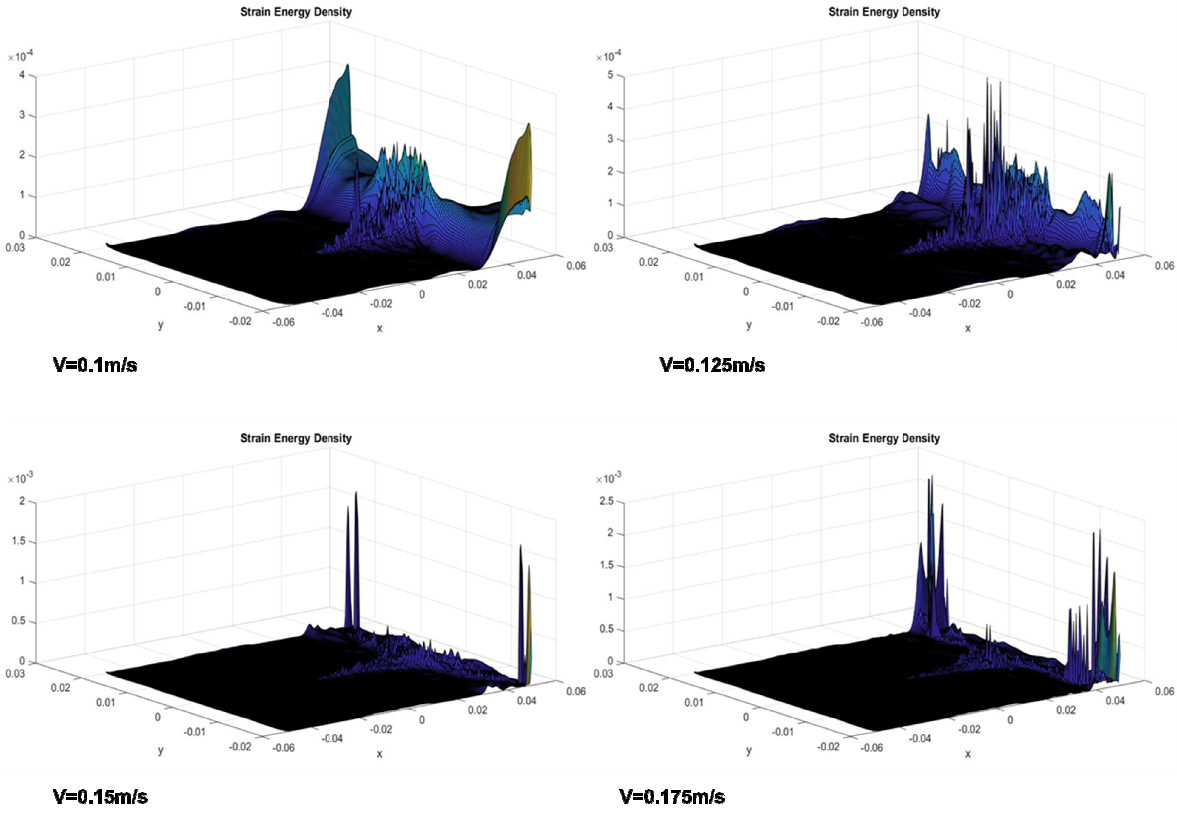
A partir das fotos da Figura 32 acima, é possível observar inicialmente que na prática também há propagação e ramificação de trincas após o material em questão (*soda-lime glass*) sofrer um impacto destrutivo.

Nota-se em (a), na face esquerda, um entalhe inicial onde, após sofrer a ação de um impacto, é prolongado como uma trinca até criar ramificações. As fotos (b) e (c) mostram resultados desse tipo de experimento mais de perto. Na realidade não são totalmente simétricos, mas é importante analisar o comportamento de propagação e ramificação. Percebe-se principalmente em (b) ramos primários e secundários.

Comparando a Figura 32 dos experimentos com a anterior, contendo o índice de danos, nota-se a diferença principalmente em relação ao formato das ramificações e seu lugar de nascimento. Existem inúmeras possibilidades para essa diferença, a maioria referentes às escolhas de configuração para a simulação, como por exemplo as condições de contorno, os parâmetros peridinâmicos, o modelo constitutivo utilizado, o intervalo de tempo, a capacidade computacional da máquina, etc. Em contrapartida, nota-se que com o programa foi possível obter a propagação da trinca e sua ramificação em ramos primários e secundários .

Os próximos gráficos, presentes na Figura 33, apresentam novamente a densidade de energia de deformação em relação aos eixos cartesianos. Os resultados nesta figura são interessantes pelo fato de que para menores velocidades, e consequentemente menores propagações de trincas, menor é a energia de deformação, porém mais abrangente é sua distribuição pela configuração. A explicação para isso é justamente por conta das ligações ainda presentes entre os nós antes de ocorrer as suas quebras pela propagação e ramificação das trincas. Enquanto não ocorre a ruptura das ligações, os nós ainda “compartilham” forças internas e a energia de deformação, que é função do deslocamento relativo ou elongamento entre os pontos materiais. Por essa razão que no gráfico com a velocidade prescrita mais alta (v = 0,175 m/s), onde há os maiores danos ocasionados pelo crescimento da trinca, há uma menor distribuição da energia de deformação.

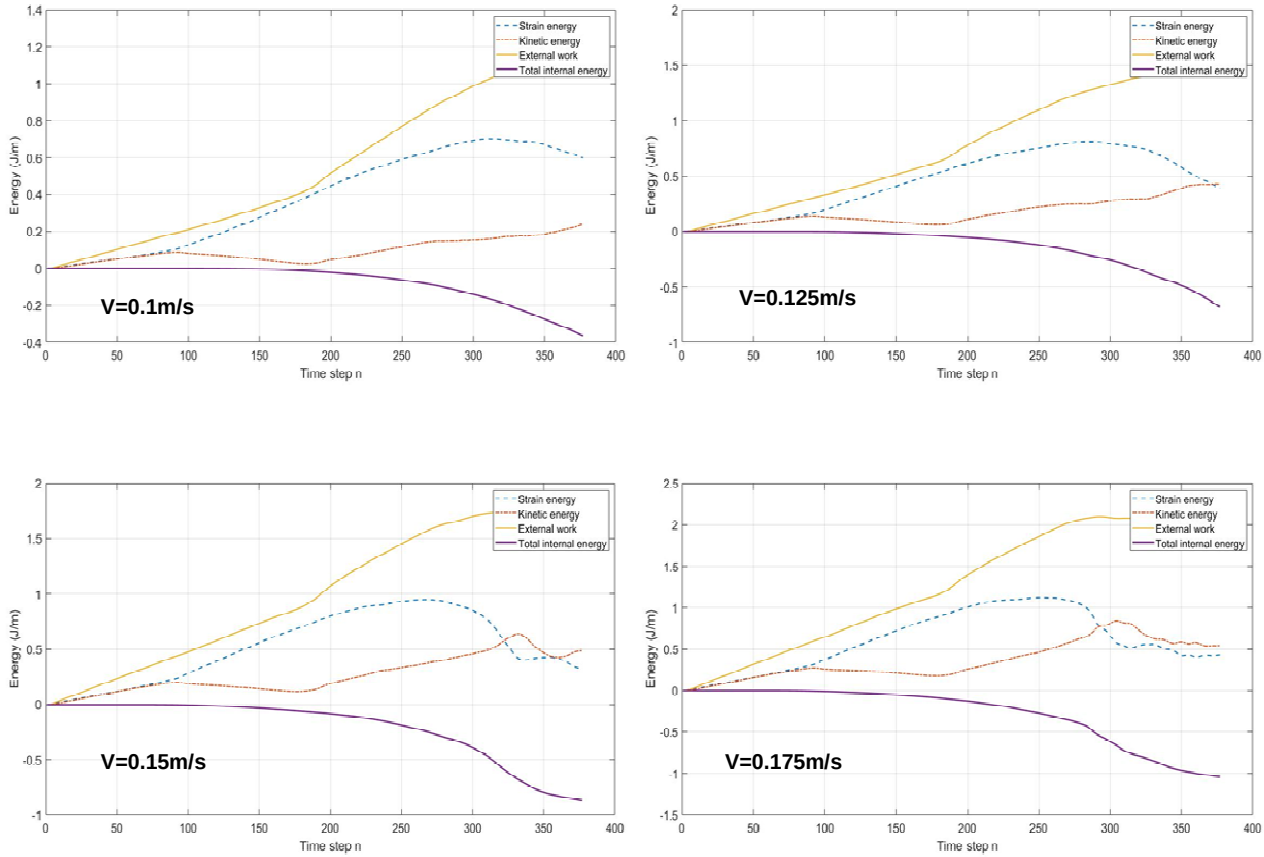
Figura 33 - Densidade de energia de deformação para simulação com trinca.



Fonte: Elaborado pelo próprio autor.

Os últimos gráficos da simulação, presentes na Figura 34, apresentam o balanço de energia por passo de tempo. O primeiro ponto importante a se comentar é a curva roxa, que motra a energia interna total. Como mencionado na seção de implementação numérica, quando este valor torna-se negativo $\left(\frac{dE}{dt} < 0\right)$, existe dissipação de energia, que no caso dessa implementação indica nucleação e propagação de trincas. Isto é comprovado não só por estes últimos gráficos mas por todos os outros presentes nesta subseção.

Figura 34 - Balanço de Energia para simulação com trinca.



Fonte: Elaborado pelo autor.

A seguir uma tabela com valores interessantes sobre essas curvas de balanço de energia. Ela contém para cada simulação os valore de todas as energias (energia interna total E, trabalho externo W^E, energia cinética K e energia de deformação W) em relação ao passo de tempo final e o passo de tempo aproximado onde ocorreu a propagação da falha (n_{falha}). Relembrando que o passo de tempo final para todas as simulações é de 377, pois os valores de incremento de tempo e tempo final, respectivamente 0,02μs e 30μs são os mesmos para todas. O programa do autor converte estes dois últimos valores no passo de tempo n.

Tabela 2 - Valores dos gráficos de balanço de energia.

V[m/s]	E[J/m]	W^E[J/m]	K[J/m]	W[J/m]	n_{falha}
0.1	-0.36293	1.2001	0.23707	0.60006	113
0.125	-0.67851	1.5023	0.42726	0.39649	90
0.15	-0.86835	1.6798	0.48495	0.32653	76
0.175	-1.0419	2.0133	0.54235	0.42901	74

Fonte: Elaborado pelo próprio autor.

É interessante notar que os valores seguem um padrão. No geral W^E, K e W aumentam conforme V aumenta, enquanto que E e n_{falha} diminuem enquanto V aumenta. Os valores de n_{falha}, por mais que sejam uma aproximação bem grosseira, condizem no seu decrescimento. As velocidades prescritas escolhidas nada mais são do que um valor de entrada do usuário para o método ou abordagem do *dynamic/explicit solver*, onde pela implementação numérica será traduzido em uma força ou tensão aplicada. Por isso, quanto maior a velocidade, maior a "tensão", menor o tempo para nucleação e propagação da trinca e maior o dano total.

5. CONCLUSÃO

Reiterando, o objetivo deste trabalho foi analisar a eficácia do método peridinâmico como possível ferramenta para o campo de manutenção preditiva. O principal recurso para tal análise foi um complexo programa computacional escrito no software MATLAB. Sem dúvidas, a implementação numérica em si é de grande qualidade e bem robusta. Ela apresenta resultados muito satisfatórios especialmente nos gráficos oriundos do pós-processamento, dadas as devidas proporções e limitações para simulações em 2D.

Com relação às comparações realizadas entre a utilização dos *solvers quasi-static* e *dynamic/explicit,* verificou-se que ambos os casos são úteis e apresentam resultados satisfatórios. A escolha de se realizar uma simulação utilizando um ou outro vai depender diretamente do problema a ser abordado.

Escolhendo o *quasi-static solver,* é importante ter em mente que para a esse solucionador, a solução estática é parte do estado estacionário da resposta transitória da solução, então basicamente ele servirá idealmente para simulações mais simples, apenas com deslocamento e deformação, sem a intenção de realizar a inserção e propagação de trincas.

Com o *dynamic/explicit*, tanto problemas simples como os mencionados anteriormente, quanto problemas mais complexos envolvendo a propagação dinâmica de trincas podem ser perfeitamente abordados. Há uma pequena diferença nos resultados, no que diz respeito à direção de deslocamento por exemplo, mas são questões que podem ser contornadas, como por exemplo confeccionando uma malha mais refinada, utilizando funções de influência para minimizar efeitos de superfície, alterando intervalo de tempo, etc.

Com relação às simulações envolvendo propagação e ramificação de trincas, obteve-se igualmente resultados satisfatórios, mesmo que não tão semelhantes visualmente aos experimentos práticos de comparação. O fato das ramificações das trincas obtidas serem simétricas para uma configuração também simétrica (trinca inicial contida em um dos eixos da malha, tensões de mesma magnitude mas com sentidos opostos nas partes superior e inferior), indicam também bons resultados. Outro detalhe importante e curioso nos resultados foi que a ramificação das trincas não surge apenas do dano inicial, o que reforça a implementação numérica robusta.

Durante a realização deste trabalho a maior dificuldade encontrada foi em descobrir como o programa funciona e suas possibilidades. Provavelmente com um entendimento mais aprofundado do programa do autor original, Túlio V. B. Patriota (Patriota, 2018) do software MATLAB, juntamente com a exploração de diferentes e mais complexos modelos constitutivos.

Outra questão que também foi uma dificuldade foi de encontrar um valor de energia crítica de deformação, G_0, para materiais diferentes do *soda-lime glass*. Na literatura existem diversas formas de se calcular este valor, porém utilizando funções muito extensas e iterativas. Como a maioria dos exemplos de simulações e experimentos práticos para o tópico de propagação de trincas e/ou peridinâmica utilizam este material como base, o valor em questão já existe tabelado. Por essas razões foi utilizado o mesmo material neste trabalho, o *soda-lime glass.*

Por fim, com base em todos os resultados obtidos, justifica-se que o método peridinâmico é uma ferramenta de manutenção preditiva muito promissora para a engenharia e a indústria.

REFERÊNCIAS

ABECOM. Entenda o que é manutenção preventiva e quando deve ser realizada, 22 jul. 2021. Disponível em: <https://www.abecom.com.br/o-que-e-manutencao-preventiva/> . Acesso em: 10 dez. 2021.

A. C. Gonçalves and M. A. Bazani, Utilização da Peridinâmica como ferramenta de manutenção preditiva para acompanhamento de propagação de trincas - Faculdade de Engenharia de Ilha Solteira, Departamento de Engenharia Mecânica, Unesp, Ilha Solteira, 2018.

A. F. M. Azevedo, “Método dos Elementos Finitos”, 2003, Faculdade de Engenharia da Universidade do Porto.

B. M. Sundaram and H. V. Tippur, Dynamic fracture of soda-lime glass: A full-field optical investigation of crack initiation, propagation and branching, Journal of the Mechanics and Physics of Solids, Department of Mechanical Engineering, Auburn University, US, 2018.

IClass. Manutenção Preditiva, Preventiva e Corretiva. Disponível em: https://www.iclass.com.br/blog/manutencao-preditiva-preventiva-e-corretiva/. Acesso em: 20 abr. 2022.

I. Marinelli. Diferenças entre manutenção corretiva, preventiva e preditiva: guia definitivo, 2021. Disponível em: <https://tractian.com/blog/diferencas-entre-manutencao-corretiva-preventiva-e-preditiva-guia-definitivo-2021/>. Acesso em: 24 fev. 2022

E. Madenci and E. Oterkus, Peridybamic Theory and Its Applications, Springer, New York, 2014.

J. P. Dias, M.A. Bazani, A.T. Paschoalini, L. Barbanti, A Review of Crack Propagation Modeling using Peridynamics, 2017, Springer, New York.

J. Teles. MANUTENÇÃO CORRETIVA – COMO E QUANTO USAR, 16 jul. 2018. Disponível em: <https://engeteles.com.br/manutencao-corretiva/#:~:text=Segundo%20a%20Norma%20NBR%2D5462,mais%20preju%C3%ADzo%20para%20a%20empresa>. Acesso em: 24 fev. 2022.

J. Teles. Manutenção Preditiva: O que é e como ela pode te ajudar!, 4 abr. 2017. Disponível em: https://engeteles.com.br/manutencao-preditiva. Acesso em: 25 jun. 2021.

S. A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, Journal of the Mechanics and Physics of Solids 48.1(2000), 175-209.

T. V. B. Patriota, Numerical investigation of damage models in two-dimensional peridynamics using MATLAB. 2018. 174 f. Tese (Mestrado de ciências em Engenharia Mecânica) – Politecnico di Milano, School of Industrial and Information Engineering, 2018.

ANEXO A – Algoritmo para *Velocity-Verlet scheme.*

Algoritmo 1: Esquema *Velociy-Verlet*

1: **procedimento**: Atualizar o deslocamento (u_i^{n+1}) no próximo passo de tempo

2: $v_i^{n+1/2} \leftarrow v_i^n + \frac{\Delta t}{2} a_i^n$

3: $u_i^{n+1} \leftarrow u_i^n + \Delta t v_i^{n+1/2}$

4: $a_i^{n+1} \leftarrow \frac{1}{\rho_i}\left(\sum_{j \in \mathcal{F}_i} f_{ij}^{n+1} \Delta V_{ij} + b_i^{n+1}\right)$

5: $v_i^{n+1} \leftarrow v_i^{n+1/2} + \frac{\Delta t}{2} a_i^{n+1}$

6: **fim de procedimento**

ANEXO B – Algoritmo para *quase-static solver.*

Algoritmo 2: *Quase-Static solver*

procedimento: Estimando a solução de passo de carregamento $(\bar{\mathrm{u}}^n)$
    Passo 1: Iniciando o vetor de deslocamento de grau de liberdade
$\bar{\mathrm{u}}^0 = 0;$
    para cada passo de carregamento n de 1 até n_{tot} **faça**:
        Passo 2: carregar as condições de carregamento e a hipótese de solução
        $\bar{\mathrm{u}}^n \leftarrow \bar{\mathrm{u}}^{n-1}$
        Passo 3: calcular o vetor residual, $\mathbf{r}$, e o residual r.
        Determinar o critério de convergência para o passo de carregamento.
        $\mathbf{r} \leftarrow \bar{\mathrm{f}}^n_{\mathrm{model}} + \bar{\mathrm{b}}^n\bar{\mathrm{V}}^T$
        $r \leftarrow \|\mathbf{r}\|_\infty$
        $r_\varepsilon \leftarrow \varepsilon \max\left\{\left\|\bar{\mathrm{f}}^n_{\mathrm{model}}\right\|_\infty, \left\|\bar{\mathrm{b}}^n\bar{\mathrm{V}}^T\right\|_\infty\right\}$
        enquanto $r > r_{max}$ **faça:**
            Passo 4: calcular a matriz de rigidez tangente
            $\mathbf{K}^n \leftarrow \mathrm{tangentStiffness}(\bar{\mathrm{u}}^n)$
            Passo 5: calcular a solução incremental e atualizar a hipótese
            $\Delta\bar{\mathrm{u}} = -(K^n)^{-1}\mathbf{r}$
            $\bar{\mathrm{u}}^n = \bar{\mathrm{u}}^n + \Delta\bar{\mathrm{u}}$
            $\mathbf{r} \leftarrow \bar{\mathrm{f}}^n_{\mathrm{model}} + \bar{\mathrm{b}}^n\bar{\mathrm{V}}^T$
            $r \leftarrow \|\mathbf{r}\|_\infty$
        fim enquanto
    fim para
fim procedimento