

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA
CÂMPUS DE ILHA SOLTEIRA**

GABRIEL CESTARO MUNHOZ

**APLICAÇÃO DE CÓDIGO PERIDINÂMICO COM MODELO CONSTITUTIVO
PMB DTT EM PYTHON.**

**Ilha Solteira
2023**



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

GABRIEL CESTARO MUNHOZ

**APLICAÇÃO DE CÓDIGO PERIDINÂMICO COM MODELO CONSTITUTIVO
PMB DTT EM PYTHON.**

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
2023

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

M966a Munhoz, Gabriel Cestaro.
Aplicação de código peridinâmico com modelo constitutivo PMB DTT em Python / Gabriel Cestaro Munhoz. -- Ilha Solteira: [s.n.], 2023
67 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Mecânica) -
Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2023

Orientador: Márcio Antônio Bazani

Inclui bibliografia

1. Peridinâmica. 2. Teoria não local. 3. Simulação. 4. Implementação numérica. 5. Python.

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

CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

ATA DE DEFESA – TRABALHO DE GRADUAÇÃO

TÍTULO: APLICAÇÃO DE CÓDIGO PERIDINÂMICO COM MODELO
CONSTITUTIVO PMB DTT EM PYTHON

ALUNO: Gabriel Cestaro Munhoz

R.A.: 151052387

ORIENTADOR: Prof. Dr. Márcio Antônio Bazani

APROVADO (X) – REPROVADO () PELA COMISSÃO EXAMINADORA

Nota obtida: 9,0

COMISSÃO EXAMINADORA:

Prof. Dr. Márcio Antônio Bazani
Presidente/Orientador

Felipe Perroni Chaim

Eng. Felipe Perroni Chaim

Davi Moraes Monelli

Eng. Davi Moraes Monelli

Gabriel Cestaro Munhoz

Gabriel Cestaro Munhoz

Ilha Solteira/SP, 14 de junho de 2023

Dedicatória

Dedico este trabalho de graduação à minha família que sempre me apoiou em minhas escolhas na vida e também ao professor Luciano Barbanti (in memoriam) que foi o idealizador da Peridinâmica na Unesp.

Agradecimentos

Gostaria de agradecer à minha família, por todo o apoio incondicional, ao meu professor orientador Dr. Márcio Antonio Bazani pela paciência, orientações e ensinamentos transmitidos ao longo da graduação e nesse trabalho e a todos os meus amigos feitos ao longo dessa trajetória.

“Quanto mais me aprofundo na Ciência mais me
aproximo de Deus.”

(Albert Einstein)

Resumo

Este trabalho realiza uma aplicação do método peridinâmico como uma alternativa de estudo estrutural à Metodologia dos Elementos Finitos. O trabalho inicialmente apresenta uma introdução contendo as características gerais da teoria local, metodologia dos elementos finitos, teoria não local e a teoria peridinâmica. Em seguida faz-se uma revisão teórica da teoria da peridinâmica com os princípios básicos, como horizonte, posição relativa, peridinâmica *bond-based* e dano. A metodologia utilizada para a aplicação foi simulação de uma placa, o primeiro caso é aplicação de tensão de tração na placa e o segundo caso é aplicação da tensão de tração na placa com inserção de uma trinca inicial. Para a implementação numérica, foi utilizado um programa escrito na linguagem de programação *open source* Python, o *solver* utilizado para ambos os casos foi o *quasi-static*. Nesse trabalho também foi implementado o modelo constitutivo Modified *Prototype Micro Brittle (PMB DTT)* e também a correção do balanço de energia. Foi possível observar que a aplicação da ferramenta possui grandes desafios devido a seu desenvolvimento recente, porém os resultados foram satisfatórios e a ferramenta possui grande potencial para o estudo de deformações e deslocamentos de sólidos.

Palavras-chave: peridinâmica, teoria não local, simulação, implementação numérica, python.

Abstract

This work applies the peridynamic method as an alternative for structural study to the Finite Element Method. The work initially presents an introduction containing the general characteristics of local theory, the Finite Element Method, non-local theory, and peridynamic theory. Then, a theoretical review of peridynamic theory is presented, including basic principles such as horizon, relative position, bond-based peridynamics, and damage. The methodology used for the application involved the simulation of a plate, with the first case being the application of tensile stress on the plate, and the second case being the application of tensile stress on the plate with the insertion of an initial crack. For the numerical implementation, a program written in the open-source programming language Python was used, and the solver used for both cases was the quasi-static solver. In this work, the Modified Prototype Micro Brittle (PMB DTT) constitutive model was also implemented, and the energy balance was corrected. It was observed that the application of this tool has great challenges due to its recent development, but the results were satisfactory, and the tool has great potential for the study of deformations and displacements of solids.

Keywords: peridynamics, non local theory, simulation, numeric implementation, python.

Lista de Figuras

Figura 1 – Campo de tensões principais em uma malha	14
Figura 2 – Relação entre modelos contínuo local e não-local	15
Figura 3 – Interação dos pontos materiais $x_{(k)}$ e $x_{(i)}$	19
Figura 4 – Pontos materiais peridinâmicos após deformação	20
Figura 5 – Modelo PMB com dano	23
Figura 6 – Força de interação entre pontos e fator de dano para o modelo	25
Figura 7 – Fluxograma para execução do programa em <i>python</i>	26
Figura 8 – Esquema de domínio peridinâmico discretizado	29
Figura 9 – Domínio discretizado com vizinhança e classes de pontos	37
Figura 10 – Família gerada pelo algoritmo FA	37
Figura 11 – Família gerada pelo algoritmo PA-AC	38
Figura 12 – Malha discretizada da placa	40
Figura 13 – Comparação de famílias geradas pelos algoritmos PA-AC e FA	40
Figura 14 – Condições de contorno aplicado a malha sem trinca inicial	41
Figura 15 – Campo de deslocamento da malha sem trinca inicial	42
Figura 16 – Deslocamento da malha sem trinca inicial devido a carga	44
Figura 17 – Representação vetorial dos deslocamentos da malha sem trinca inicial	45
Figura 18 – Gráfico de deformação da malha sem trinca inicial devido a carga de tração	46
Figura 19 – Densidade de energia de deformação ao longo da malha sem trinca inicial	47
Figura 20 – Evolução da energia para o caso sem trinca inicial	48
Figura 21 – Condições de contorno aplicadas a malha com a trinca inicial	49
Figura 22 – Campo de deslocamento da malha com trinca inicial	49
Figura 23 – Deslocamento da malha com trinca inicial	51
Figura 24 – Representação vetorial do deslocamento da malha com a trinca inicial	52
Figura 25 – Campo de deformação da malha com trinca inicial	53
Figura 26 – Campo de densidade de energia de deformação ao longo da malha com trinca	54
Figura 27 – Evolução da energia para o caso com trinca inicial	55
Figura 28 – Índice dano ao longo da malha	56

Sumário

1 INTRODUÇÃO	12
1.1 Objetivos	12
1.2 Considerações gerais	12
2 FUNDAMENTOS TEÓRICOS	18
2.1 Base da peridinâmica	18
2.2 Peridinâmica <i>bond-based</i>	21
2.3 Dano em peridinâmica	22
2.4 Modelo <i>PMB</i> modificado	24
3 METODOLOGIA	26
4 IMPLEMENTAÇÃO NUMÉRICA	29
4.1 <i>Solver</i>	30
4.1.1 <i>Solver Dynamic/Explicit</i>	30
4.1.2 <i>Solver Quasi-Static em tempo implícito</i>	31
4.2 Condições de contorno	32
4.2.1 Restrição de deslocamento	32
4.2.2 Restrição de velocidade	33
4.2.3 Carregamento externo	33
4.3 Balanço de energia	34
4.4 Modelos constitutivos	35
4.4.1 <i>Prototype Micro Brittle (PMB)</i>	35
4.4.2 <i>Modified Prototype Micro Brittle (PMB DTT)</i>	35
4.5 Área parcial	36
5 RESULTADOS E DISCUSSÃO	39
5.1 Simulação sem trinca inicial	41
5.2 Simulação com trinca inicial	48
6 Conclusão	57
Referências	59
APÊNDICE A – CÓDIGO FONTE DO PROGRAMA E BIBLIOTECAS	60
ANEXO A – ALGORITMO DO ESQUEMA <i>VELOCIT - VERLET</i>	61
ANEXO B – ALGORITMO PARA O <i>QUASI-STATIC</i>	62
ANEXO C – ALGORITMOS DE FORÇA DE INTERAÇÃO IMPLEMENTADA	63

<u>ANEXO D – ALGORITMOS DE CÁLCULO DE ÁREA PARCIAL</u>	65
--	----

1 INTRODUÇÃO

1.1 Objetivos

Esse trabalho tem como objetivo fazer uma aplicação do método peridinâmico como uma ferramenta alternativa ao método dos elementos finitos. Para tal finalidade será abordado as características da teoria peridinâmica e também será feita uma análise qualitativa dos resultados obtidos através das simulações. A implementação numérica será feita através da linguagem computacional *open source python*.

Como objetivos secundários será a implementação numérica do modelo constitutivo *Modified Prototype Micro Brittle (PMB DTT)*, que será utilizado nas simulações e também a correção da implementação do balanço de energia na simulação.

1.2 Considerações gerais

Uma das suposições da teoria clássica é sua localidade. A teoria clássica da continuidade assume que um ponto material interage somente com seus vizinhos imediatos, ou seja, os mais pertos. A interação dos pontos materiais é governada pelas várias leis de balanço. Portanto, no modelo local um ponto material somente troca massa, energia e momento com seus vizinhos mais próximos. Como resultado, na mecânica clássica, o estado de tensão em um ponto depende da deformação naquele ponto somente. A validade dessa suposição se torna questionável através de diferentes escalas de comprimento. Em geral, em macroescala esta suposição é aceitável. Mas a existência de forças de longo alcance é evidente da teoria atômica e está suposição de interações locais cai à medida que o comprimento de escala fica menor e se aproxima de escala atômica. Até em macroescala, existem situações em que a validade da localidade se torna questionável, por exemplo quando pequenas características e microestruturas influenciam toda a macroestrutura. (MADENCI; OTERKUS, 2014).

Apesar do desenvolvimento de muitos conceitos importantes para prever iniciação de trinca e seu crescimento em materiais. Ainda é um desafio grande dentro da estrutura da mecânica clássica do contínuo. As principais dificuldade estão na

formulação matemática, que assume que um corpo permanece contínuo enquanto deforma. Portanto, a estrutura básica matemática da formulação cai sempre que uma descontinuidade aparece em um corpo. Matematicamente, a teoria clássica é formulada usando equações diferenciais parciais espaciais, e essas derivadas espaciais são indefinidas nas descontinuidades. Tornando-as limitadas pois perdem a definição na presença de descontinuidades como trincas.(MADENCI; OTERKUS, 2014).

As soluções no reino da mecânica do contínuo clássica, resultam em infinitas tensões nas pontas das trincas, o que levou ao conceito de Linear Elastic Fracture Mechanics (LEFM). Dentro do mundo da LEFM, uma trinca pré existente no material é necessária, e as tensões na ponta da trinca são matematicamente singulares. Portanto, o conjunto de inicialização da trinca e o crescimento da trinca são tratados separadamente ao introduzir critério externo, como taxa de liberação de energia crítica, que não são partes das equações governantes da mecânica do contínuo. Além do mais, nucleação na LEFM ainda permanece não resolvido.(MADENCI; OTERKUS, 2014).

A mecânica clássica do contínuo prevê não dispersão enquanto experimentos mostram de outra forma para a propagação de ondas planas elásticas com curtos comprimentos de ondas em sólidos elásticos. No campo da teoria clássica do contínuo (local), um ponto material no contínuo é influenciado somente por outros pontos materiais que estão localizados dentro de suas imediações. Portanto não existe parâmetro de comprimento interno que distingue diferentes escalas comprimentos.(MADENCI; OTERKUS, 2014).

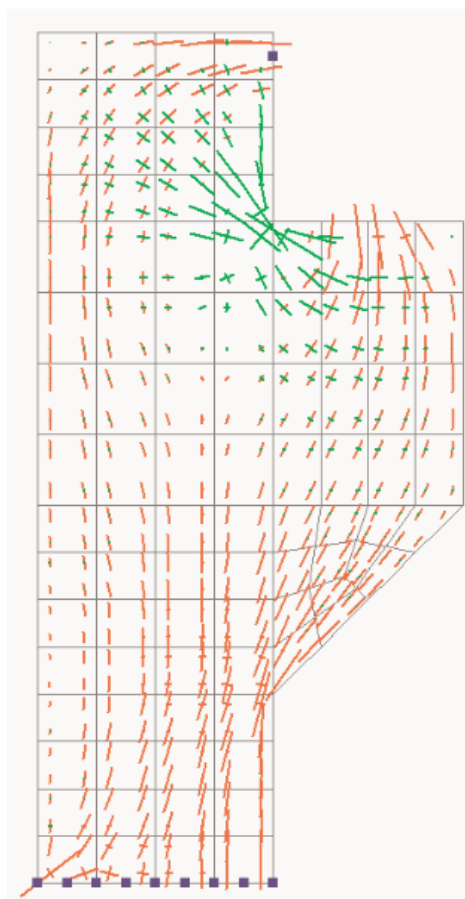
Mesmo que a teoria clássica do contínuo é incapaz de distinguir diferentes escalas de tamanho, ela pode capturar certos processos de falhas e pode ser aplicada a uma ampla variedade de problemas de engenharia, especialmente empregando o Método dos Elementos Finitos (FEM).

O FEM é o método numérico mais aplicado tanto em pesquisa quanto na área industrial para modelar problemas relacionados a dano estrutural. Uma larga variedade de problemas estruturais envolvendo fraturas por fadigas podem ser modelados com o FEM. Porém o FEM geralmente não leva em conta tensões singulares em volta da ponta da trinca, o que, portanto, leva a imprecisão do FEM ao modelar problemas de trincas. O ponto fraco desse método é que ele assume que o corpo continua contínuo à medida que ele deforma. Consequentemente, se torna

difícil a construção das formulações matemáticas na região perto de uma singularidade ou descontinuidade, como no caso das trincas (DIAS et al., 2017).

O FEM é robusto em particular para determinar campos de tensões e deformações como visto na Figura 1, e também é excepcionalmente capaz para modelar estruturas possuindo geometrias complexas e diferentes materiais sob condições de carregamentos gerais. Mas, suas equações governantes são derivadas baseadas na mecânica clássica do contínuo, e também sofre da presença de derivadas espaciais indefinidas de deslocamentos nas pontas das trincas ou ao longo das superfícies das trincas. (MADENCI; OTERKUS, 2014).

Figura 1 – Campo de tensões principais em uma malha



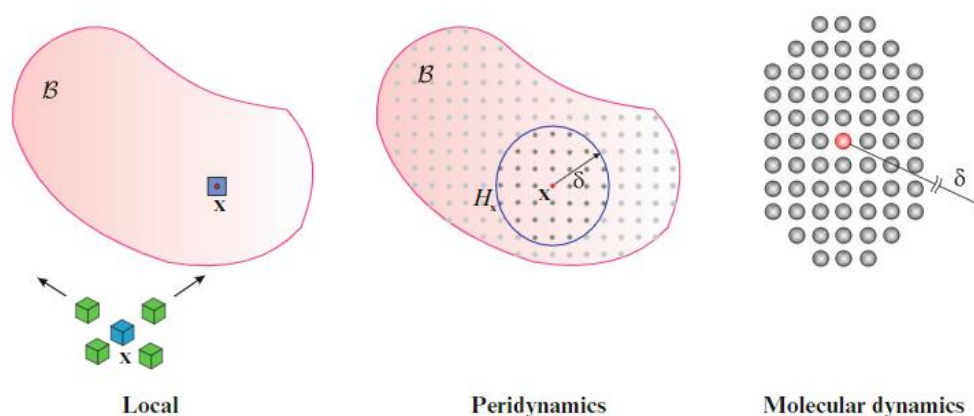
Fonte: (AZEVEDO, 2003)

A teoria não-local da continuidade estabelece a conexão entre mecânica clássica do contínuo e dinâmica molecular. No caso do modelo contínuo local, o estado do ponto material é influenciado por pontos materiais localizados na proximidade

imediate. No caso do modelo contínuo não local, o estado do ponto material é influenciado por pontos materiais localizados dentro de uma região de raio finito. À medida que o raio se torna infinitamente grande, a teoria não local se torna a versão contínua do modelo dinâmico molecular. Portanto, a teoria não-local da continuidade estabelece uma conexão entre a mecânica clássica do contínuo e modelos dinâmicos moleculares (MADENCI; OTERKUS, 2014).

Qualquer ponto x interage com outros pontos materiais dentro de uma distância δ . Os pontos materiais dentro da distância δ de x são chamados de família de x , H_x , como visto na Figura 2. O número de pontos materiais na família de x , dentro do campo da mecânica clássica do contínuo é 3,5,7 (se incluindo) para um, dois e 3 dimensões, respectivamente. (JAVILI et al., 2019).

Figura 2 – Relação entre modelos contínuo local e não-local



Fonte: (MADENCI; OTERKUS, 2014)

Várias teorias não-locais, foram introduzidas no passado. Trabalho recente de Eringen & Edelen (1972) resultaram em uma teoria contínua não-local que contou para não-localidade nas leis de balanceamento e afirmações da termodinâmica. Mas, as equações resultantes foram complicadas e o trabalho posterior por esses pesquisadores simplificaram a teoria contando não-localidade na relação constitutiva enquanto mantendo equações do equilíbrio e cinemática na forma local. Atualmente a maioria das teorias não-locais contam com não-localidade através de relações constitutivas. Geralmente, modelos materiais não-locais do tipo integral em mecânica do contínuo possuem uma lei constitutiva que relaciona as forças (tensões) em um ponto material com alguma média ponderada de deformação (tensão de deformação)

de outros pontos que estão a uma distância finita. Por outro lado, modelos não-locais do tipo gradiente incluem derivadas de ordem maiores para contar pelo campo na proximidade imediata do ponto, como a primeira derivada lei constitutiva da tensão no local. Ambos os tipos de modelos não-locais possuem um comprimento característico associado, que podem ser relacionados aos comprimentos físicos como tamanho de grão, tamanho da zona do processo de fratura ou tamanho do poro. (MADENCI; OTERKUS, 2014).

A teoria contínua não-local foi notada pela sua habilidade de não somente capturar efeito em macroescala, como também em escala atômica e molecular. Foi também expandida para capturar predição de crescimento de trincas.(MADENCI; OTERKUS, 2014).

A teoria da peridinâmica, que é não-local, foi introduzida por Silling (2000) e Silling et al. (2007) em uma tentativa de lidar com as descontinuidades. A teoria peridinâmica emprega deslocamentos em vez de derivadas de deslocamentos em sua formulação. Basicamente, é a reformulação das equações do movimento na mecânica dos sólidos que é melhor aplicado para modelar corpos com descontinuidades como trincas. A teoria usa equações integrais espaciais que podem ser aplicadas descontinuidades. As equações peridinâmicas governantes são definidas nas superfícies de fratura, além disso, dano material é parte das leis constitutivas peridinâmicas. Esses atributos permitem que inicialização de fratura e propagação sejam modeladas com caminhos aleatórios. Sem a necessidade de tratamento especial de crescimento de trinca. (MADENCI; OTERKUS, 2014)

Na peridinâmica, pontos materiais interagem com cada um diretamente através da função de resposta prescrita, que contém todas as informações constitutivas associadas ao material. A função resposta inclui um parâmetro de comprimento chamado comprimento interno (horizonte), δ . A localidade das interações depende do horizonte e as interações se tornam mais locais com a diminuição do horizonte. Assim que as interações entre pontos materiais terminam, trincas podem iniciar e alinhar-se ao longo das superfícies que forma as trincas, e ainda as equações integrais permanecem validas (MADENCI; OTERKUS, 2014).

As forças internas são expressadas através de interações não-locais entre pares de pontos materiais dentro do corpo contínuo, e dano é parte do modelo constitutivo. Interfaces entre materiais diferentes possui suas próprias propriedades e dano pode propagar quando e onde é energeticamente favorável para isso. A teoria

da peridinâmica providencia a habilidade de ligar diferentes tamanhos de escala (MADENCI; OTERKUS, 2014).

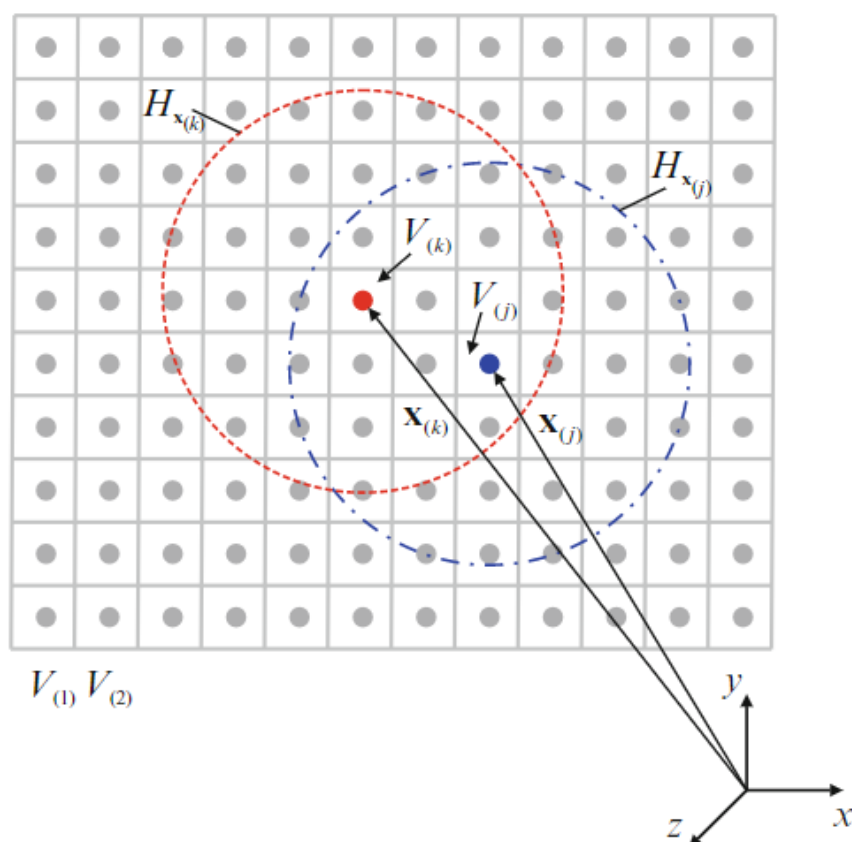
A solução das equações peridinâmicas requerem integração numérica tanto em tempo quanto em espaço. Apesar da equação peridinâmica do movimento incluir efeitos de inércia, é possível usá-la para problemas *quasi-static* ao permitir o termo de inércia sumir através de esquemas (MADENCI; OTERKUS, 2014).

2 FUNDAMENTOS TEÓRICOS

2.1 Base da peridinâmica

Em qualquer instante de tempo, todo ponto no material denota a localização da partícula material, e esses pontos materiais infinitos continuam o contínuo. Em um não deformado estado do corpo, cada ponto material é identificado por suas coordenadas, $x_{(k)}$ com ($k=1,2 \dots$, infinito), e está associado com um volume incremental, $V_{(k)}$, e uma densidade de massa $\rho(x_{(k)})$. Cada ponto material pode estar sujeito a cargas corporais prescritas, deslocamento ou velocidade, resultando em movimento e deformação. Em relação ao sistema de coordenadas Cartesianas, o ponto matéria $x_{(k)}$ sofre deslocamento, $u_{(k)}$, e sua localização é descrita pelo vector posição $y_{(k)}$ no estado deformado. Os vetores de deslocamento e carga de corpo no ponto material $x_{(k)}$ são representados por $u_{(k)}(x_{(k)},t)$ e $b_{(k)}(x_{(k)},t)$, respectivamente. O movimento de um ponto material age de acordo com a descrição Lagrangiana (MADENCI; OTERKUS, 2014).

De acordo com a teoria peridinâmica introduzida por Silling (2000), o movimento de um corpo é analisado ao considerar a interação de um ponto material, $x_{(k)}$, com outro ponto material $x_{(j)}$, com ($j=1,2, \dots$, infinito), no corpo. Portanto um número infinito de interações pode existir entre o ponto material na localização $x_{(k)}$ e outros pontos materiais. Mas, a influência dos pontos materiais com $x_{(k)}$ é assumida que desaparece além do horizonte, denotado por $H_{x_{(k)}}$, mostrado na Figura 3. Similarmente, $x_{(j)}$ interage com pontos em sua família $H_{x_{(j)}}$ (MADENCI; OTERKUS, 2014).

Figura 3– Interação dos pontos materiais $x^{(k)}$ e $x^{(j)}$ 

Fonte: (MADENCI; OTERKUS, 2014)

A teoria da peridinâmica está preocupada com a física do corpo material no ponto que interage com todos os pontos dentro do raio δ , referido como horizonte. O conjunto de pontos materiais dentro da distância δ do ponto $x^{(k)}$ é chamado de família $H_{x^{(k)}}$. A interação dos pontos materiais é descrita através de um micropotencial que depende da deformação e propriedades constitutivas do material. A localidade das interações depende do horizonte, e as interações se tornam mais locais com a diminuição do horizonte (MADENCI; OTERKUS, 2014).

Em seu estado inicial, sem deformação, o vetor posição relativa inicial, vetor posição entre dois pontos, é definido como:

$$\xi^{(k)}(j) = X(j) - X(k) \quad (1)$$

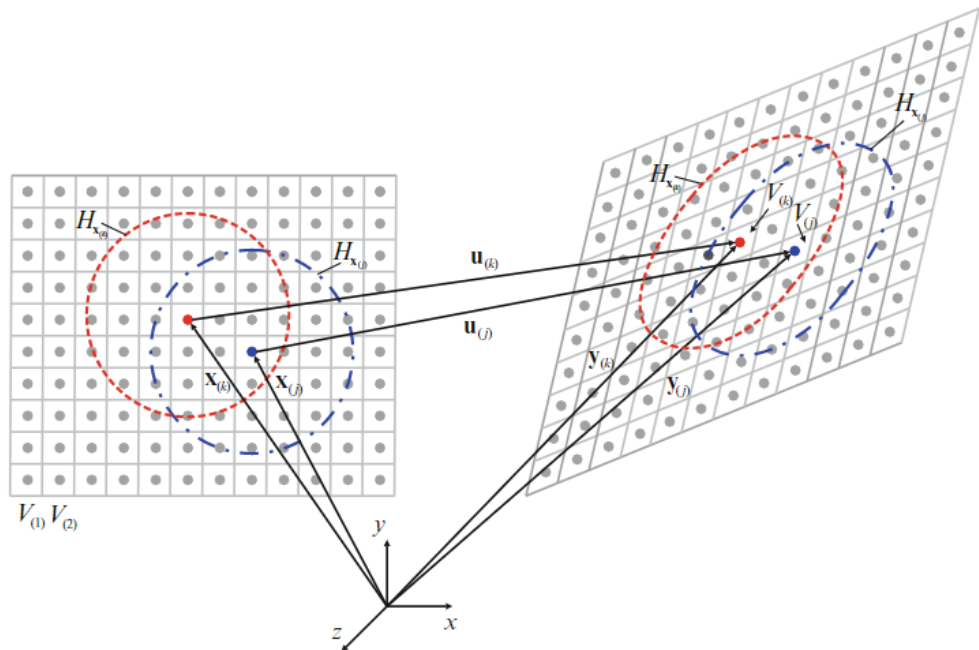
O deslocamento relativo entre dois pontos pode ser definido como:

$$\eta^{(k)}(j) = u(j) - u(k) \quad (2)$$

O ponto material $x^{(k)}$ interage com sua família de pontos materiais, $H_{x^{(k)}}$, e é influenciado pela deformação coletiva desses pontos materiais. Similarmente, o ponto

material $x_{(j)}$ é influenciado pela deformação dos pontos materiais de sua vizinhança $H_{x_{(j)}}$. No corpo deformado, o ponto material $x_{(k)}$ e $x_{(j)}$ sofrem deslocamentos, $u_{(k)}$ e $u_{(j)}$, respectivamente, como mostrado na Figura 4. O vetor de posição inicial relativa ($x_{(j)} - x_{(k)}$) anterior a deformação se torna ($y_{(j)} - y_{(k)}$) após a deformação (MADENCI; OTERKUS, 2014).

Figura 4 – Pontos materiais peridinâmicos após deformação



Fonte: (MADENCI; OTERKUS, 2014)

O alongamento relativo entre os pontos materiais $x_{(k)}$ e $x_{(j)}$ é definido por:

$$s_{(k)(j)} = \frac{(|y_{(j)} - y_{(k)}| - |x_{(j)} - x_{(k)}|)}{|x_{(j)} - x_{(k)}|} \quad (3)$$

Combinando as Equações (2) e (3) obtém-se:

$$s_{(k)(t)} = \frac{|\xi_{(k)(j)} + \eta_{(k)(j)}| - |\xi_{(k)(j)}|}{\xi_{(k)(j)}} \quad (4)$$

Dentro do domínio $H_{x_{(k)}}$, as interações entre os pares de pontos materiais seguem a Segunda Lei de Newton, a equação peridinâmica do movimento é dada a seguir:

$$\rho(x)\ddot{u}(x,t) = \int_{H_x} f(\eta, \xi) dV_x + b(x,t) \quad (5)$$

Sendo ρ a densidade no ponto x , \ddot{u} é a aceleração do ponto material, $f(\eta, \xi)$ é a função força interna por unidade de volume que age na ligação de dois pontos materiais. É possível notar que a Equação 5 é uma equação integral diferencial, em que o diferencial está no termo de inércia, sendo uma derivada temporal (PATRIOTA, 2019).

2.2 Peridinâmica *bond-based*

A interação entre dois pontos materiais depende das quantidades avaliadas somente nos pontos. Para um sólido peridinâmico, por exemplo, a força entre dois pontos materiais x e x' é uma função de campo de deslocamento nesses pontos somente. Desde que esses pontos formam uma ligação (um vetor que liga esses pontos), o modelo depende somente dessa ligação e, portanto, esse ramo da peridinâmica é conhecido por *bond-based* (MADENCI; OTERKUS, 2014). Nesse ramo, o modelo constitutivo mais simples é o material micro-elástico, onde a função de força de interação é chamada de função força par e tem a forma:

$$f(x, x', u) \equiv \hat{f}(\xi, \eta) = \underline{f}(\xi) \underline{e}\langle \xi + \eta \rangle \quad (6)$$

onde

$$\underline{e}\langle \xi + \eta \rangle = \frac{\xi + \eta}{|\xi + \eta|} \quad (7)$$

é o vetor unitário na direção da ligação deformada. A função força par é derivada de uma função micro-elástica potencial $\omega(\xi, \eta)$ como

$$\hat{f}(\xi, \eta) = \frac{\partial \omega}{\partial \eta}(\xi, \eta) \quad (8)$$

esta função micro-potencial pode ser vista como a densidade de energia de deformação de uma ligação. A energia micro-elástica é avaliada como a soma de metade de todas as ligações micro potenciais

$$W(x) = \frac{1}{2} \int_{H_x} \omega(\xi, \eta) dV_\xi \quad (9)$$

um modelo *bond-based* amplamente utilizado é o modelo *prototype-micro-brittle* (PMB) (PATRIOTA, 2019). Nesse modelo, o vetor f é dado por:

$$f(\xi) = c(\xi)s(\xi) \quad (10)$$

Sendo $c(\xi)$ a função micro-módulo, que pode assumir a forma:

$$c(\xi) = Cj(|\xi|) \quad (11)$$

Onde C é uma constante e $j(\cdot) : R \rightarrow R$ é uma função positiva scalar no comprimento da ligação, que implica isotropia ao material. Originalmente no modelo PMB o micro-modulo é constante, portanto, $j(|\xi|) = 1$. Para determinar o valor de $c(\xi)$, é assumido que a clássica energia de deformação é igual a energia de deformação peridinâmica, para uma deformação homogênea. Portanto é possível relacionar o micro-módulo com as clássicas constantes elásticas, que em 2D seria:

$$C = \frac{12E}{\pi t \delta^3 (1 + \nu)} \quad (12)$$

Onde E é o módulo de Young, ν é o coeficiente de Poisson e t é a espessura da placa. Para caso 2D o coeficiente de Poisson possui o valor de $1/3$. Sendo o módulo de Young e coeficiente de Poisson características do material (PATRIOTA, 2019).

A densidade de energia de deformação, no modelo PMB é dada por:

$$W_{PMB}(x) = \frac{1}{4} \int_{H_x} c(\xi) s(\xi)^2 |\xi| dV_x \quad (13)$$

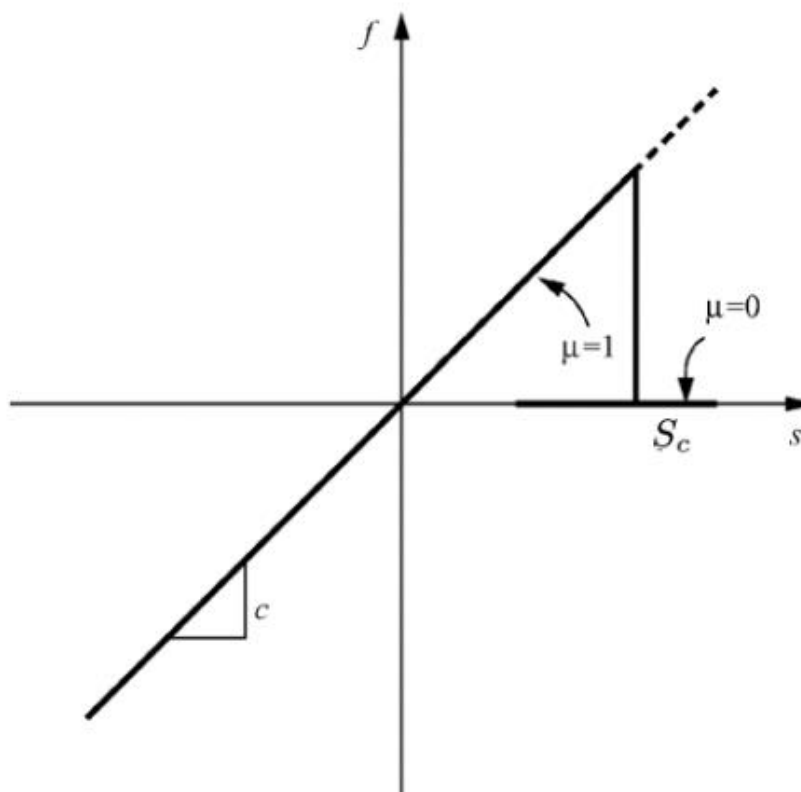
2.3 Dano em peridinâmica

Dano em sólidos peridinâmicos foi introduzido por Silling através do material frágil micro-elástico, é um modelo *bond-based* em que a força escalar, para baixos alongamentos relativos, o material se comporta linearmente, à medida que o alongamento aumenta, a resposta é constante até a falha da ligação, possuindo um comportamento quase plástico. (PATRIOTA, 2019)

(SILLING; ASKARI, 2005) usaram o modelo PMB para modelar a fratura frágil. ela é parecida com o modelo anterior, porém sem a deformação quase plástica, como mostrado na Figura 5.

A falha estrutural em peridinâmica ocorre quando todas as ligações dentro do horizonte se rompem, ou seja, se eles se alongam além dos valores críticos. Em seguida, as forças de contato entre os pontos são terminadas e eles não interagem mais entre si. Esse evento particular mostra o início de uma trinca cuja direção e velocidade de propagação será determinada a medida que outras ligações no domínio são quebradas. (DIAS et al., 2017)

Figura 5 – Modelo PMB com dano



Fonte:(SILLING; ASKARI, 2005)

Onde μ é o fator de dano e S_c é o alongamento crítico.

Pode se redefinir a força escalar PMB para contabilizar ligações partidas:

$$f(\xi) = \begin{cases} \mu(s(\xi))c(\xi)s(\xi), & \text{se } -1 < s(\xi) < S_c \\ 0, & \text{caso contrário} \end{cases} \quad (14)$$

Onde S_c é o alongamento crítico.

O fator de dano μ é definido por:

$$\mu(x) = \begin{cases} 1, & x < S_c \\ 0, & x \geq S_c \end{cases} \quad (15)$$

O valor de S_c pode ser encontrado através da relação mostrada na Equação 16 (PATRIOTA, 2019).

$$S_c = \sqrt{\frac{4\pi G_0}{9E\delta}} \quad (16)$$

Onde G_0 é energia crítica de fratura.

2.4 Modelo *PMB* Modificado

DU; TAO; TIAN (2018) propuseram uma versão modificada do modelo *PMB*, chamado de DTT. Eles consideraram que tanto a função de força de interação quanto o fator de dano são contínuos. Portanto, foram introduzidos os parâmetros S_0^-, S_1^-, S_0^+ e S_1^+ . Sendo eles:

$$-1 < S_1^- < S_0^- < 0 \leq S_0^+ < S_1^+ < \infty$$

O vetor de forças internas é dado por:

$$f = c\mu(s)F(s) \quad (17)$$

Sendo F:

$$F(x) = \begin{cases} S_0^- \frac{x - S_1^-}{S_0^- - S_1^-}, & \text{se } x \in (S_1^-, S_0^-), \\ x, & \text{se } x \in [S_0^-, S_0^+] \\ S_0^+ \frac{S_1^+ - x}{S_0^+ - S_1^+}, & \text{se } x \in (S_0^+, S_1^+), \\ 0, & \text{caso contrário} \end{cases} \quad (18)$$

E

$$\mu = \begin{cases} 1, & \text{se } x \in [-1, S_0^+), \\ \frac{S_1^+ - x}{S_0^+ - S_1^+}, & \text{se } x \in [S_0^+, S_1^+] \\ 0, & \text{caso contrário} \end{cases} \quad (19)$$

A Figura 6 mostra a o modelo DTT com as expressões acima. As mudanças do DTT são sutis o suficiente para garantir uma proximidade entre os modelos e garantir melhor postura do modelo.

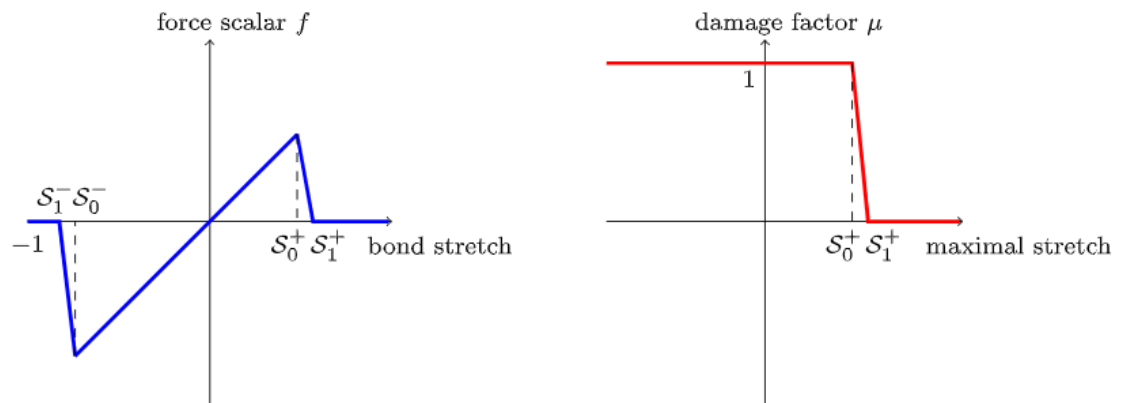
Nota-se que os valores de S_0^- e S_1^- são maiores que -1. o alongamento relativo é -1 quando $\eta = -\xi$, ou seja, quando o ponto material $x_{(j)} = x_{(k)} + \xi$ ocupar o valor de $x_{(k)}$ após a deformação. Essa aproximação é conservativa no sentido de que a utilização usual de modelos de dano é relacionada a testes de tensão para nucleação de trincas e propagação e, portanto, o alongamento nunca chegaria a -1. Em geral o

comportamento de forças de interação deve ser modificado para ficarem próximas de -1 para contabilizar forças de contato. (PATRIOTA, 2019)

Baseado em resultados numéricos, DU; TAO; TIAN (2018) sugerem que os valores de S_0^-, S_1^-, S_0^+ e S_1^+ sejam:

$$\begin{aligned} S_0^+ &= 0.95S_c, & S_1^+ &= 1.05S_c \\ S_0^- &= -0.98, & S_1^- &= 0.99 \end{aligned} \quad (20)$$

Figura 6 – Força de interação entre pontos e fator de dano para o modelo DTT



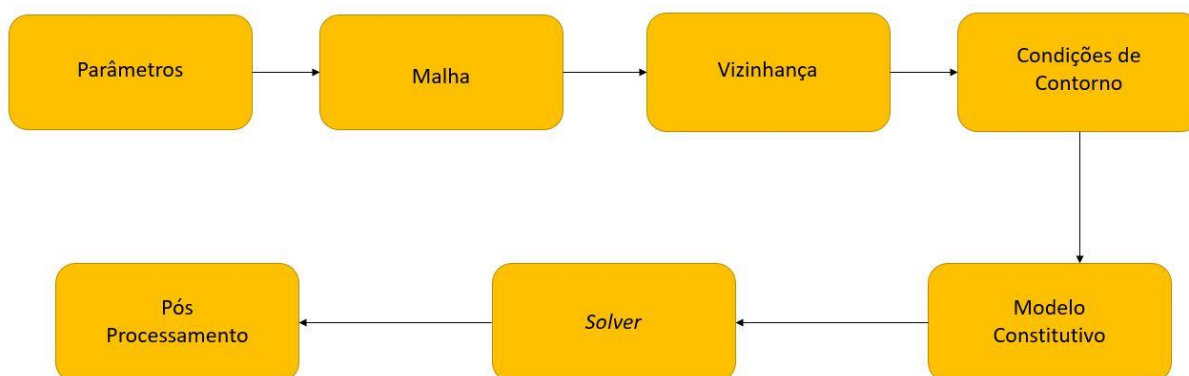
Fonte: (DU; TAO; TIAN, 2018)

3 METODOLOGIA

A realização desse Trabalho de Graduação, teve como base a teoria peridinâmica, o estudo de *papers*, teses e livros sobre o assunto. Foi aplicado o código computacional na linguagem de programação *python*, pois é uma linguagem *open source*, logo mais pessoas podem ter acesso ao código e também ajudar a melhorá-lo. O código em *python* desenvolvido por Chaim (2022) foi baseado no programa em *Matlab* desenvolvido por Patriota (2019) em sua tese de doutorado. O código utilizado neste trabalho, assim como o código desenvolvido por Chaim (2022) e Patriota (2019) se encontram no Apêndice A.

Sobre o programa, ele pode ser dividido em um fluxograma em 7 partes, como mostrado na Figura 7.

Figura 7 – Fluxograma para execução do programa em *python*



Fonte: Elaborado pelo autor

Antes de iniciar as etapas é necessário importar o programa e as bibliotecas necessárias, para isso utiliza-se o *import*. Nesse trabalho foi utilizado as bibliotecas *numpy*, *matplotlib* e foi importado o programa como se fosse uma biblioteca.

A primeira etapa foi colocar os inputs em variáveis importantes para a simulação, as características do material foram o módulo de Young, massa específica e energia crítica de fratura, as características geométricas foram a altura e comprimento da malha e os parâmetros peridinâmicos foram o horizonte, razão de malha e o espaçamento da malha e uma variável que foi chamada de *par_omega*, ela contém 3 valores, sendo eles o horizonte, e os outros dois são referentes a função influência considerada.

A segunda etapa é a criação da malha, é utilizada a função *Mesh* para criar um objeto e poder gerar e manipular a malha, como parâmetros é necessário fornecer o tamanho da célula, o comprimento e altura da malha. O objeto criado representando a malha será utilizado nas próximas etapas. A malha foi criada em formato retangular nesse trabalho. Existem também, atributos gerados automaticamente que podem ser utilizados para adquirir, como por exemplo número de linhas, número de colunas e área de cada elemento da malha.

Em seguida vem a etapa para se gerar a família de cada ponto da malha e também, nesse algoritmo é feito o cálculo das áreas parciais. Para isso é criado um objeto utilizando a função *Family*, em que é preciso fornecer a malha (o objeto criado na etapa anterior), o horizonte, a razão de malha e também uma *string* com o nome do algoritmo a ser utilizado para o cálculo das áreas parciais.

Na quarta etapa é criado um objeto utilizando a função *BoundaryCondition* para manipular as condições de contorno da simulação. As condições de contorno irão depender do tipo de *solver* escolhido para ser utilizado na simulação, ou seja, *quasi-static solver* ou *dynamic/explicit*. Caso seja o *quasi-static solver* deverá ser inserido restrições de deslocamento e as forças de corpo. Caso seja o *dynamic/explicit* deverá ser inserido restrições de deslocamento e velocidade prescrita de cada nó que terá as prescrições. Para cada condição de contorno utilizada é preciso utilizar seu método. Para a restrição de deslocamento utilizá-se o método *set_displacement_constraint*, nela é necessário passar como parâmetro o objeto malha, um vetor contendo os nós que possuem a restrição, um vetor booleano com valores *True* ou *False* para cada ponto da restrição indicando se é verdadeiro na posição x, um vetor booleano como o anterior mas para a posição y, um vetor indicando o valor do deslocamento para cada ponto restrito em x e outro vetor para o valor do deslocamento restrito de cada ponto em y. Para a restrição de velocidade utiliza-se o método *set_velocities_constraint* e é utilizado como parâmetro como foi utilizado para a restrição de deslocamento. Para indicar a condição de carregamento externo é utilizado o método *set_body_forces*, como parâmetro é utilizado o objeto criado para a malha, um vetor contendo os valores das tensões na direção x,y e xy, e um vetor contendo os nós em que as tensões são aplicadas. É também nesta etapa onde se inclui a trinca inicial, utilizando o método *set_crackIn* onde é necessário passar como parâmetro as coordenadas iniciais e finais da trinca.

Após criado o objeto das condições de contorno, é necessário criar um objeto para definir qual modelo constitutivo a ser utilizado, através da função *model* do programa. Um dos objetivos desse trabalho também é adicionar o modelo constitutivo *PMB DTT*. Como parâmetro desta função, é preciso passar uma *string* com o nome do modelo a ser utilizado, a variável *par_omega*, o Módulo de Young, a energia crítica de fratura e o objeto criado na etapa das condições de contorno.

A próxima etapa é a utilização do solver, aqui é utilizado a função *solver* do programa e como parâmetro, é escolhido se será *quasi-static* ou *dynamic/explicit*. A função *solver* irá retornar algumas variáveis, que serão usadas na etapa de pós processamento. No caso desse trabalho foi utilizado o *solver.QuasiStatic* e como parâmetro foi dado o objeto criado na malha, o número de passos, o objeto criado na condição de contorno, o objeto criado na família, o objeto criado no modelo constitutivo, a variável *par_omega*, e uma variável chamada de *penalty* que foi atribuído o valor de 1.10^{10} .

Na etapa de pós processamento é utilizado as variáveis retornadas pela função *solver* para a plotagem dos gráficos utilizados, podendo plotar gráficos de descolamentos, tanto na direção x quanto y e deslocamento da malha utilizando a função *plot_displacement*. gráfico de densidade de energia de deformação com a função *plot_energy*, gráficos de deformações com a função *plot_strain*, e o índice de dano com a função *plot_damage* e como objetivo desse trabalho está a correção do gráfico de evolução da energia à função *plot_energy*, onde as curvas da energia de deformação, energia cinética, trabalho externo e a energia interna total são mostradas.

4 IMPLEMENTAÇÃO NUMÉRICA

As soluções analíticas para problemas peridinâmicos normalmente são desconhecidos devido por serem complexas, portanto, é necessária uma simulação numérica para obter uma solução aproximada desses problemas para se avançar no entendimento ou aplicação prática. (PATRIOTA, 2019)

Neste tópico, será discutido os aspectos principais do procedimento de implementação numérica.

Tendo como base a Figura 8, tendo \mathcal{B}_D como o domínio discretizado bidimensional. Cada ponto i é onde variáveis físicas são calculadas e possuem a posição x_i , no centro de cada célula quadrada τ_i com volume h^2 e densidade ρ_i . A equação governante é integrada espacialmente através da regra ' quadratura do ponto médio. Portanto para um ponto $x_i \in \mathcal{B}_D$, a equação governante discreta da peridinâmica pode ser escrita como (PATRIOTA, 2019):

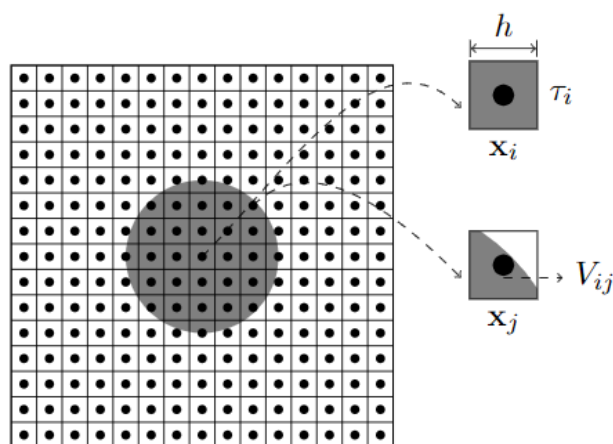
$$\rho_i \ddot{u}_i = \sum_{j \in \mathcal{F}_i} f_{ij} \Delta V_{ij} + b_i \quad (21)$$

Onde $\ddot{u}_i = \ddot{u}(x_i)$, $b_i = b(x_i)$, \mathcal{F}_i é a família de pontos que estão relacionados ao ponto i através:

$$\mathcal{F}_i = \{j \mid |x_j - x_i| < \delta\},$$

$f_{ij} = f(x_i, x_j, u)$ é a força de interação entre os nós com os índices i e $j \in \mathcal{F}_i$ e ΔV_{ij} é o volume do nó j dentro da vizinhança do nó i .

Figura 8 – Esquema de domínio peridinâmico discretizado



Fonte: (PATRIOTA, 2019)

É importante ressaltar que o tamanho do horizonte δ é um parâmetro material, portanto não deve depender da geometria. Em microestruturas, o horizonte pode ser visto como a distância de interação microscópica entre átomos. Para macroestruturas, seu tamanho é escolhido sendo pequeno o suficiente para ter resultados conclusivos sem ter desnecessários custos computacionais (PATRIOTA, 2019). Ao se tratar de simulações numéricas, é preciso contabilizar a razão de malha

$$d = \frac{\delta}{h} \quad (22)$$

Esta razão mede o refino da malha na vizinhança. Quanto maior o valor de d , mais precisa é a quantidade integral avaliada na vizinhança do ponto material. É uma boa prática definir uma razão de malha custo-efetivo que garante precisão, geralmente assumindo os valores:

$$\begin{aligned} 2D & 4 < d < 6 \\ 3D & 6 < d < 9 \end{aligned} \quad (23)$$

Em seguida é feita uma análise δ -convergence para escolher um tamanho de horizonte adequado, em que d é mantido fixo e δ vai variando e aproximando de zero, desse modo a solução aproximada converge para a solução clássica local. Um pequeno horizonte com uma pequena razão de malha pode levar a soluções incorretas (PATRIOTA, 2019).

4.1 Solver

Com a finalidade de testes e simulações do programa, existem duas opções de solvers, o *quasi-static* e o *dynamic/explicit*.

4.1.1 Solver Dynamic/Explicit

Em simulações dinâmicas, existe uma integração no tempo realizada pelo método da diferença central, utilizando um esquema de tempo explícito chamado de Velocity-Verlet como mostrado no Algoritmo 1 presente no Anexo A.

Devido ao uso de um método explícito, o incremento de tempo Δt deve ser pequeno suficiente para garantir estabilidade (PATRIOTA, 2019).

SILLING & ASKARI (2005) propuseram uma expressão para definir o incremento de tempo crítico:

$$\Delta t_{crit} = \min \left(\sqrt{\frac{2\rho}{\sum_{j \in \mathcal{F}_i} C_{ij} \Delta V_{ij}}} \right) \quad (24)$$

Onde C_{ij} é o micro-módulo avaliado na ligação $\xi = x_j - x_i$. Tal expressão é derivada para o caso unidimensional. Para maiores dimensões, é um estimador conservador. Se o material for não-linear, a expressão do micro-módulo pode ser derivada de:

$$C_{ji} = |C_{ji}| \quad C_{ji} = \frac{\partial f}{\partial \eta} |_{\eta=0, \xi = \xi_{ij}} \quad (25)$$

Na equação acima, f é a função de força de interação entre pontos materiais e η é o deslocamento relativo entre dois pontos. Manipulando as Equação (25) e Equação (10), obtém-se:

$$C(\xi) = \frac{C \omega_\delta(|\xi|)}{|\xi|^3} \begin{bmatrix} \xi_1^2 & \xi_1 \xi_2 \\ \xi_2 \xi_1 & \xi_2^2 \end{bmatrix} \quad (26)$$

Onde na expressão os subscritos de ξ são componentes cartesianos do vetor de posição relativa. Para contabilizar efeitos não lineares, define-se o incremento de tempo menor que o valor crítico. A constante C é inversamente dependente do horizonte δ . Valores de C constante são dados pela Equação (12) (PATRIOTA, 2019).

4.1.2 Solver Quasi-Static em tempo implícito

Para experimentos em regime permanente, em que o termo da inércia $\rho \ddot{u}$ é igual a zero, a equação governante discreta assume a forma:

$$\sum_{j \in \mathcal{F}_i} f_{ij} \Delta V_{ij} + b(x_i) = 0 \quad (27)$$

Admitindo o primeiro e segundo componentes Cartesianos do campo de deslocamento do nó i , tendo como l e k graus de liberdade do sistema, pode-se reescrever a equação acima na forma de vetor para todo o sistema:

$$\bar{f}_{int} + \bar{b} = 0 \quad (28)$$

Onde os componentes l e k de \bar{f}_{int} ficam na forma:

$$\bar{f}_{int,l} = \sum_{j \in \mathcal{F}_i} f_{ij,1} \Delta V_{ij} \quad \bar{f}_{int,k} = \sum_{j \in \mathcal{F}_i} f_{ij,2} \Delta V_{ij} \quad (29)$$

Onde $f_{ij,1}$ e $f_{ij,2}$ são componentes Cartesianos da força de interação f_{ij} . Cada linha da variável vetorial sem uma sobrelinha estará referenciando a um índice de nó enquanto cada linha da variável vetorial com sobrelinha estará referenciando a um índice de grau de liberdade. A forma vetorial é mais vantajosa que a forma de matriz pois permite melhor lidar com restrições cinemáticas (PATRIOTA, 2019).

Para este caso estático, usa-se um solver quasi-static *implicit time*, baseado no método de Newton. As condições de carregamento são atualizadas a cada passo ao invés do passo de tempo, conduzindo assim para uma configuração desbalanceada que implica em uma solução atualizada do campo de deslocamento. O método avalia a solução do campo de deslocamento que minimiza um resíduo escalar r para um dado passo do carregamento (PATRIOTA, 2019).

Para cada passo do carregamento, o valor de r , que tem unidade de força, deve ser nulo para uma solução numérica da Equação (27). De outro modo, utiliza-se um esquema iterativo para alcançar a solução aproximada da Equação (27) que irá aproximar r para um valor menor que um limite dado por r_e . Esse algoritmo pode ser visualizado no Anexo B Algoritmo 2 deste trabalho.

4.2 Condições de contorno

Pode-se aplicar 3 tipos de condições de contorno: restrição de deslocamento, restrição de velocidade e carga externa (MADENCI; OTERKUS, 2014).

4.2.1 Restrição de deslocamento

Seja Γ_d o conjunto de restrições de graus de liberdade onde deslocamento é imposto. O vetor contendo os componentes do deslocamento prescrito no tempo n é dado por:

$$\bar{u}^n(\Gamma_d) = u_p \quad (30)$$

Onde u_p é o vetor que coleta o deslocamento prescrito.

As restrições são aplicadas diretamente nos nós. No trabalho somente deslocamentos constantes podem ser prescritos (MADENCI; OTERKUS, 2014).

4.2.2 Restrição de velocidade

Similarmente à última seção sobre restrição de deslocamento, definindo Γ_v como o conjunto de restrições de graus de liberdade onde velocidade constante é imposta. O vetor v_p contém todas as velocidades prescritas nesse conjunto e o vetor contendo os componentes do deslocamento correspondentes em Γ_v no tempo n é:

$$\bar{u}^n(\Gamma_v) = \bar{u}^{n-1}(\Gamma_v) + v_p \Delta t \quad (31)$$

Onde Δt é o incremento de tempo (MADENCI; OTERKUS, 2014).

4.2.3 Carregamento externo

Forças de contato externas são aplicadas como força de tração, por unidade de área, em rotinas baseadas em modelos da mecânica clássica. Na peridinâmica as forças de interação entre pontos materiais agem a distância e, portanto, não podem ser impostas na superfície. Assim a força de tração é aproximada à uma força de corpo, por unidade de volume, atuando em uma camada fina próxima do contorno onde a força de tração é aplicada. Considerando Ω_t o conjunto de nós no contorno de B_d onde as forças de tração são descritas, então o vetor de força de corpo é dado pela Equação 32 (PATRIOTA, 2019).

$$b(x_{\Omega_t}) = \frac{t \Delta x}{d A_{\Omega_t}} \quad (32)$$

Onde Δx é o comprimento da célula normal à força que está no contorno, no caso de malha homogênea $\Delta x = h$, d é a razão de malha, A_{Ω_t} é a área da célula, sendo $A_{\Omega_t}^2$ para malha homogênea e

$$t = \sigma n = t_1 e_1 + t_2 e_2 \quad (33)$$

É a força de tração, sendo σ o tensor de tensão e n é o vetor unitário normal à superfície. Uma vez que o campo de força de corpo está definido perto do contorno onde as forças de tração são aplicadas, pode se obter o vetor de graus de liberdade das forças de campo \bar{b} (MADENCI; OTERKUS, 2014).

4.3 Balanço de energia

A energia total do sistema não deve aumentar e, no caso de não houver dissipação, ela deve ser constante. A energia através do tempo ajuda a entender se a simulação está estável (PATRIOTA, 2019). A energia total de um corpo deformado é avaliada como

$$E(t^n) = K(t^n) + P(t^n) - W^E(t^n) \quad (34)$$

Onde todas as quantidades são avaliadas no tempo $t^n = n\Delta t$

Sendo $K(t^n)$ a energia cinética discreta, dada como:

$$K(t^n) = \frac{1}{2} \sum_{i \in \mathcal{B}_D} \rho_i |v_i^n|^2 V_i \quad (35)$$

$P(t^n)$ sendo a energia potencial, dada como:

$$P(t^n) = \sum_{i \in \mathcal{B}_D} W(x_i, t^n) V_i \quad (36)$$

Onde V_i é o volume da célula i , v_i é o vetor velocidade para cada ponto e W é função densidade de energia de deformação (PATRIOTA, 2019).

$W^E(t^n)$ é o trabalho externo realizado, ele depende da aplicação da condição de contorno. Se a condição de tração for aplicada como força de corpo, e, se o carregamento for constante no tempo, a expressão é dada como (PATRIOTA, 2019):

$$W^E(t^n) = \sum_{i \in \mathcal{B}_D} b_i \cdot u_i^n V_i \quad (37)$$

Caso a condição de contorno seja restrição de velocidade, a equação se torna mais complexa. Tendo restrição de velocidade, estará prescrevendo uma força externa, implicitamente (PATRIOTA, 2019). Porém não será abordado esse caso nesse trabalho.

Caso o sistema não tenha nenhuma fonte de dissipação de energia, todo o trabalho externo será transferido para acelerar os pontos e deformá-los. Portanto, a energia total deverá ser:

$$E(t) = 0, \quad \forall t \in [0, t_{final}] \quad (38)$$

Se houver dissipação, então espera-se que:

$$\frac{dE}{dt} < 0 \quad (39)$$

4.4 Modelos Constitutivos

Para que um modelo constitutivo seja implementado no código, é necessário implementar uma rotina com o objetivo de avaliar a força de interação f_{int} e avaliar densidade da energia de deformação.

Nessa seção será apresentado algoritmos para avaliação da força de interação para os modelos implementados, como também a avaliação dos parâmetros elásticos. É importante salientar que a simulação e, portanto, o modelo implementado é em 2D com condição de tensão plana.

4.4.1 Prototype Micro Brittle (PMB)

Para este modelo, a força de interação é avaliada como

$$f_{int}(x_i) = \sum_{j \in \mathcal{F}_i} C_{PMB} \omega_\delta(|\xi_{ij}|) \mu(s_{ij,max}) \frac{|\xi_{ij} + \eta_{ij}| - |\xi_{ij}|}{|\xi_{ij}|} \underline{e}_{ij} \Delta V_{ij} \quad (40)$$

Onde μ é a função de dano, C_{PMB} é o micro módulo, e_{ij} é o vetor unitário na direção da ligação deformada e $s_{ij,max}$ é o alongamento relativo máximo da ligação entre nós i e j no tempo atual da simulação (PATRIOTA, 2019). A avaliação numérica da Equação (40) está implementada no Algoritmo 3 do Anexo C.

A densidade de energia de deformação discretizada é

$$W_{PMB}(x_i) = \frac{1}{4} \sum_{j \in \mathcal{F}_i} C_{PMB} \omega_\delta(|\xi_{ij}|) \mu(s_{ij,max}) (\underline{s}_{ij})^2 |\xi_{ij}| \Delta V_{ij} \quad (41)$$

4.4.2 Modified PMB (DTT)

A função da força de interação é dada pela Equação 42 (PATRIOTA, 2019)

$$f_{int}(x_i, u) = \sum_{j \in \mathcal{F}_i} C_{PMB} \omega_\delta(|\xi_{ij}|) \bar{\mu}(s_{ij,max}) F(\underline{s}_{ij}) \underline{e}(\xi_{ij} + \eta_{ij}) \Delta V_{ij} \quad (42)$$

Onde $F(s_{ij})$ é dada pela Equação (18) e $\bar{\mu}(s_{ij,max})$ é dado pela Equação (19).

A avaliação numérica da Equação (42) está implementada no Algoritmo 4 do Anexo C.

A função da densidade da energia de deformação é dada pela Equação 43

$$W_{DTT}(x_i) = \frac{1}{4} \sum_{j \in \mathcal{F}_i} C_{PMB} \omega_\delta(|\xi_{ij}|) \mu(s_{ijmax}) p\left(\frac{|\xi_{ij} + \eta_{ij}| - |\xi_{ij}|}{|\xi_{ij}|}\right) |\xi_{ij}| \Delta V_{ij} \quad (43)$$

Onde $p(x)$ é dado por:

$$p(x) = \begin{cases} \frac{S_0^-}{S_0^- - S_1^-} \left(\frac{x^2}{2} - S_1^- x \right) + A_1, & x \in [S_0^-, S_1^-), \\ x^2/2, & x \in [S_1^-, S_0^+], \\ \frac{S_0^+}{S_1^+ - S_0^+} \left(S_1^+ x - \frac{x^2}{2} \right) + A_3, & x \in (S_0^+, S_1^+], \\ A_4, & x \in (S_1^+, \infty), \end{cases} \quad (44)$$

onde, A_1, A_3 e A_4 são constantes a serem determinadas aplicando condições de continuidade (PATRIOTA, 2019).

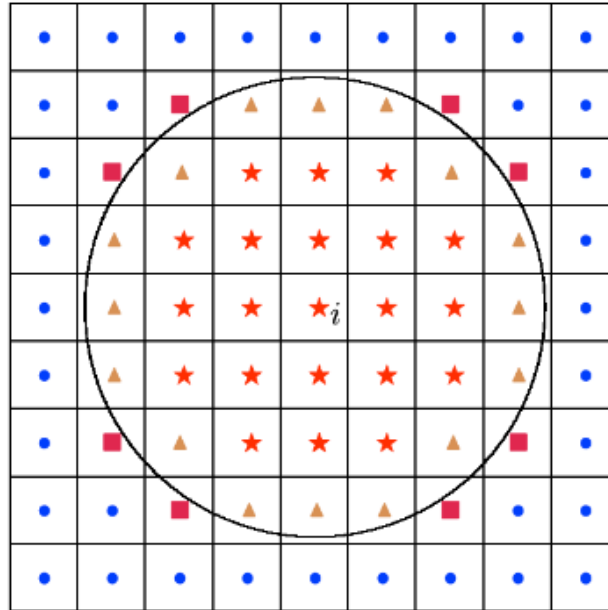
4.5 Área parcial

A integração espacial na equação peridinâmica é geralmente feita com a regra da quadratura do ponto médio. Nessa seção será estudada a avaliação da área parcial, no caso 2D.

Considerando o domínio discretizado \mathcal{B}_D , é possível dividir as células em 4 classes (SELESON, 2014)

- Classe 1: pontos representados por estrelas laranjas na Figura 9, são os pontos cujas células estão dentro da vizinhança no nó i .
- Classe 2 pontos, representados pelos triângulos amarelos na Figura 9, estão localizados dentro da vizinhança do nó i , mas cujas células estão parcialmente dentro da vizinhança.
- Classe 3 pontos, representados pelos quadrados vermelhos na Figura 9, que estão fora da vizinhança do nó i mas cujas células estão parcialmente dentro da vizinhança.
- Classe 4 pontos, representados pelos pontos azuis na Figura 9, cujas células estão fora da vizinhança do nó i .

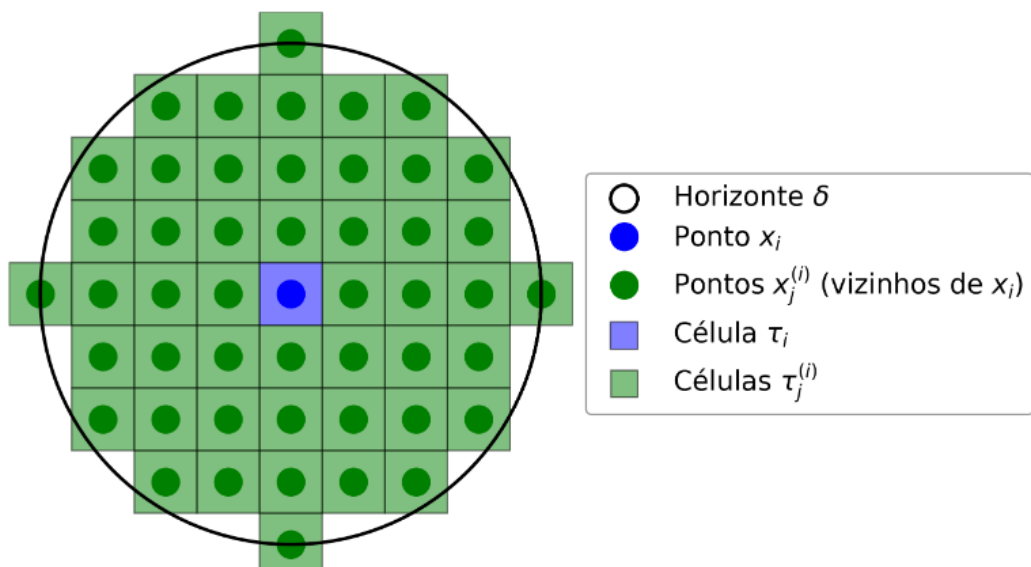
Figura 9 – Domínio discretizado com vizinhança e classes de pontos



Fonte: (SELESON, 2014)

O algoritmo conhecido como full área (FA) é o mais simples, e originalmente foi implementado por SILLING & ASKARI (2005). Nele, como mostrado na Figura 10, a vizinhança é modificada de modo que ela inclua as células da Classe 2 e exclua as células da Classe 3. Portanto, não existirá áreas parciais mais, porém, ocorrerá duas fontes de erros: as áreas faltantes das células da Classes 3 e as áreas a mais incluídas pelas células da Classe 2.

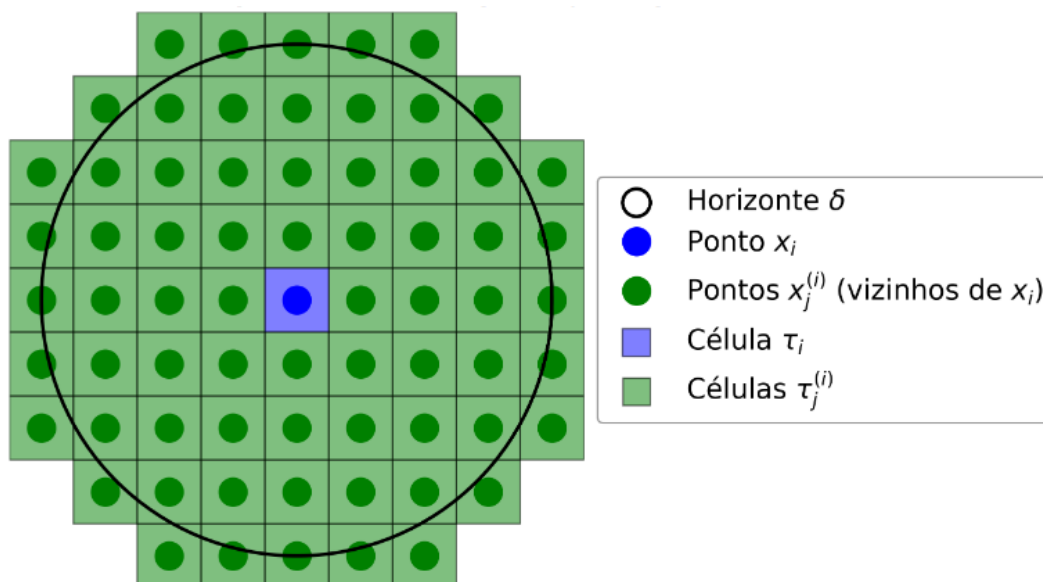
Figura 10 – Família gerada pelo algoritmo FA



Fonte: (CHAIM, 2022)

O algoritmo *Partial Area Analytical Calculation (PA-AC)*, desenvolvido por (SELESON, 2014), avalia as áreas parciais e possui uma performance melhor que o FA, ao incluir as células de Classe 3 na vizinhança, como mostrado na Figura 11.

Figura 11 – Família gerada pelo algoritmo PA-AC



Fonte: (CHAIM, 2022)

5 RESULTADOS E DISCUSSÃO

Neste trabalho, foi abordado duas simulações, a primeira delas será a aplicação de uma carga de tração em uma placa fina, a segunda será uma simulação semelhante a primeira, porém com a inserção de uma trinca inicial. Ambas as simulações serão abordadas e discutidas detalhadamente mais à frente.

A placa simulada possui dimensões de 1 *m* de comprimento e 0.15 *m* de altura, sendo o seu formato retangular. O material que constitui a placa é o *soda-lime glass*, vidro comum muito utilizado em garrafas para bebidas, cujas propriedades estão listadas na Tabela 1.

Tabela 1 – Propriedades materiais do *soda-lime glass*

Propriedade	Valor
Modulo de Young (GPa)	72
Densidade (Kg/m ³)	2440
Energia Crítica de Fratura (JM ⁻²)	3.8

Fonte: (PATRIOTA, 2019)

Além das propriedades materiais e geométricas, é muito importante para as simulações as propriedades peridinâmicas. Elas estão listadas na Tabela 2:

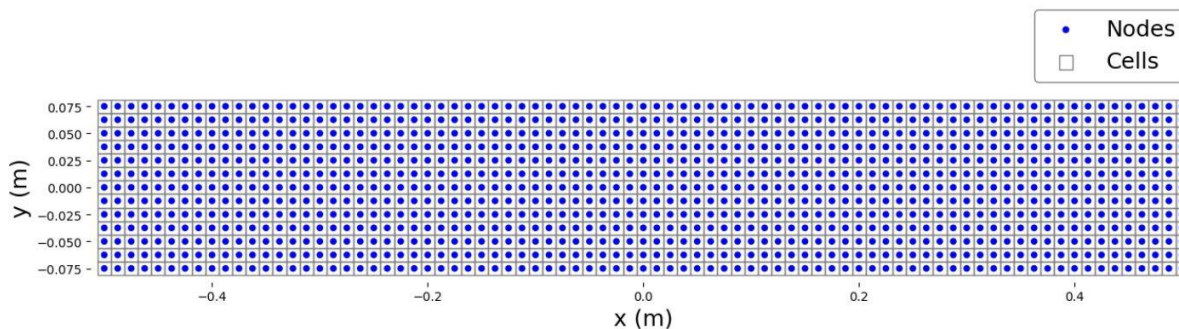
Tabela 2 – Propriedades peridinâmica

Propriedade	Valor
Horizonte (mm)	50
Razão de malha	4
Tamanho de cada célula (mm)	12.5

Fonte: Elaborada pelo autor

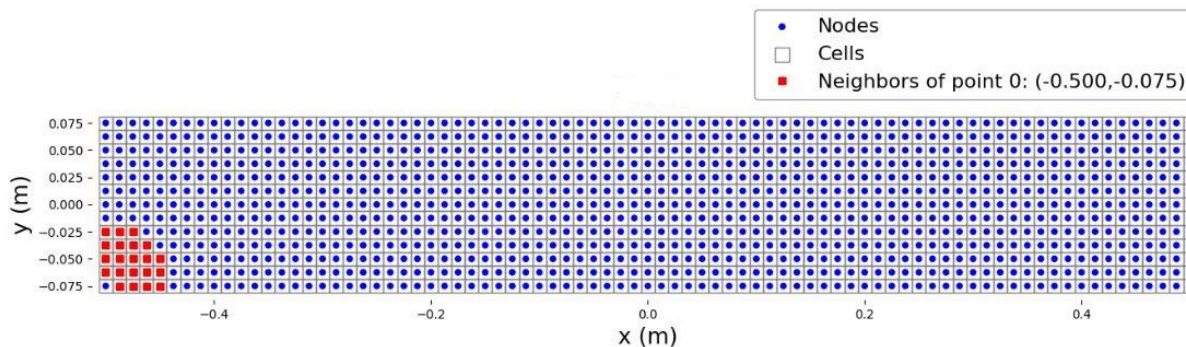
Após definir as propriedades, a próxima etapa do programa é a criação da malha a ser simulada. A Figura 12, mostra a malha feita para as simulações, ela possui 81 colunas, 13 linha e, portanto, 1053 pontos materiais. Cada célula possui dimensões *h* de 12.5 mm.

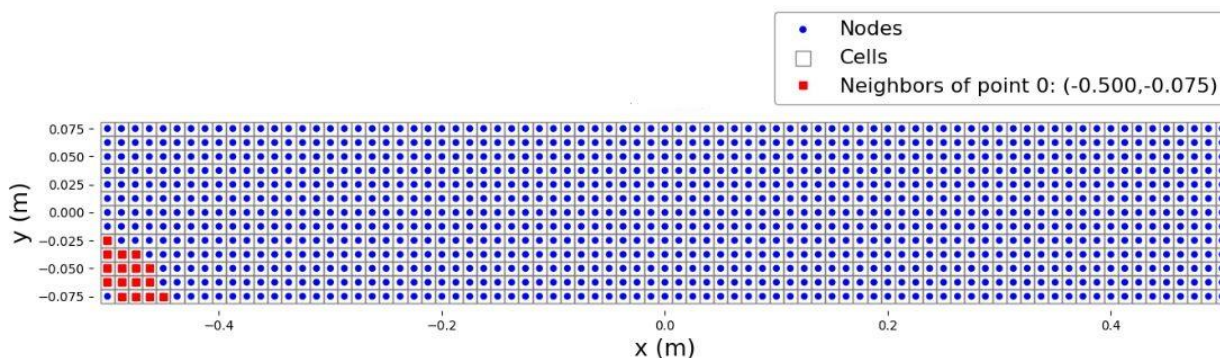
Figura 12 – Malha discretizada da placa



Fonte: Elaborado pelo autor

Em seguida é necessário a criação da família de cada ponto material da malha. No programa, para a criação da família é preciso fornecer a malha, algumas propriedades peridinâmicas e o nome do algoritmo para cálculo da área parcial. Nesse trabalho foi escolhido utilizar o algoritmo *PA-AC*, pois é um algoritmo que calcula as áreas parciais com uma expressão analítica e calcula corretamente. A Figura 13 mostra a vizinhança do ponto material de índice 0, tanto para o algoritmo *PA-AC* quanto o algoritmo *FA*.

Figura 13 – Comparação de famílias geradas pelos algoritmos *PA-AC* e *FA*(a) Algoritmo *PA-AC*

(b) Algoritmo *FA*

Fonte: Elaborado pelo autor

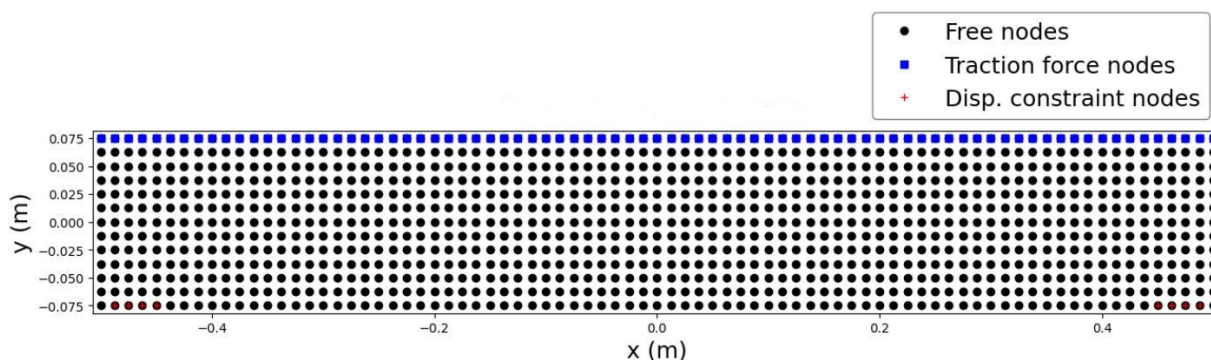
Nota-se que o algoritmo *PA-AC* acaba incluindo mais pontos materiais na família do ponto material que o algoritmo *FA*, isso ocorre pois o algoritmo *FA* não inclui os pontos que estão fora do horizonte, mas cujas células estão parcialmente dentro.

5.1 Simulação sem trinca inicial

Nessa seção será discutido os resultados obtidos pela simulação na placa $1 \times 0.15\text{m}$ sem trinca inicial. Foi utilizado o modelo constitutivo *PMB DTT* e o solver utilizado foi o *quasi-static*. Devido ao *solver* utilizado, não foi considerado dano na simulação.

As condições de contorno utilizadas na placa foram deslocamento prescrito e carga de tração, como visto na Figura 14. Os pontos em vermelho são os deslocamentos prescritos, os pontos em azul são os nós com tensões de tração e os pontos pretos são os pontos livres.

Figura 14 – Condições de contorno aplicado a malha sem trinca inicial

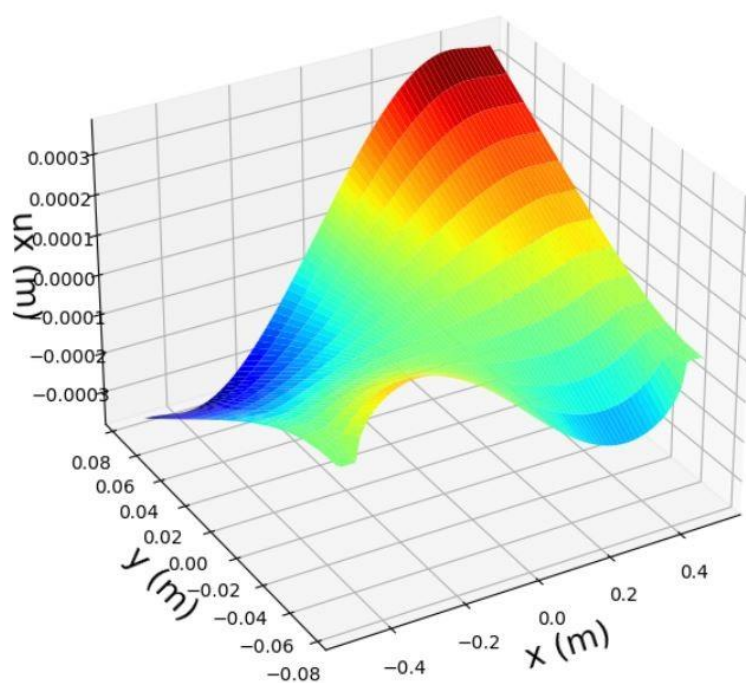


Fonte: Elaborado pelo autor

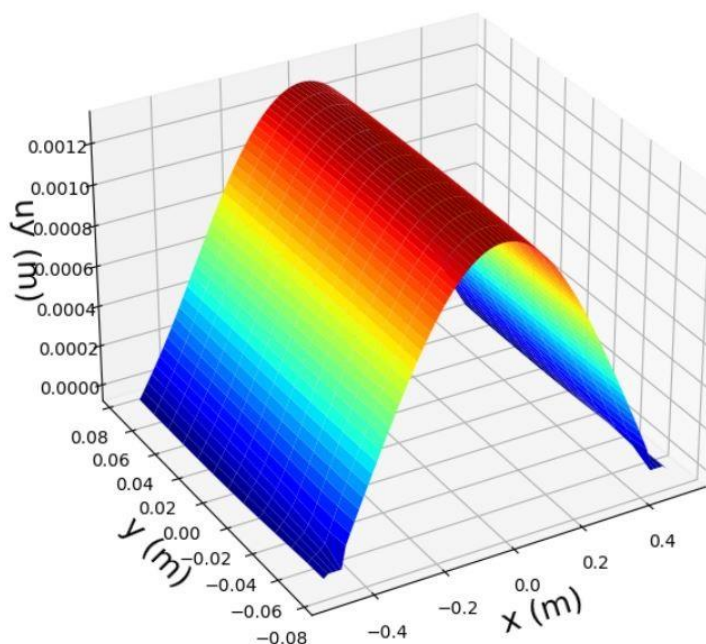
O deslocamento foi prescrito como nulo nas direções x e y nos pontos 1,2,3,4,76,77,78,79, na primeira linha da malha. Enquanto as tensões de tração foram prescritas como sendo 6 MPa nos nós começando em 972 até 1053, ou seja, todos os nós da última linha da malha.

A Figura 15 mostra os deslocamentos tanto em x quanto em y dos pontos.

Figura 15 – Campo de deslocamento da malha sem trinca inicial



(a) Deslocamento na direção x



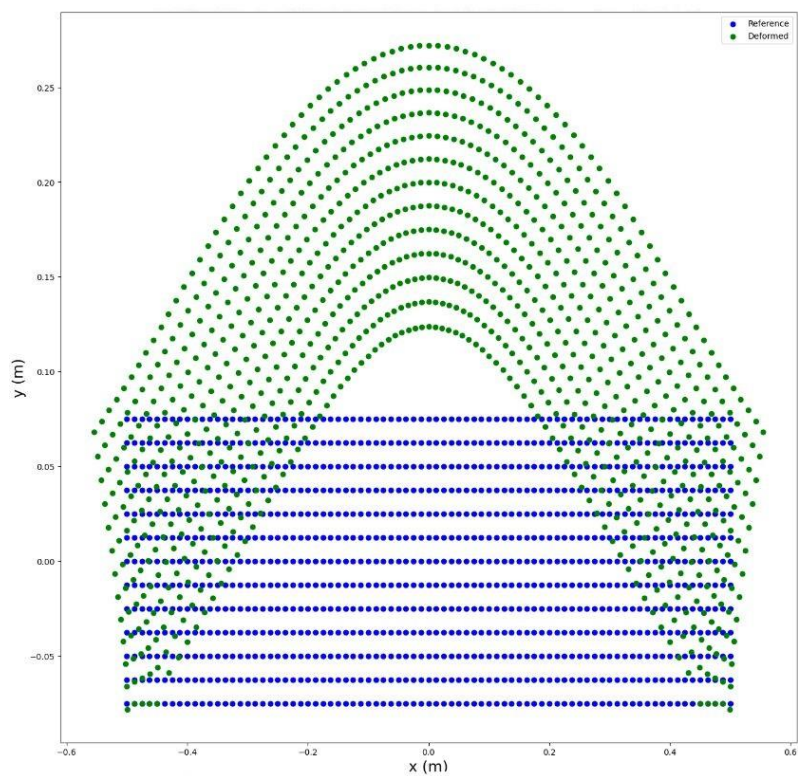
(b) Deslocamento na direção y

Fonte: elaborado pelo autor

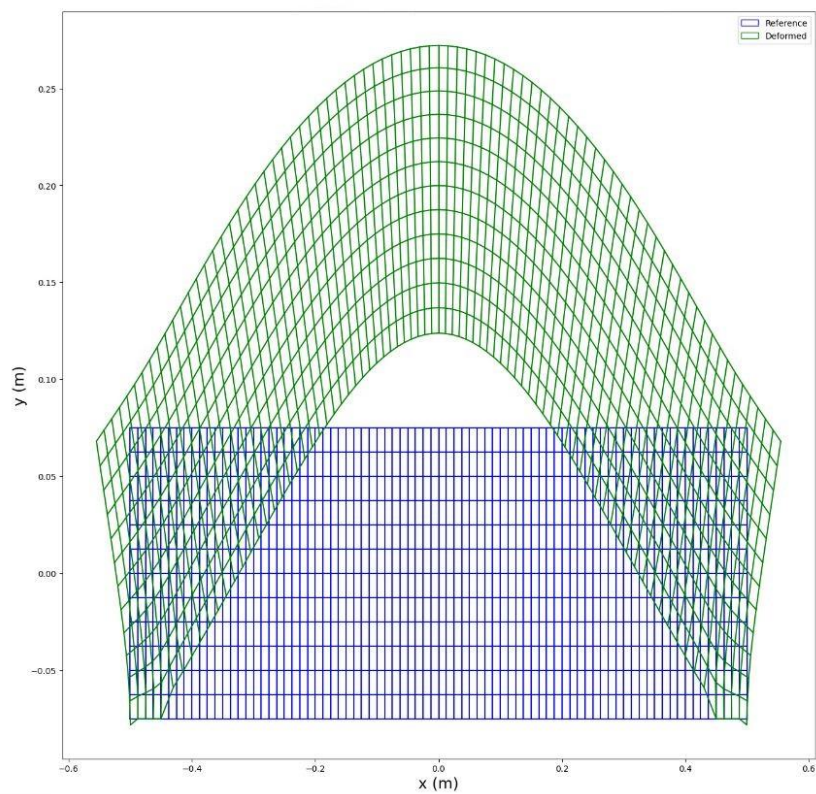
Ambos os gráficos mostram os deslocamentos decorrente da carga de tração aplicada, tanto no eixo x quanto no eixo y. No gráfico de deslocamentos na direção x o deslocamento máximo está localizado no canto direito superior da malha, tendo um valor de aproximadamente $u_x = 0.37 \text{ mm}$, enquanto que o canto superior esquerdo possui o menor deslocamento, sendo de $u_x = -0.37 \text{ mm}$. Já o gráfico dos deslocamentos em y, é possível notar um comportamento diferente, tendo um comportamento de parábola ao longo do eixo x. Tendo o maior deslocamento localizado no centro da malha, sendo aproximadamente $u_y = 1.3 \text{ mm}$, enquanto o menor deslocamento está localizado nas extremidades direita e esquerda da malha, em relação ao eixo x, tendo seu valor $u_y = -0.053 \text{ mm}$.

A Figura 16 mostra o deslocamento da malha após aplicação da tensão de tração, tanto em forma de malha quanto em forma de nós.

Figura 16 – Deslocamento da malha sem trinca inicial devido a carga



(a) Deslocamento da malha em forma de pontos

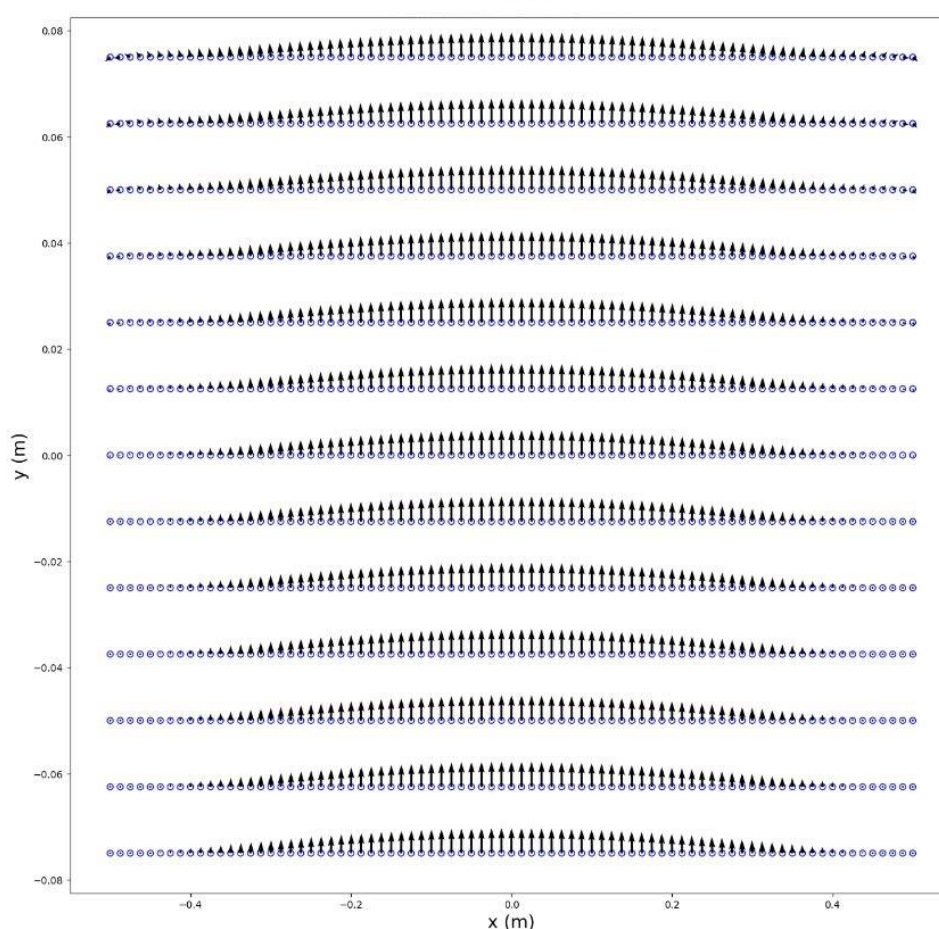


(b) Deslocamento da malha

Fonte: elaborada pelo autor

A malha foi deslocada em direção ao eixo positivo de y devido a carga de tração aplicada a ela. Os pontos onde foi prescrito deslocamento nulo não sofrem deformação, ficando mais nítido na Figura 16 (a). Também por causa das restrições de deslocamento, a malha se desloca em formato de arco, tendo os maiores deslocamentos no centro. Isso fica mais claro ao analisar a Figura 17.

Figura 17 – Representação vetorial dos deslocamentos da malha sem trinca inicial

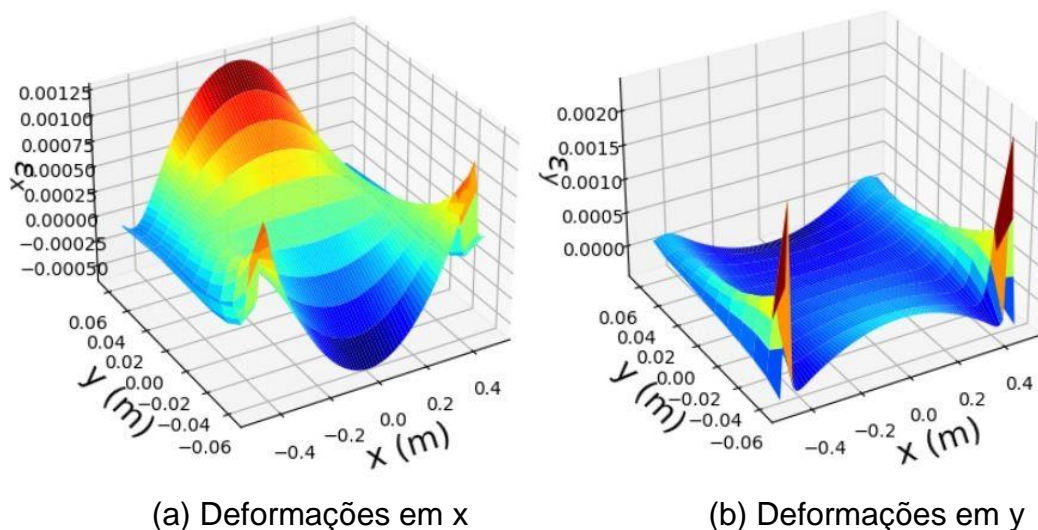


Fonte: elaborado pelo autor

Nela é possível notar os vetores representando o deslocamento dos pontos materiais. Os vetores possuem mais intensidade à medida que se aproximam do centro da malha em relação ao eixo x pois eles estão livres da condição de contorno do deslocamento prescrito.

A Figura 18 mostra as deformações da malha para as direções x, y .

Figura 18 – Gráfico de deformação da malha sem trinca inicial devido a carga de tração



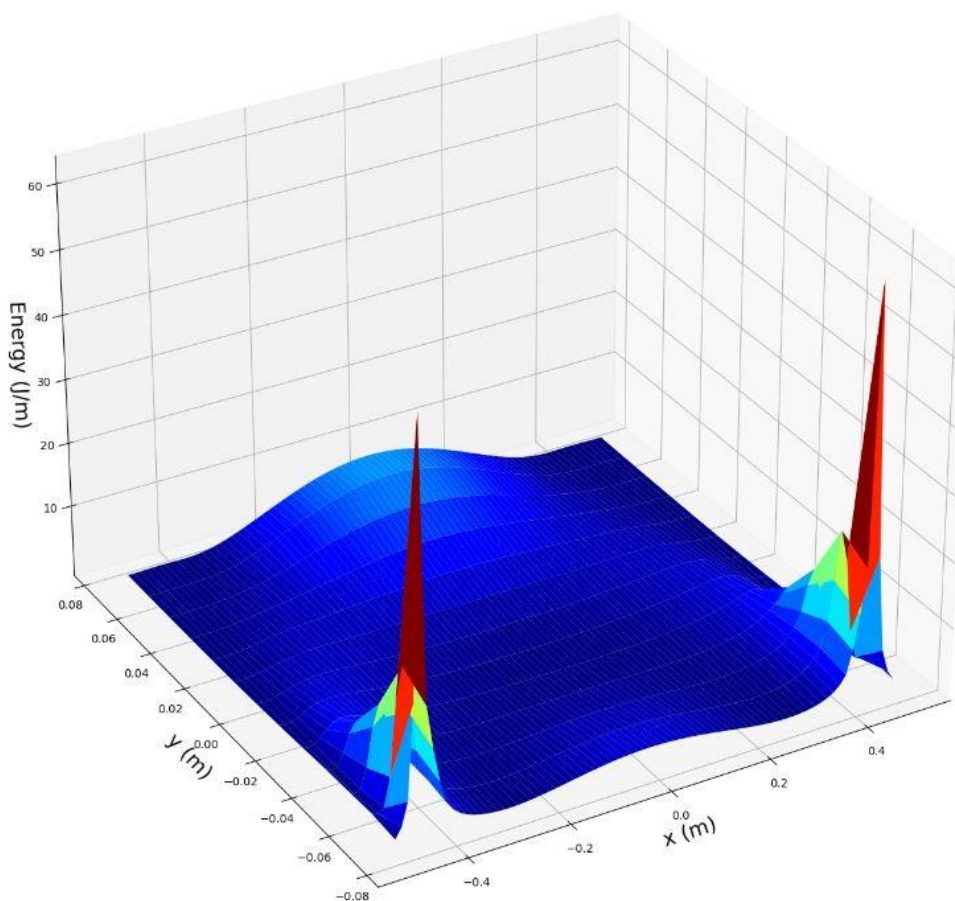
Fonte: elaborado pelo autor

Na Figura 18 (a), onde mostra a deformação na direção x (ϵ_x), é possível observar que os maiores valores (em módulo) estão concentradas ao longo da altura y da malha com $x = 0$, sendo o valor máximo é na altura $y = 0.06 \text{ m}$ e o valor mínimo na altura $y = -0.06 \text{ m}$. Os pontos onde estão as restrições de deslocamento também possuem altos valores de deformações.

A Figura 18(b) mostra as deformações na direção y (ϵ_y), nesse caso, os valores de maiores deformação estão concentrados nos pontos de restrições de deslocamento. Os valores vão aumentando a medida em que o eixo y tendo ao negativo. Nas extremidades os aumentos são mais consideráveis que no centro da malha.

Em relação a energia de deformação, a Figura 19 mostra a distribuição da densidade de energia de deformação ao longo da malha.

Figura 19 – Densidade de energia de deformação ao longo da malha para o caso sem trinca inicial



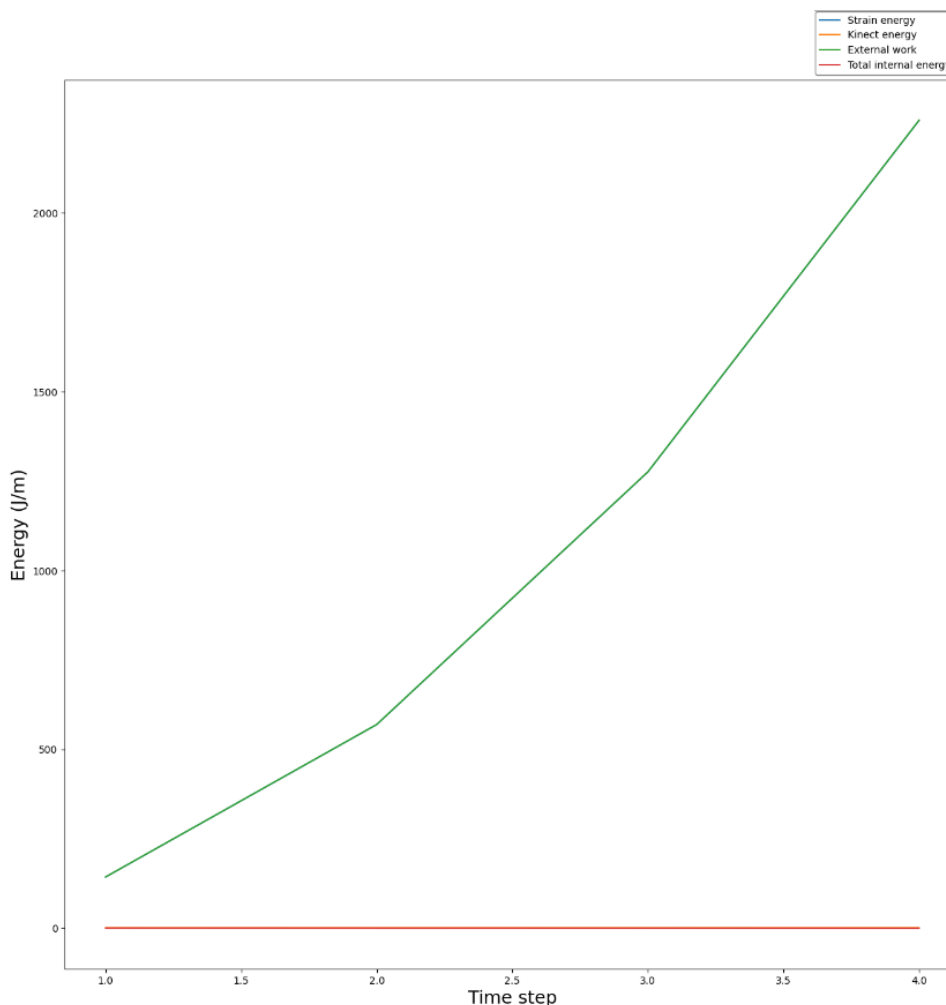
Fonte: elaborado pelo autor

Esse gráfico se assemelha a junção dos gráficos de deformações, na Figura 18, isso ocorre, pois, a densidade da energia de deformação é proporcional a deformação da malha. Destaca-se que os maiores valores de densidade de energia de deformação estão nos pontos de restrição de deslocamento, porém também se vê um aumento na extremidade superior da malha, onde estão aplicadas as tensões de tração. À medida que o valor do eixo y diminui a densidade de energia de deformação da malha tende a diminuir também até ficar praticamente constante.

Por fim, a Figura 20 mostra o gráfico da energia para cada passo na simulação. Nela, é visto que a energia de deformação e o trabalho externo realizado pela carga de

tração possuem o mesmo valor, a energia cinética é nula pois não ocorre desenvolvimento de trinca. Portanto a energia interna se torna nula.

Figura 20 – Evolução de energia para caso sem trinca inicial

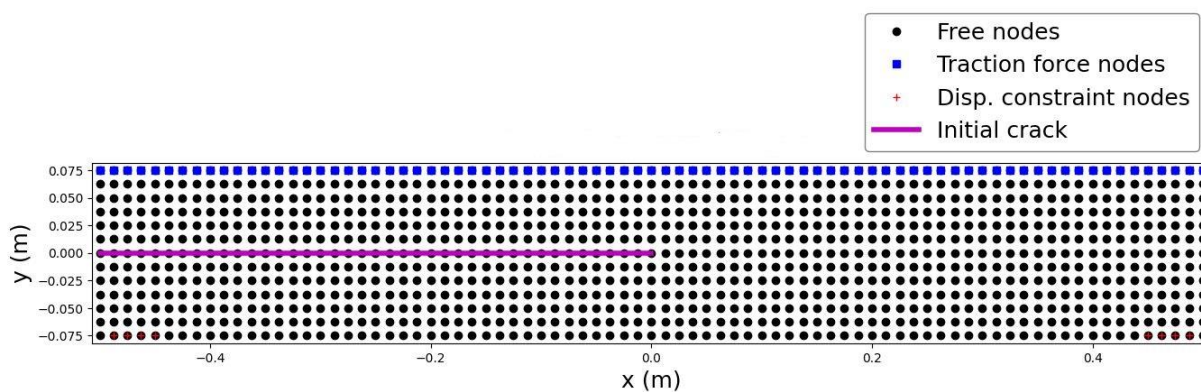


Fonte: elaborado pelo autor

5.2 Simulação com trinca inicial

Nessa seção será discutido os resultados obtidos pela simulação de uma placa de dimensões 1 x 0.15 m, semelhante a discutida na seção anterior, porém agora foi inserida uma trinca inicial. A trinca inicial possui 0.5 m na direção horizontal e está localizada em $y = 0$. As condições de contorno foram iguais as utilizadas na seção anterior mais o acréscimo da trinca inicial, ou seja, deslocamentos prescritos como nulos em pontos inferiores da malha e aplicação de tensão de tração de 6 MPa na direção y e em todos os pontos da última linha da malha, como visto na Figura 21.

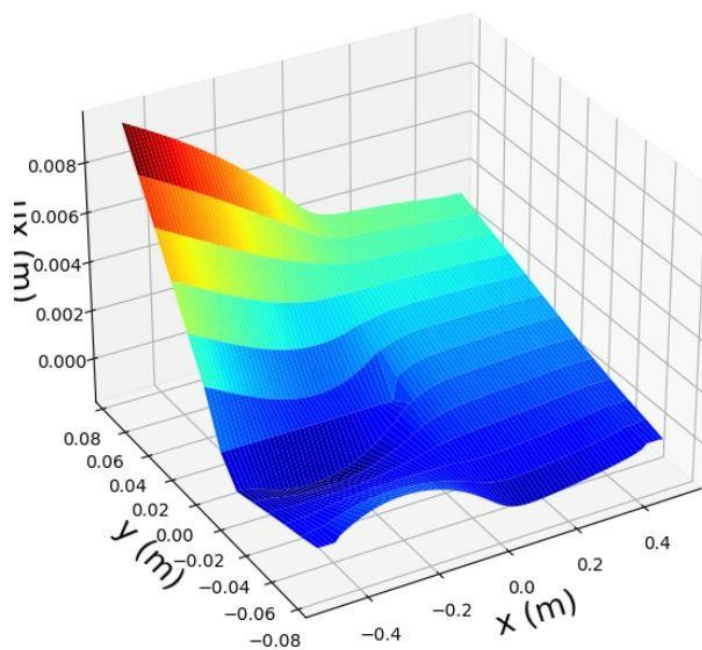
Figura 21 – Condições de contorno aplicadas a malha com a trinca inicial



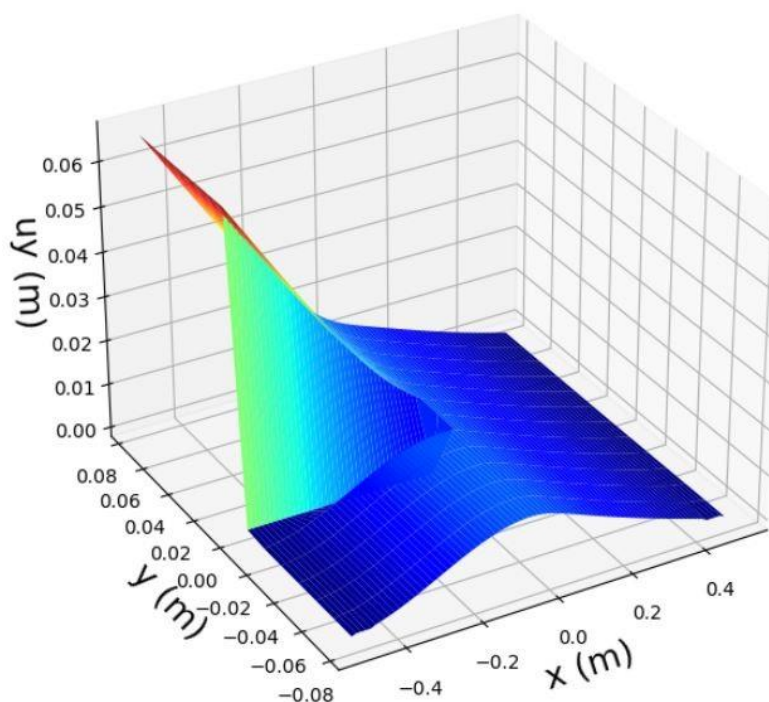
Fonte: elaborado pelo autor

A Figura 22 mostra o campo de deslocamento da malha tanto na direção x quanto na direção y.

Figura 22 – Campo de deslocamento da malha com trinca inicial



(a) Deslocamento na direção x



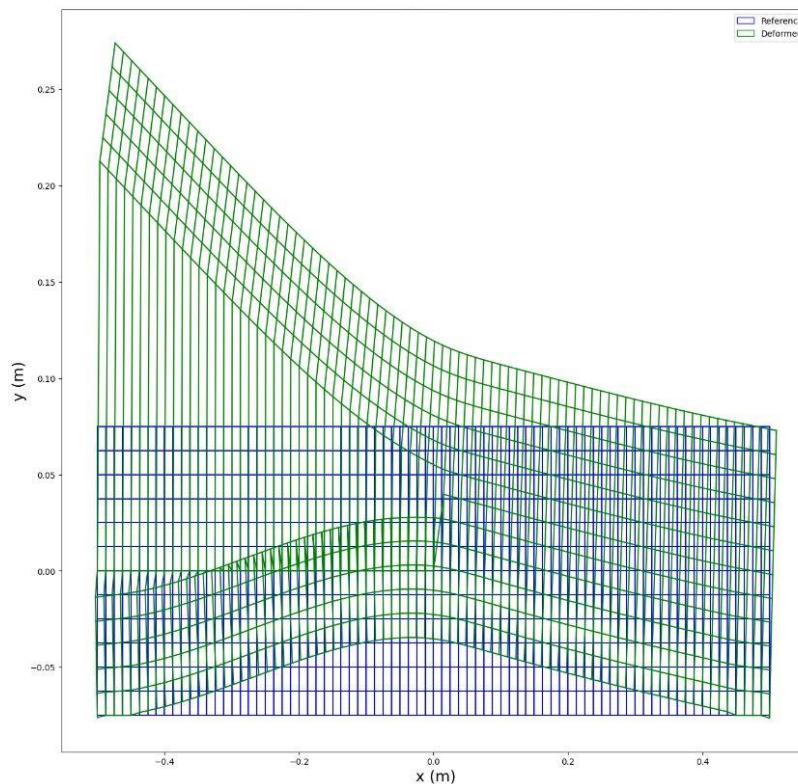
(b) Deslocamento na direção y

Fonte: elaborado pelo autor

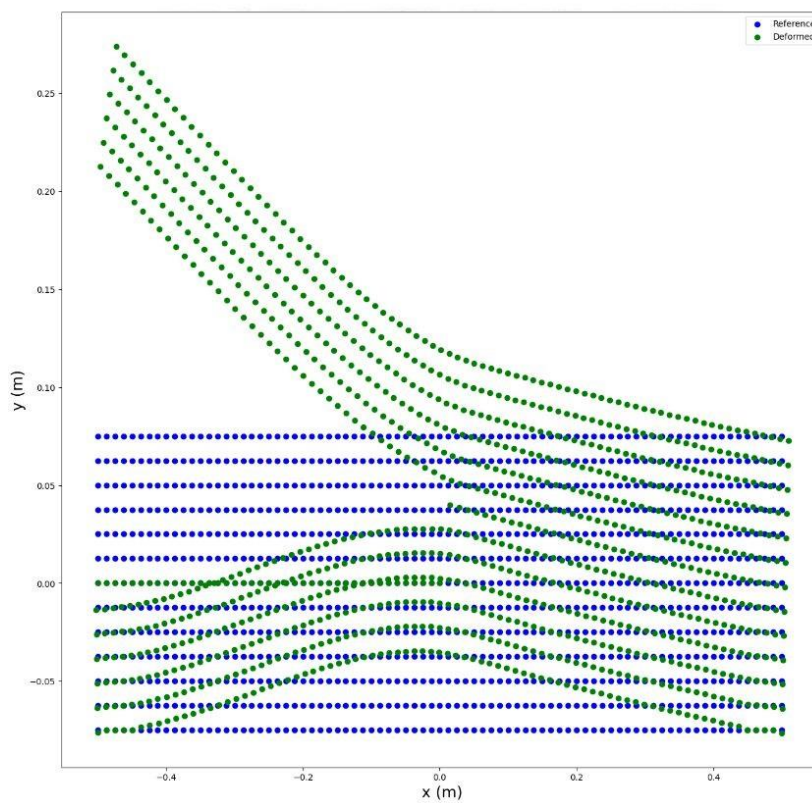
Logo no primeiro gráfico da Figura 22 já é possível notar diferenças quanto o campo de deslocamento mostrado na Figura 15, nele o maior deslocamento que a malha sofre ocorre no canto superior esquerdo, sendo $u_x = 0.0098 \text{ m}$ e vai diminuído à medida que y e x diminuem, sendo o menor deslocamento na região inferior da malha. Na Figura 22 (b), que mostra o campo de deslocamento da malha na direção y, o maior deslocamento também está no canto superior esquerdo da malha, sendo $u_y = 0.067 \text{ m}$ e diminui à medida que y e x diminuem, mas mais drasticamente. Tanto nos deslocamentos na direção x e em y estão com os maiores valores estão no canto superior direito da malha devido à inserção da trinca inicial que está na direita da malha.

A Figura 23 mostra o deslocamento da malha sofrido pela carga de tração.

Figura 23 – Deslocamento da malha com trinca inicial



(a) Campo de deslocamento da malha



(b) Campo de deslocamento da malha em pontos

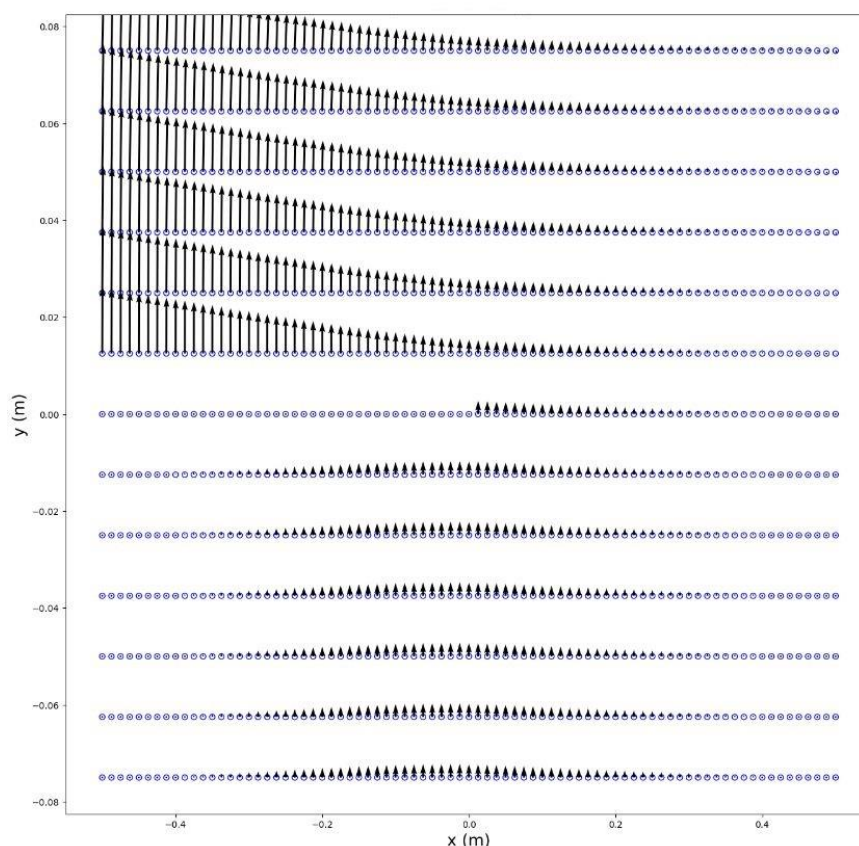
Fonte: elaborado pelo autor

É possível observar o maior deslocamento da malha no canto superior direito, como analisado anteriormente, e devido a carga de tração a malha é deslocada para o eixo y positivo. À medida que vai se afastando da trinca o deslocamento da malha fica menor. Nota-se, que os pontos materiais que representam a trinca inicial na malha não sofrem deslocamento, podendo ser melhor notado na Figura 23 (b).

Devido a trinca inicial, não existem mais ligações, ξ , entre os pontos cuja ligação passa pela trinca, portanto ocorre esse efeito da malha se “abrir” na região da trinca. É devido a esse efeito também que a deformação é maior na região superior à trinca e menor abaixo dela.

Na Figura 24, onde é mostrado os vetores de deslocamento de cada ponto da malha, mostrando que os pontos que estão logo acima da trinca inicial possuem as maiores intensidades de deslocamento em direção a y e diminuem com o aumento da distância da trinca.

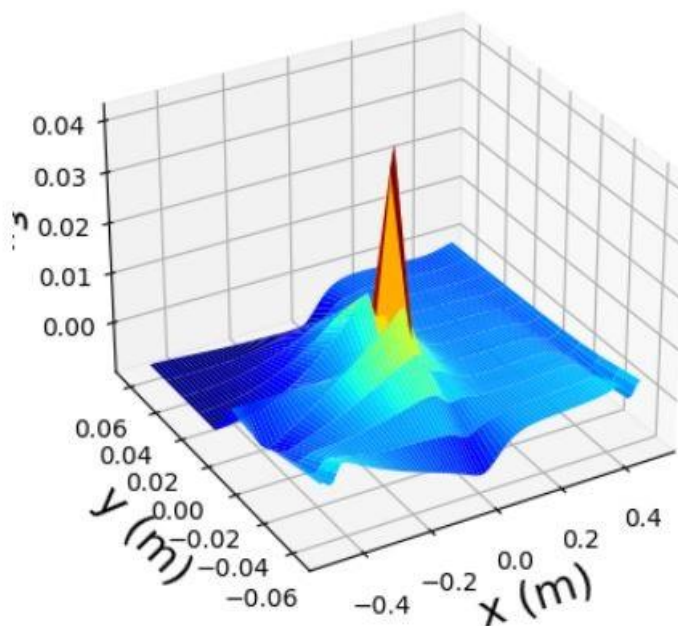
Figura 24 – Representação vetorial do deslocamento da malha com a trinca inicial



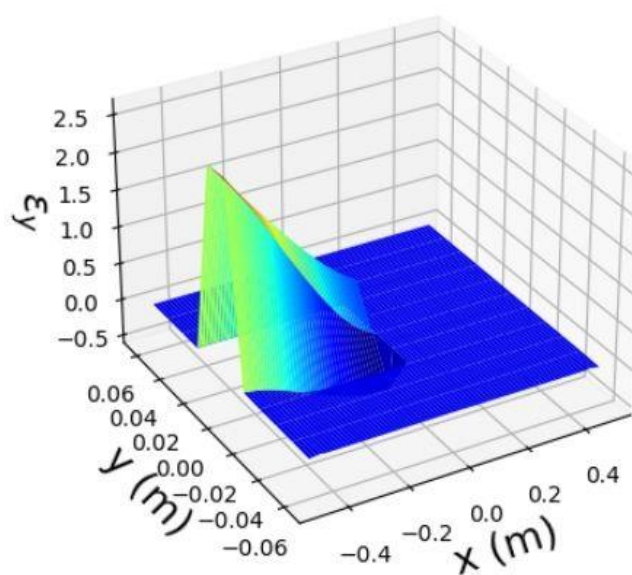
Fonte: elaborado pelo autor

A Figura 25 mostra o campo de deformação da malha em na direção do eixo y e x .

Figura 25 – Campo de deformação da malha com trinca inicial



(a) Campo de deformação em direção ao eixo x



(b) Campo de deformação em direção ao eixo y

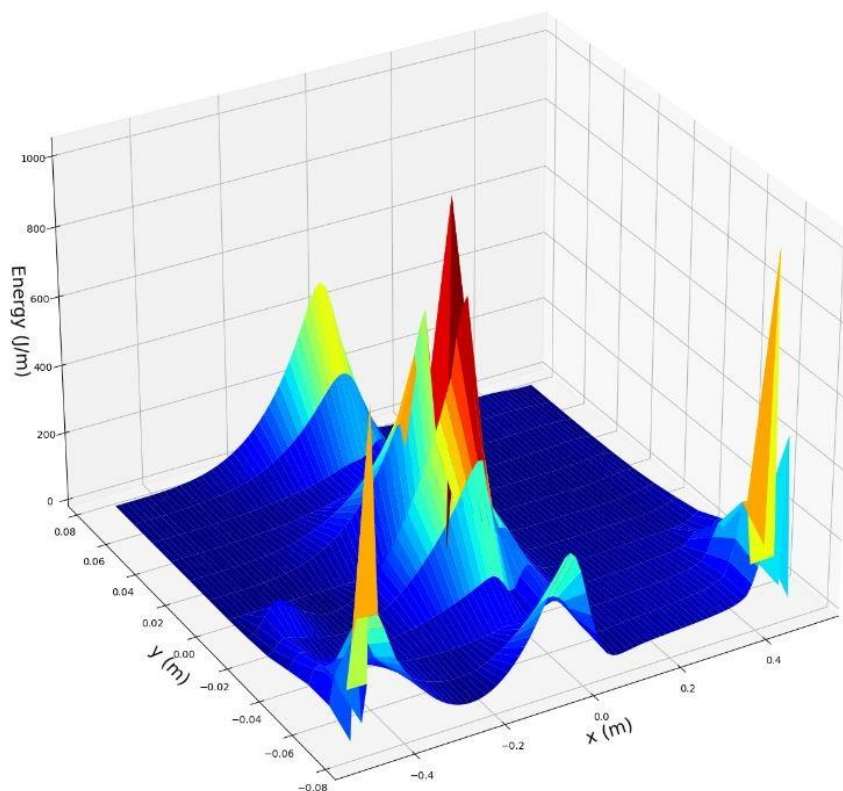
Fonte: elaborado pelo autor

A Figura 25 (a) mostra que a região central da malha, após a trinca inicial, foi onde ocorreu as maiores deformações na direção x (ε_x) e tendo variações por toda a malha com exceção do canto superior esquerdo que não sofreu deformação nessa direção. A Figura 25 (b) mostra que as deformações na direção y (ε_y) ficaram concentradas em volta da trinca inicial e à medida que se afasta da trinca elas diminuem até se tornarem nulas em todo o restante da malha.

Algo importante mencionar é que por causa da trinca inicial, as deformações ficaram concentradas em volta dela e, diferente do caso sem trinca, não tiveram altos valores de deformações nos pontos materiais onde foi determinado restrição de deslocamento.

A Figura 26 mostra a densidade de energia de deformação ao longo da malha em J/m.

Figura 26 – Campo de densidade de energia de deformação ao longo da malha com trinca inicial

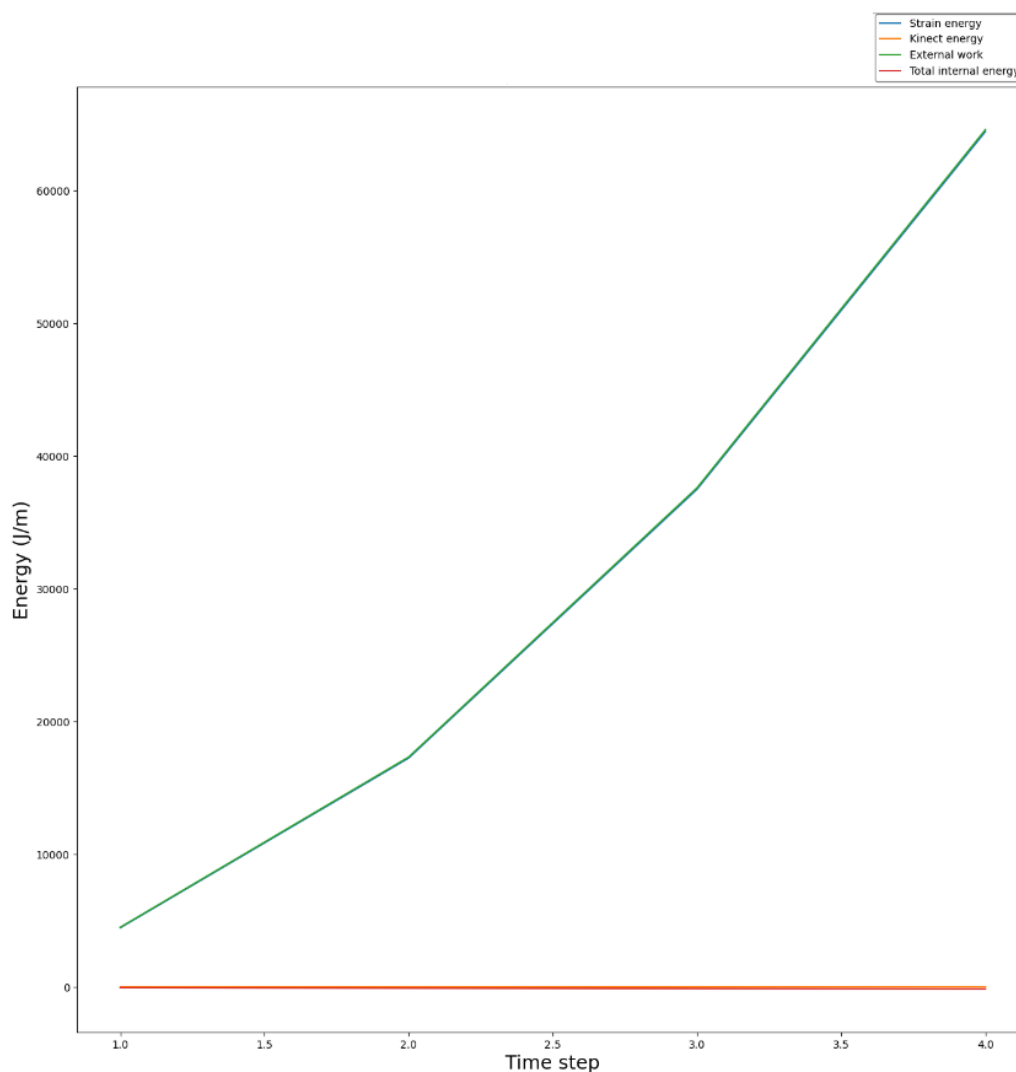


Fonte: elaborado pelo autor

O campo de densidade de energia de deformação mostra que no centro da malha é onde teve mais picos de valores, onde fixado $x = 0$ e percorrido por todo o eixo y . O centro da malha também é onde ocorre o maior valor de densidade de energia de deformação, logo após o fim da trinca inicial. Apesar de não ter tido alto valor de deformação nos pontos onde estão com restrição de deslocamento, o gráfico mostra que eles ainda possuem valores de energia de deformação significativos.

A Figura 27, mostra a evolução da energia com os ao longo dos passos, assim como na Figura 20, o balanço de energia se mantém como o esperado, com a energia de deformação tendo o mesmo valor do trabalho externo, e portanto, a energia interna total se mantém nula.

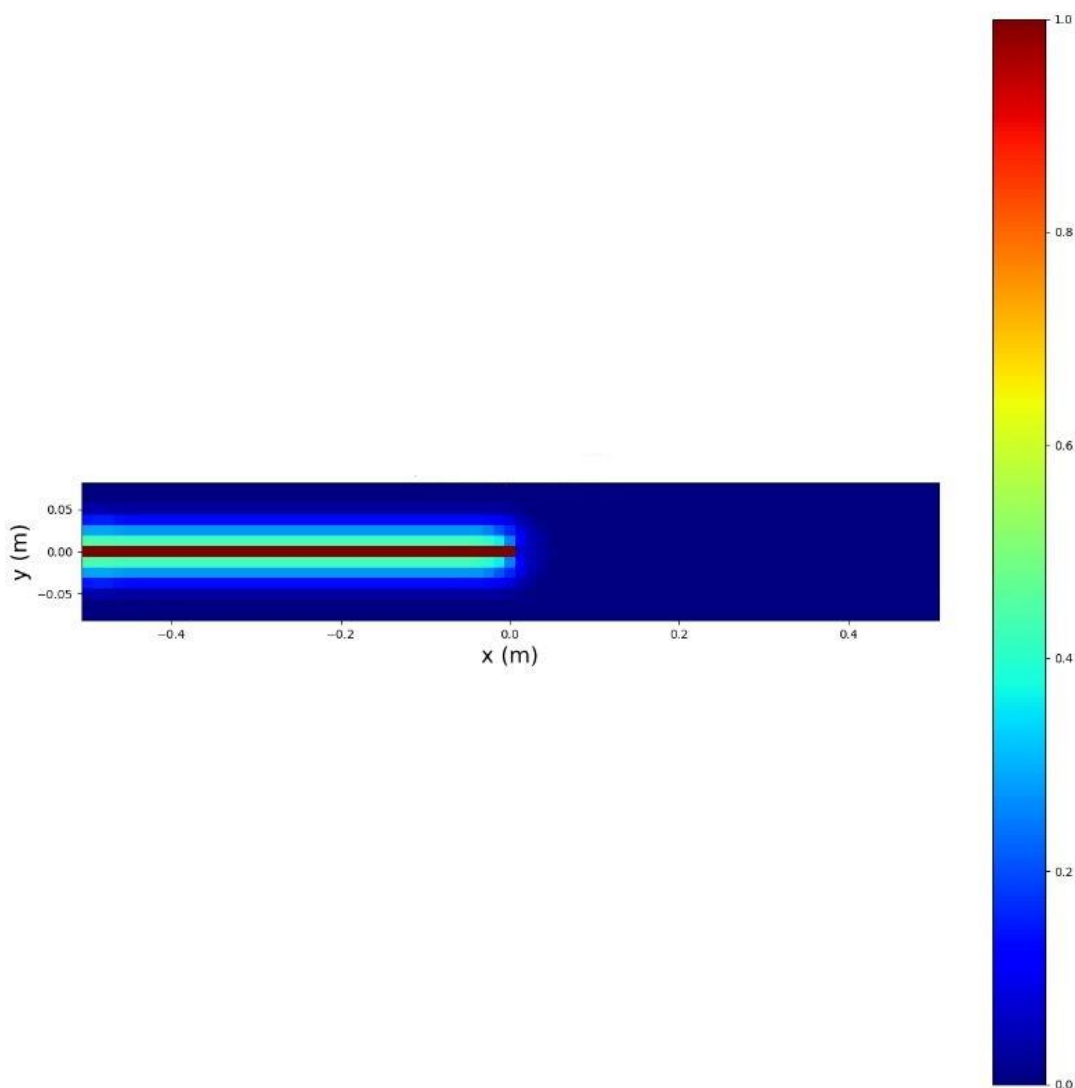
Figura 27 – Evolução de energia para o caso com trinca inicial



Fonte: elaborado pelo autor

Apesar de não considerar dano na simulação, foi possível plotar o campo de índice de deformação, visto na Figura 27. A trinca possui o índice de dano como 1 e quando se afasta dela seu valor vai diminuindo até se tornar zero e não ocorre sua propagação também.

Figura 28 – Índice dano ao longo da malha



Fonte: elaborado pelo autor

6 Conclusão

O objetivo desse trabalho foi aplicar a ferramenta peridinâmica, modelada em linguagem *Python*, em uma placa fina para duas situações, a primeira era aplicar carga de tração na placa sem a trinca inicial e a segunda é aplicar a carga de tração, mas inserindo uma trinca inicial.

A peridinâmica mostrou-se uma ótima ferramenta para análise de mecânica dos sólidos e uma alternativa ao método dos elementos finitos. Com seu código em *Python*, o programa torna-se mais leve para ser executado e mais acessível à outras máquinas computacionais. Além de ser uma linguagem de programação *open source* e podendo, portanto, ter sua utilização ainda mais acessível para mais pessoas e, também, passar por uma melhoria de seu código, ampliando sua utilidade para geometrias mais complexas.

Por ser uma ferramenta nova, ela se mostrou desafiadora, pois sofre com falta de informações na literatura, como o valor da taxa de liberação de energia G_0 que não foi possível encontrar valores para outros materiais, limitando seu uso. Outra barreira enfrentada foi a de compreender o código computacional para seu uso, apesar do código desenvolvido em *Python* ter facilitado seu uso e aplicação, ele ainda é complexo e como foi derivado do código desenvolvido em *MATLAB* por Patriota (2019), portanto muitas vezes foi necessário estudar ambos os códigos para maior clareza de algumas variáveis e para implementar o modelo constitutivo *PMB DTT* e também implementar corretamente o balanço de energia.

O *solver quasi-static* representou bem os deslocamentos e deformações sofridos pela malha, mesmo sendo um caso estático, e, portanto, mais simplificado. Com a inserção da trinca, mesmo sem considerar dano, ainda foi possível ter um resultado satisfatório. Em ambos os casos ocorreu o correto balanceamento das energias, ou seja, a energia de deformação anulou o trabalho externo da carga de tração aplicada. O modelo constitutivo *PMB DTT* utilizado não pode muito bem ser comparado e analisado pois quando não se considera dano na simulação, ele acaba recaindo ao modelo constitutivo *PMB*. Porém o resultado seria mais preciso caso fosse utilizado o *solver dynamic/explicit* e também considerando dano para analisar a propagação da trinca. Assim o *solver quasi-static* seria melhor utilizado para uma

simulação inicial e mais simplificada, que para ser mais detalhada seria preciso utilizar o *solver dynamic/explicit*.

Sugestões para futuros trabalhos:

- Utilizar o código para simular geometrias mais complexas, como inserção de orifícios.
- Aperfeiçoar o código para *Solver Dynamic/Explicit*.
- Implementar dano ao código.
- Aplicação do código para outros materiais.
- Expandir o código para simulações em 3D.

REFERÊNCIAS

- AZEVEDO, Á. F. M. Método dos elementos finitos. **Faculdade de Engenharia da universidade do Porto**, v. 1, n. 7, 2003.
- CHAIM, F. P. **DESENVOLVIMENTO DE UM MÓDULO PERIDINÂMICO BIDIMENSIONAL EM PYTHON**. Ilha Solteira: UNESP, 2022.
- DIAS, J. P. et al. A review of crack propagation modeling using peridynamics. **Probabilistic Prognostics and Health Management of Energy Systems**, p. 111–126, 2017.
- DU, Q.; TAO, Y.; TIAN, X. A peridynamic model of fracture mechanics with bond-breaking. **Journal of Elasticity**, v. 132, p. 197–218, 2018.
- ERINGEN, A. C.; EDELEN, D. G. B. On nonlocal elasticity. **International journal of engineering science**, v. 10, n. 3, p. 233–248, 1972.
- JAVILI, A. et al. Peridynamics review. **Mathematics and Mechanics of Solids**, v. 24, n. 11, p. 3714–3739, 2019.
- MADENCI, E.; OTERKUS, E. Peridynamic theory. Em: **Peridynamic theory and its applications**. [s.l.] Springer, 2014. p. 19–43.
- PATRIOTA, T. V. B. **Numerical investigation of damage models in two-dimensional peridynamics using MATLAB**. [s.l.: s.n.].
- SELESON, P. Improved one-point quadrature algorithms for two-dimensional peridynamic models based on analytical calculations. **Computer Methods in Applied Mechanics and Engineering**, v. 282, p. 184–217, 2014.
- SILLING, S. A. Reformulation of elasticity theory for discontinuities and long-range forces. **Journal of the Mechanics and Physics of Solids**, v. 48, n. 1, p. 175–209, 2000.
- SILLING, S. A. et al. Peridynamic states and constitutive modeling. 2007.
- SILLING, S. A. Linearized theory of peridynamic states. **Journal of Elasticity**, v. 99, p. 85–111, 2010.
- SILLING, S. A.; ASKARI, E. A meshfree method based on the peridynamic model of solid mechanics. **Computers & structures**, v. 83, n. 17–18, p. 1526–1535, 2005.

APÊNDICE A – CÓDIGO FONTE DO PROGRAMA E BIBLIOTECAS

O código pode ser encontrado no repositório do GitHub <https://github.com/gcmunhoz/Peridynamics>.

O código desenvolvido por Chaim (2022) pode ser encontrado no repositório do GitHub <https://github.com/FPChaim/peridynamics>.

É possível realizar o *download* do programa, que virá em uma pasta no formato *.zip* e então extraí-la, ou então clonar o repositório. Em seguida para usar o código é necessário fazer sua importação através de alguma *IDE* ou editor.

Para a instalação das bibliotecas externas utilizadas, foi utilizado o *pip do python*, que é um gerenciador de pacotes. Para instalar é necessário utilizar o comando *pip install* e em seguida o nome da biblioteca, como por exemplo *pip install numpy*.

O código desenvolvido por Patriota (2019) na linguagem de programação Matlab pode ser encontrado em https://bitbucket.org/tulio_vbp/pdlab/src/master/.

ANEXO A – ALGORITMO DO ESQUEMA *VELOCITY-VERLET*

Algoritmo 1: Esquema *Velocity-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: $a_i^{n+1} \leftarrow \frac{1}{\rho_i} (\sum_{j \in \mathcal{F}_i} f_{ij}^{n+1} \Delta V_{ij} + b_i^{n+1})$

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 QUASI-STATIC SOLVER

Algoritmo 2: Quase-Static solver

1: **procedimento:** Estimando a solução de passo de carregamento (\bar{u}^n)
2: **Passo 1:** Iniciando o vetor de deslocamento de grau de liberdade
3: $\bar{u}^0 = 0$;
4: **para** cada passo de carregamento n de 1 até n_{tot} **faça:**
5: **Passo 2:** carregar as condições de carregamento e a hipótese de solução
6:
7: $\bar{u}^n \leftarrow \bar{u}^{n-1}$
8: **Passo 3:** calcular o vetor residual, \mathbf{r} , e o residual r .
9: Determinar o critério de convergência para o passo de carregamento.
10: $\mathbf{r} \leftarrow \bar{\mathbf{f}}_{\text{model}}^n + \bar{\mathbf{b}}^n \bar{\mathbf{V}}^T$
11: $r \leftarrow \|\mathbf{r}\|_{\infty}$
12: $r_{\varepsilon} \leftarrow \varepsilon \max \left\{ \|\bar{\mathbf{f}}_{\text{model}}^n\|_{\infty}, \|\bar{\mathbf{b}}^n \bar{\mathbf{V}}^T\|_{\infty} \right\}$
13: **enquanto** $r > r_{max}$ **faça:**
14: **Passo 4:** calcular a matriz de rigidez tangente
15: $\mathbf{K}^n \leftarrow \text{tangentStiffness}(\bar{u}^n)$
16: **Passo 5:** calcular a solução incremental e atualizar a hipótese
17:
18: $\Delta \bar{u} = -(\mathbf{K}^n)^{-1} \mathbf{r}$
19: $\bar{u}^n = \bar{u}^n + \Delta \bar{u}$
20: $\mathbf{r} \leftarrow \bar{\mathbf{f}}_{\text{model}}^n + \bar{\mathbf{b}}^n \bar{\mathbf{V}}^T$
21: $r \leftarrow \|\mathbf{r}\|_{\infty}$
22: **fim enquanto**
23: **fim para**
24: **fim procedimento**

ANEXO C – ALGORITMOS DE FORÇA DE INTERAÇÃO IMPLEMENTADAS

Algoritmo 3: Cálculo da densidade da força de interação para o modelo PMB

```

1: procedimento Avaliar a densidade da força interna  $f(x)_{int}$ 
2:   para cada nó  $i$  faça
3:     ▷ Inicializar a densidade de força interna
4:      $f_{int,i} \leftarrow 0$ 
5:     para cada nó  $j$  em  $\mathcal{F}_i$  faça
6:        $\xi_{ij} \leftarrow x_j - x_i$ 
7:        $\eta_{ij} \leftarrow u_j - u_i$ 
8:        $s_{ij} \leftarrow \frac{|\xi_{ij} + \eta_{ij}| - |\xi_{ij}|}{|\xi_{ij}|}$ 
9:        $e_{ij} \leftarrow \frac{\xi_j + \eta_j}{|\xi_j + \eta_j|}$ 
10:      ▷ Atualizar o alongamento máximo
11:      se  $s_{ij} > s_{ij,max}$  então
12:         $s_{ij,max} \leftarrow s_{ij}$ 
13:      fim se
14:       $\mu_{ij} \leftarrow \mu(s_{ij,max})$ 
15:       $f_{int,i} \leftarrow f_{int,i} + C_{PMB} \frac{\omega(|\xi_{ij}|)}{|\xi_{ij}|} \mu_{ij} s_{ij} \Delta V_{ij} e_{ij}$ 
16:    fim para
17:  fim para
18:  Retornar  $f_{int}$ 
19: --fim procedimento

```

Algoritmo 3: Cálculo da densidade da força de interação para o modelo DTT

```

1: procedimento Avaliar a densidade da força de interação  $f_{int}$ 
2:   para cada nó  $i$  faça
3:     ▷ Inicializar a densidade de força interna
4:      $f_{int,i} \leftarrow 0$ 
5:     para cada nó  $j$  faça
6:        $\xi_{ij} \leftarrow x_j - x_i$ 
7:        $\eta_{ij} \leftarrow u_j - u_i$ 
8:        $s_{ij} \leftarrow \frac{|\xi_{ij} + \eta_{ij}| - |\xi_{ij}|}{|\xi_{ij}|}$ 
9:        $e_{ij} \leftarrow \frac{\xi_j + \eta_j}{|\xi_j + \eta_j|}$ 
10:      ▷ Atualizar o alongamento relativo máximo
11:      se  $s_{ij} > s_{ij,max}$  então
12:         $s_{ij,max} \leftarrow s_{ij}$ 
13:      fim se
14:       $\mu_{ij} \leftarrow \mu(s_{ij,max})$ 
15:       $F_{ij} \leftarrow F_{ij}$ 
16:       $f_{int,i} \leftarrow f_{int,i} + C_{PMB} \frac{\omega(|\xi_{ij}|)}{|\xi_{ij}|} \mu_{ij} F_{ij} \Delta V_{ij} e_{ij}$ 
17:    fim para
18:  fim para

```

19: Retornar f_{int}
20: **fim procedimento**

ANEXO D – ALGORITMOS DE CÁLCULO DE ÁREA PARCIAL

Algoritmo 4: FA

```

1:  $\|\xi\| = \|x_j - x_i\|$  ▷ Computar as distâncias entre os centros das células
2: if  $\|\xi\| \leq \delta$  then ▷ Verificar se o ponto  $j$  pertence à família de  $i$ 
3:    $A_j^{(i)} = h^2$ 
4: else
5:    $A_j^{(i)} = 0$ 
6: end if
7: return  $A_j^{(i)}$ 

```

Algoritmo 5: PA-AC interação com a família

```

1:  $N_x = \left\lceil \frac{\delta}{h} + \frac{1}{2} - \epsilon \right\rceil$  ▷ Computar o máximo de vizinhos possível ao longo da direção x
2: for  $k = i - N_x : i + N_x$  do
3:   if  $k = i$  then
4:      $N_y = N_x$ 
5:   else
6:      $|\xi_1| = |x_j - x_i|$ 
7:      $N_y = \left\lceil \frac{\sqrt{\delta^2 - (|\xi_1| - \frac{h}{2})^2}}{2} + \frac{1}{2} - \epsilon \right\rceil$ 
8:   end if
9:   for  $l = j - N_y : j + N_y$  do
10:    Computar interações
11:   end for
12: end for

```

Algoritmo 6: PA-AC

```

Mapear a célula para o quadrante superior direito:
1: if  $x_j < x_i$  then
2:    $x_j = x_i + (x_i - x_j)$  ▷ Transladar a célula para a direita
3: end if
4: if  $y_j < y_i$  then
5:    $y_j = y_i + (y_i - y_j)$  ▷ Transladar a célula para a cima
6: end if
7: if  $x_j = x_i$  then ▷ Rotacionar a célula do eixo +y para o eixo +x
8:    $\Delta = y_j - y_i$ 
9:    $y_j = y_i$ 
10:   $x_j = x_i + \Delta$ 
11: end if

Contar os números de cantos de  $\tau_j$  dentro da vizinhança de  $i$ :
12: counter = 0
13: for  $n = -1, 1$  do
14:   for  $m = -1, 1$  do
15:    if  $(x_j + n\frac{h}{2} - x_i)^2 + (y_i + m\frac{h}{2} - y_i)^2 < \delta^2$  then
16:      counter = counter + 1

```

```

17:   end if
18: end for
19: end for
    Computar área parcial
20: if counter = 4 then
21:    $A_j^{(i)} = h^2$            ▷Caso I
22: else if counter = 3 then
23:   Computar  $A_j^{(i)}$  com Caso II
24: else if counter = 2 then
25:   if  $y_j = y_i$  then
26:     if  $x_j + \frac{h}{2} \geq x_i + \delta$  then
27:       Computar  $A_j^{(i)}$  como o Caso III(a2)
28:     else
29:       Computar  $A_j^{(i)}$  com Caso III(b)
30:     end if
31:   else
32:      $r_{br}^2 = \left(x_j + \frac{h}{2} - x_i\right)^2 + \left(y_j + \frac{h}{2} - y_i\right)^2$  ▷Distância ao quadrado do canto inferior direito
    de  $i$  até  $\tau_j$ 
33:     if  $r_{br}^2 > \delta^2$  then
34:       Computar  $A_j^{(i)}$  com o Caso III(a1)
35:     else
36:       Computar  $A_j^{(i)}$  com o Caso III©
37:     end if
38:   end if
39: else if counter = 1 then
40:   Computar  $A_j^{(i)}$  com o Caso IV
41: else
42:   if  $x_j = \frac{h}{2} < x_i + \delta$  then
43:     Computar  $A_j^{(i)}$  com o Caso V
44:   else
45:      $A_j^{(i)}$ 
46:   end if
47: end if
48: return  $A_j^{(i)}$ 

```
