



UNIVERSIDADE FEDERAL DA
INTEGRAÇÃO LATINO-AMERICANA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA CIVIL (PPGECI)

MODELAGEM DO MEIO CONTÍNUO BIDIMENSIONAL ATRAVÉS DE
REDE TRELIÇADA PARA ESTRUTURAS GEOMETRICAMENTE NÃO
LINEARES UTILIZANDO UMA ESTRATÉGIA DE DISCRETIZAÇÃO

EDMUNDO SAHD NETO

Foz do Iguaçu

2025



UNIVERSIDADE FEDERAL DA
INTEGRAÇÃO LATINO-AMERICANA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA CIVIL (PPGECI)

MODELAGEM DO MEIO CONTÍNUO BIDIMENSIONAL ATRAVÉS DE
REDE TRELIÇADA PARA ESTRUTURAS GEOMETRICAMENTE NÃO
LINEARES UTILIZANDO UMA ESTRATÉGIA DE DISCRETIZAÇÃO

EDMUNDO SAHD NETO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal da Integração Latino-Americana, como requisito parcial à obtenção do título de Mestre em Engenharia Civil.

Orientador: Prof. Dr. Aref Kalilo Lima Kzam

Foz do Iguaçu

2025

Catálogo elaborado pelo Setor de Tratamento da Informação
Catálogo de Publicação na Fonte. UNILA - BIBLIOTECA LATINO-AMERICANA - CENTRAL

S131m

Sahd Neto, Edmundo.

Modelagem do meio contínuo bidimensional através de rede treliçada para estruturas geometricamente não lineares utilizando uma estratégia de discretização / Edmundo Sahd Neto. - Foz do Iguaçu, 2025.
83 fls.: il.

Universidade Federal da Integração Latino-Americana, ILATIT, Programa de Pós-Graduação em Engenharia Civi.

Orientador: Aref Kalilo Lima Kzam.

1. Materiais - Propriedades. 2. Treliças (Construção civil). 3. Cinemática. I. Kzam, Aref Kalilo Lima. II. Título.

CDU 620.19

EDMUNDO SAHD NETO

**MODELAGEM DO MEIO CONTÍNUO BIDIMENSIONAL ATRAVÉS DE
REDE TRELIÇADA PARA ESTRUTURAS GEOMETRICAMENTE NÃO
LINEARES UTILIZANDO UMA ESTRATÉGIA DE DISCRETIZAÇÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal da Integração Latino-Americana, como requisito parcial à obtenção do título de Mestre em Engenharia Civil.

BANCA EXAMINADORA

Prof. Dr. Aref Kalilo Lima Kzam (Orientador)
UNILA

Prof. Dr. André Jacomel Torii
UNILA

Prof.a Dra. Carolina Quintero Ramirez
UNIFEI

Foz do Iguaçu

2025

DEDICATÓRIA

Para Lílian...

AGRADECIMENTOS

À minha Esposa Lílian de Melo Alencar Sahd, meus pais, irmão e a toda minha família que, apesar dos percalços, sempre estiveram ao meu lado me apoiando incondicionalmente.

Ao meu orientador e amigo Prof. Dr. Aref Kalilo Lima Kzam por todo o ensinamento, paciência, dedicação e apoio, elementos sem os quais este trabalho não teria se realizado.

À todo o corpo docente e administrativo do Programa de Pós-Graduação em Engenharia Civil-PPGECI da UNILA por todo o apoio prestado

À equipe e amigos da Prefeitura Universitária da UNILA, sempre presentes e dispostos a ajudar

A UNILA por todo o apoio, em especial, pela liberação das atividades laborais, algo tão importante neste período onde a dedicação exclusiva se faz tão necessária.

“A matemática é a linguagem com a qual
Deus escreveu o universo”
Galileo Galilei

RESUMO

As redes treliçadas são um arranjo ordenado, composto por partes sólidas (elementos de treliça) e vazios. Essas estruturas são usadas para obter um metamaterial dado seu arranjo geométrico e leveza, fundamentais em aplicações estruturais. Este trabalho apresenta um método de homogeneização para sólidos bidimensionais usando células periódicas, a partir do elemento finito de treliça não linear geométrica. Com base na distribuição das células unitárias de treliças é possível descrever de maneira equivalente as propriedades do meio contínuo. A cinemática dessas células é determinada usando a posição em vez das deformações das barras. Emprega-se o modelo constitutivo de Saint-Venant-Kirchhoff, que relaciona a deformação de Green-Lagrange ao tensor de tensão de segunda espécie de Piola-Kirchhoff. A novidade do modelo consiste em propor uma disposição periódica de treliças capaz de representar o comportamento mecânico de sólidos bidimensionais. A influência do fator de escala correspondente ao tamanho das células unitárias e as técnicas de ajuste do modelo são apresentadas. Para resolver o problema não linear, o método de Newton-Raphson será usado com controle de força e deslocamento. Exemplos são apresentados demonstrando a aplicabilidade do modelo.

Palavras-chave: Metamateriais; Homogeneização; Lattice-truss; MEFP.

ABSTRACT

Lattice trusses are an ordered arrangement of solid parts (truss elements) and voids. These structures are used to obtain a metamaterial due to their geometric arrangement and lightness, which are fundamental in structural applications. This work presents a homogenisation method for two-dimensional solids using periodic cells from the geometric nonlinear finite truss element. Based on the distribution of the unit cells of the trusses, it is possible to describe the properties of the continuous medium in an equivalent way. The kinematics of these cells is determined by the position rather than the deformation of the beams. The Saint-Venant-Kirchhoff constitutive model is used, which relates the Green-Lagrange deformation to the Piola-Kirchhoff second species stress tensor. The novelty of the model is to propose a periodic arrangement of trusses capable of representing the mechanical behaviour of two-dimensional solids. The influence of the scale factor corresponding to the size of the unit cells and the model fitting techniques are presented. The Newton-Raphson method with force and displacement control is used to solve the non-linear problem. Examples are presented and compared with the literature and consolidated analytical solutions to demonstrate the applicability of the model.

Keywords: metamaterials; homogenization; lattice-truss;

LISTA DE ILUSTRAÇÕES

Figura 1 - Evolução das publicações a respeito das propriedades mecânicas em estruturas treliçadas	15
Figura 2 - Fluxograma para o método de Newton-Rahpson	26
Figura 3 - Mapeamento cinemático do elemento finito de chapa quadrilateral . . .	28
Figura 4 - Células unitárias propostas	35
Figura 5 - Elementos compartilhados	36
Figura 6 - Decomposição da malha para o cálculo dos coeficientes	37
Figura 7 - Convergência devido ao refinamento da malha em viga engastada . . .	42
Figura 8 - Viga em balanço com carga na extremidade	43
Figura 9 - Comparação das células unitárias para viga em balanço - $P=0,5$ KN .	45
Figura 10 - Comparação das células unitárias para viga em balanço - $P=100$ KN .	46
Figura 11 - Convergência em função da densidade de malha	47
Figura 12 - Pórtico de Lee	48
Figura 13 - Pórtico de Lee - Gráfico força x deslocamento	49
Figura 14 - Colapso local devido ao passo de carregamento	49
Figura 15 - Arcos testados	50
Figura 16 - Comparação das células unitárias para os arcos H1, H2 e H3	51
Figura 17 - Colapso do arco H3 de malha E2	51
Figura 18 - Coluna submetida a carregamento excêntrico	52
Figura 19 - Deslocamento horizontal normalizado em função da carga	53

LISTA DE TABELAS

Tabela 1 - Fatores m	38
Tabela 2 - Dados das malhas 100 mm, 50 mm e 25 mm para a viga engastada . .	44
Tabela 3 - Dados das malhas 10 mm e 5 mm para a viga engastada	44
Tabela 4 - Parâmetros adotados para o Pórtico de Lee	47
Tabela 5 - Dados das malhas geradas para os Arcos	50
Tabela 6 - Dados das malhas geradas para a coluna	53

LISTA DE ABREVIATURAS E SIGLAS

MEF	Método de Volumes Finitos
MEFP	Método dos elementos finitos posicional
SVK	Saint-Venant Kirshhoff
NLG	Não Linear Geométrico
RVE	Elemento de Volume Representativo

LISTA DE SÍMBOLOS

t	Espessura
Π	Funcional da energia potencial total
H	Matriz Hessiana
X	Coordenadas iniciais
Y	Coordenadas atualizadas
P	Energia potencial devido as cargas externas, Carregamento
U	Energia potencial de deformação
u_e	Energia específica de deformação
$F_{\text{ext}}^{\vec{}}$	Vetor forças externas
E	Módulo de elasticidade longitudinal
\mathbb{E}	Deformação de Green-Lagrange
l	Comprimento inicial
l_0	Comprimento final
v	Volume
A	Área
A_0	Área inicial
G	Função resíduo
σ	Tensão normal
ϵ	Deformação específica
D	Matriz de rigidez
ν	Coefficiente de Poisson
ϕ	Função de forma
ξ	Coordenadas adimensionais
N	Número de nós
J	Matriz Jacobiana
S	Tensor de Piolla-Kirschhoff
\mathbb{C}	Tensor constitutivo
f_k	Coefficiente de forma

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Hipótese	15
1.2	Objetivos Gerais	16
1.3	Objetivos Especificos	16
1.4	Justificativa	16
1.5	Estrutura do trabalho	16
2	REVISÃO BIBLIOGRÁFICA	17
3	FORMULAÇÃO DO ELEMENTO FINITO POSICIONAL	21
3.1	Apresentação do Capítulo	21
3.2	O Método dos Elementos Finitos Posicional (MEFP)	21
3.3	Energia Potencial Total e o Princípio da Estacionariedade	21
3.4	Não-Linearidade Geométrica	23
3.5	MÉTODO DOS ELEMENTOS FINITOS POSICIONAL APLICADO AO PROBLEMA DE TRELIÇAS PLANAS	23
3.5.1	Aplicação do princípio da estacionariedade	25
3.5.2	Aplicação do Método de Newton-Rahpson	25
3.6	MÉTODO DOS ELEMENTOS FINITOS DE CHAPA PLANA	26
3.7	Relação tensão-deformação e estado plano de tensões	26
3.7.1	Cinemática do elemento finito de chapa quadrilateral (Mapeamento)	27
3.7.2	Tensor Gradiente da Função Mudança de Configuração	28
3.8	Modelo Constitutivo para Elementos de Chapa (Modelo de Saint-Venant-Kirchhoff)	29
3.8.1	Considerações de Estado Plano de Tensões / Estado Plano de Deformações	29
3.9	Derivação das Forças Internas	30
3.10	Derivação da Matriz Hessiana	31
4	REPRESENTAÇÃO DO MEIO CONTÍNUO UTILIZANDO ESTRUTURA TRELIÇADA	32
4.1	Apresentação do capítulo	32
4.2	Representação do meio contínuo utilizando estruturas treliçadas	32
4.3	Homogenização	32
4.4	Materiais	33
4.4.1	Gerador de Malha	33
4.4.2	Otimização do Código em Fortran Para Elementos Treliçados	33
4.4.3	Modelagem no software ABAQUS	34
4.4.4	Células unitárias propostas	34
4.5	Metodologia	34

4.5.1	Aplicação de carregamentos ou deslocamentos	35
4.6	Determinação do elemento treliçado equivalente	35
4.6.1	Propriedades do material	37
4.6.2	Coeficiente de forma da estrutura(f_k)	37
4.7	Equivalência energética treliça/contínuo	38
4.7.1	Energia de deformação da chapa	38
4.7.2	Energia de deformação da treliça	39
4.7.3	Aplicação da equivalência energética	41
4.8	Fechamento	41
5	EXEMPLOS E ANÁLISE DOS RESULTADOS	42
5.1	Apresentação do Capítulo	42
5.2	Resultados obtidos pelo método iterativo	42
5.2.1	Viga engastada com carga concentrada	42
5.3	Resultados obtidos com o processo de equivalência energética	43
5.3.1	Exemplo 1 - Viga engastada com carga na extremidade livre	43
5.3.2	Exemplo 2 - Pórtico de Lee	47
5.3.3	Exemplo 3 - Arco	50
5.3.4	Exemplo 4 - Coluna com carregamento excêntrico	52
5.3.5	Fechamento	53
6	CONCLUSÃO	54
	REFERÊNCIAS	55
	APÊNDICE A – Código para geração de malha e Discretização	59
A.1	Gerador de malha	59

1 INTRODUÇÃO

O cálculo estrutural é provavelmente um dos mais importantes elementos em um projeto de engenharia, pois o correto dimensionamento destas estruturas impacta diretamente em requisitos como confiabilidade, segurança, economia, entre outros. A introdução de novas tecnologias, sejam elas a adoção de novas técnicas construtivas ou a adoção de novos materiais permitiu a quebra de paradigmas levando os projetos estruturais à níveis até então considerados impossíveis.

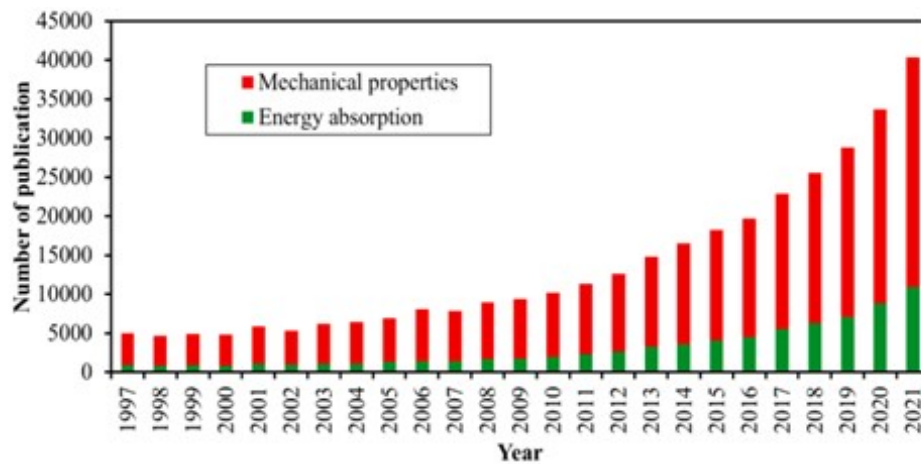
Materiais que antigamente eram muito limitados por diversos fatores hoje são frutos de pesquisa e desenvolvimento. Entre estes avanços é possível citar o desenvolvimento dos chamados metamateriais, uma classe de materiais não naturais, cujas propriedades podem ser manipuladas de modo a se obter um comportamento específico desejado como, por exemplo, um melhor desempenho estrutural. Para Zhang et al. (2020) as propriedades dos metamateriais dependem muito mais de sua geometria do que de sua composição.

A utilização de estruturas treliçadas periódicas, também chamadas *lattice-truss*, apresentam diversas vantagens como, elevada relação resistência/peso, adaptabilidade e eficiência estrutural. Estudos como os de Gu, Shterenlikht e Pavier (2019), Romijn e Fleck (2007) e Zhang et al. (2009) indicam que estas estruturas periódicas costumam apresentar propriedades únicas, possuindo aplicabilidade nas mais diversas áreas. Spina et al. (2023) investigou as propriedades mecânicas de materiais extrudados que foram modelados como uma estrutura treliçada. O modelo passou por um processo de homogeneização de suas propriedades. As simulações indicaram que tanto o modelo treliçado quanto o modelo homogeneizado apresentaram a mesma resposta mecânica. Miao et al. (2024) apresentam dados demonstrando que, apenas no repositório da editora Elsevier, as publicações focadas no estudo das propriedades mecânicas de estruturas treliçadas apresentaram um crescimento expressivo nos últimos anos, totalizando 254.869 publicações ao final de 2021. Essa informação é apresentada no gráfico da Figura 1. O trabalho ainda aponta que as pesquisas mais atuais envolvendo estruturas treliçadas estão focadas basicamente em cinco frentes: Otimização da topologia, ajuste de parâmetros geométricos, inovação em materiais e técnicas de manufatura.

No relatório apresentado em 2016 pela empresa de consultoria Grand View Research (2016) há uma estimativa apontando que o mercado global de meta-materiais, avaliado em USD 316 milhões apresentaria (considerando apenas o mercado norte americano) uma taxa de crescimento anual próxima a 18% entre os anos de 2017 a 2025.

Uma das formas mais tradicionais de se resolver problemas de engenharia é por meio de técnicas de solução de equações diferenciais, método que consiste em encontrar uma solução analítica para um problema específico, dentro de suas hipóteses de validade, geralmente expressa por meio de equações diferenciais que descrevem o comportamento

Figura 1 - Evolução das publicações a respeito das propriedades mecânicas em estruturas treliçadas



Fonte: Miao et al. (2024)

do fenômeno estudado. No entanto esta abordagem apresenta limitações pois nem sempre é possível encontrar uma solução analítica para problemas complexos ou não lineares.

De forma alternativa, estes problemas, podem ser resolvidos a partir de métodos numéricos discretos, como é o caso do Método dos Elementos Finitos (MEF), no qual realiza-se a discretização do domínio, nos chamados de elementos finitos. Estes elementos se conectam por pontos comuns denominados de nós. Cada elemento é descrito por meio de um conjunto de funções de forma capaz de realizar a interpolação de pontos discretos para descrever todo o domínio de solução do problema. A solução aproximada do problema é obtida pela montagem de um sistema de equações algébricas que relacionam os graus de liberdade dos nós de todos os elementos, resultando numa matriz global que guarda as informações físicas do problema. A solução é então obtida por métodos numéricos apropriados. Atualmente o MEF é uma ferramenta relevante na simulação de problemas com geometria e condições de contorno complexas, auxiliando nas análises.

Proposto por Coda (2018), o Método dos Elementos Finitos Posicional (MEFP) difere do método tradicional, uma vez que aborda o problema de um ponto de vista energético em função das coordenadas nodais independentes. O MEFP traz simplicidade à problemas complexos, como por exemplo, a consideração da não linearidade geométrica, facilitando sua compreensão e modelagem, se comparado a técnicas tradicionais, como o elemento finito corrotacional.

1.1 Hipótese

É possível estabelecer um modelo não-linear geométrico e de arranjo periódico, capaz de representar o meio contínuo bidimensional utilizando o MEFP?

1.2 Objetivos Gerais

O objetivo principal deste estudo é desenvolver um modelo matemático/mecânico capaz de representar o comportamento do meio contínuo bidimensional utilizando uma rede de treliças discretas.

1.3 Objetivos Específicos

- Formular uma célula unitária como base da discretização do elemento finito de treliça posicional, não linear geométrico (NLG).
- Propor um modelo de homogeneização das células unitárias.
- Comparar e validar o modelo homogeneizado com a solução NLG do elemento de chapa

1.4 Justificativa

A motivação para esta proposta reside na simplicidade inerente aos elementos de treliça que possuem dois graus de liberdade por nó, reduzindo a complexidade do modelo mecânico em comparação com elementos bidimensionais. Além disso, a abordagem permite explorar analogias com materiais estruturados, como treliças periódicas ou metamateriais, oferecendo potencial para aplicações em design estrutural inovador e otimização.

1.5 Estrutura do trabalho

Essa dissertação está estruturada nos seguintes capítulos:

Capítulo 2: Apresenta-se uma breve revisão bibliográfica abordando os conceitos envolvidos necessários à compreensão da modelagem do problema bem como os conceitos relativos à mecânica do contínuo e não-linearidade geométrica.

Capítulo 3: Apresenta-se neste capítulo as bases do método dos elementos finitos posicional e conceitos fundamentais ao tema

Capítulo 4: Apresenta-se o desenvolvimento do modelo

Capítulo 5: Analisa-se os resultados da pesquisa demonstrando sua aplicabilidade.

Capítulo 6: Apresenta-se a conclusão e comentários a respeito dos resultados obtidos.

2 REVISÃO BIBLIOGRÁFICA

O desenvolvimento de materiais compostos visando aplicações estruturais tem sido intensamente explorado nas últimas décadas motivado por suas propriedades e desempenho superiores. Um dos principais desafios no estudo dessas estruturas é a complexidade associada à análise direta de sistemas com elevada heterogeneidade, uma característica comum entre materiais celulares e metamateriais. Yang et al. (2024), Zhang, O'Brien e Ghosh (2021)

Para Jagiello e Muñoz-Rojas (2021) um metamaterial é aquele cujo resposta mecânica global é determinada principalmente pela geometria e organização periódica de sua microestrutura, não dependendo apenas das propriedades do material base. Uma estrutura treliçada possibilita o desenvolvimento de materiais que frequentemente apresentam propriedades mecânicas superiores às dos materiais convencionais possuindo elevada razão rigidez/peso ou resposta anisotrópica adaptada à aplicação. Miao et al. (2024)

Uma forma de lidar com a complexidade inerente a tais estruturas é a adoção de métodos de homogenização, uma técnica onde um meio complexo é substituído por um meio equivalente com propriedades efetivas representativas. A aplicabilidade das técnicas de homogenização passam pelas mais diversas áreas indo desde o dimensionamento eficiente de componentes estruturais leves e estruturas espaciais até o desenvolvimento de metamateriais e biomateriais com propriedades mecânicas projetadas. Mohammadi et al. (2024)

Através dos processos de homogenização, é possível modelar estruturas complexas onde, com outras técnicas, sua representação em uma escala global seria inviável. Ao traduzir a resposta de uma célula unitária, que representa a microestrutura, para um meio contínuo equivalente, torna-se possível realizar análises estruturais em componentes reais com elevada acurácia, sem incorrer necessariamente em um aumento do custo computacional Vigliotti e Pasini (2012)

Nesse sentido, Liu e Lv (2017) apresentaram uma formulação multiescala para análise geométrica não linear de estruturas lattice-truss bidimensionais, na qual a célula unitária é convertida em um elemento contínuo equivalente. Essa abordagem, baseada na combinação do Método dos Elementos Finitos Multiescala e do método co-rotacional, permite capturar de forma eficiente as respostas locais de deslocamento, tensão e deformação a partir de simulações no nível macroscópico, reduzindo custos computacionais sem perder precisão.

Um exemplo disso é a chamada homogenização assintótica onde se realiza uma separação de escalas entre a macroestrutura e a microestrutura para derivar as propriedades efetivas do material a partir de uma célula unitária. Kalamkarov, Andrianov e Danishevs'kyi (2009). Para Noor (1988), os processos de homogeneização representam

ferramentas fundamentais na descrição do comportamento macroscópico de materiais e estruturas com periódicas como é o caso de estruturas formadas a partir de redes treliçadas.

Reis e Ganghoffer (2014) exploraram uma metodologia baseada em expansão discreta assintótica para determinar a resposta elastoplástica homogeneizada de treliças 2D repetitivas. O método permite modelar a superfície de escoamento inicial e sua evolução com o encruamento, sem impor a hipótese de deformação uniforme na célula unitária. Esse procedimento mostrou-se eficaz para prever o comportamento sob carregamentos multiaxiais, oferecendo um ferramental robusto para o dimensionamento de novas topologias bidimensionais.

Uma outra técnica de homogenização é a homogeneização por energia de deformação cuja base reside na equivalência da energia de deformação entre o comportamento mecânico de uma estrutura periódica e um meio contínuo equivalente. Glaesener et al. (2020), abordagem esta fundamentada fundamentada no chamado postulado de Hill-MandelMohammadi et al. (2024) através do qual é possível obter as propriedades efetivas do meio contínuo, como por exemplo, rigidez, anisotropia e resistência, partindo da resposta mecânica da microestruturaGasparetto e ElSayed (2021). A homogeneização baseada em energia foi utilizada aplicada em estruturas treliçadas por Xu e Zhang (2024) onde determinou o tensor equivalente e coeficientes de expansão térmica baseando-se apenas na energia de deformação de uma célula unitária.

Zschoernack, Wade e Völlmecke (2016) também analisaram células unitárias 2D sob o ponto de vista da estabilidade, investigando a flambagem não linear de geometrias quadradas reforçadas com fibras. Por meio de métodos energéticos e análises de perturbação, os autores demonstraram como o reforço pode aumentar significativamente a carga crítica de flambagem, estendendo o regime de utilização segura dessas estruturas.

A utilização de tais técnicas são especialmente atrativas pois seu desempenho em uma escala macroscópica pode ser diretamente relacionado à topologia da malha, determinando assim, as propriedades efetivas mesmo em situações de deformações não lineares e instabilidades geométricas. Phlipot e Kochmann (2019)

Ráž, Chval e Pereira (2024) destacam ainda a relevância de considerar desde células bidimensionais simples, como colmeias, até geometrias tridimensionais mais complexas na manufatura aditiva. Em seu estudo, demonstraram que a simulação numérica de estruturas 2D impressas por Multi Jet Fusion apresentou erro inferior a 8% em relação aos ensaios experimentais, validando a aplicabilidade do modelo para previsão de desempenho mecânico.

Aliado aos processos de homogenização, o método dos elementos finitos apresenta-se como uma das principais soluções numérica para a obtenção das propriedades efetivas em simulações com elementos representativos, viabilizando projetos de materiais e componentes complexos Zhang et al. (2024).

A análise parte de um elemento representativo de volume (ERV), uma pequena porção de material com formato e dimensões tais que ela seja capaz de representar o comportamento médio do meio. Kalamkarov, Andrianov e Danishevs'kyi (2009). Para Mohammadi et al. (2024) o RVE em estruturas treliçadas periódicas define-se como sendo a menor unidade estrutural capaz de preservar as características geométricas e mecânicas fundamentais do material capazes de representar as propriedades mecânicas (módulo de elasticidade, módulo de cisalhamento e coeficiente de Poisson) do material.

Guan et al. (2023) propuseram um método de homogeneização para análise modal e de amortecimento em estruturas sanduíche com núcleo lattice-truss piramidal, baseado na aplicação de condições periódicas de contorno em um elemento de volume representativo (RVE). O método, utilizando análise dinâmica no domínio da frequência, mostrou-se eficiente para determinar frequências naturais, modos de vibração e fatores de perda, apresentando boa concordância com modelos sólidos detalhados e resultados experimentais.

No contexto de otimização estrutural, Georges, Mittelstedt e Becker (2023) desenvolveram um modelo analítico que permite graduar o diâmetro das hastes dentro de um núcleo lattice-truss utilizando RVE, de modo a ajustar a rigidez local em função da distribuição de tensões. Esse tipo de gradação mostrou-se particularmente útil para lidar com cargas concentradas e evitar sobredimensionamentos, resultando em projetos mais leves e eficientes.

Park, Venter e Plessis (2023) demonstraram que a combinação de dados obtidos por microtomografia computadorizada com modelos simplificados de RVE é uma estratégia poderosa para capturar defeitos reais de fabricação em estruturas lattice. Essa abordagem, aplicada a simulações de compressão, permitiu prever o comportamento mecânico com erros inferiores a 17% na rigidez e 7% na tensão máxima, ao mesmo tempo que reduziu significativamente o custo computacional.

Haberland et al. (2024) avançaram na fundamentação matemática do uso de RVE em redes de molas elasto-plásticas, provando a convergência da aproximação periódica à medida que o tamanho do RVE aumenta. Embora o estudo tenha caráter teórico, os autores aplicaram o método a modelos bidimensionais de treliça, investigando respostas sob carregamento cíclico e monotônico, com potencial aplicação em análises de instabilidade e comportamento não linear de estruturas lattice.

Embora este estudo utilize conceitos inspirados na homogeneização, ele não se enquadra no sentido clássico do termo, que normalmente parte de uma microestrutura real para deduzir propriedades efetivas de um meio contínuo equivalente.

Neste trabalho, o processo é inverso: parte-se de um meio contínuo bidimensional em estado plano de tensão e constrói-se uma representação discreta periódica, composta por elementos de treliça, cujo comportamento mecânico é calibrado para reproduzir, de forma equivalente, a resposta do contínuo. Trata-se, portanto, de uma abordagem de modelagem numérica baseada em equivalência energética, na qual a geometria e o arranjo

das barras determinam as propriedades globais do modelo. Essa estratégia aproxima-se das técnicas aplicadas no estudo de metamateriais e materiais estruturados, nos quais a resposta global é definida pela topologia interna, mas difere da homogeneização tradicional por não ter como ponto de partida uma microestrutura física pré-existente.

Neste contexto, o Método dos Elementos Finitos Posicional (MEFP) como uma alternativa à formulação convencional baseada em deslocamentos. O MEFP considera diretamente as posições atuais dos nós como variáveis cinemáticas, permitindo assim uma formulação geometricamente exata em grandes deformações. Teles, Amorim e Leonel (2023)

3 FORMULAÇÃO DO ELEMENTO FINITO POSICIONAL

3.1 Apresentação do Capítulo

Neste capítulo serão apresentados alguns conceitos adotados no desenvolvimento do trabalho como: matriz Hessiana, deformação de Green-Lagrange e modelo constitutivo de Saint-Venant-Kirchhoff para os elementos finitos de treliça e de chapa. Esses conceitos são apresentados trazendo sua aplicabilidade ao problema do elemento finito proposto.

3.2 O Método dos Elementos Finitos Posicional (MEFP)

Amplamente utilizado em problemas complexos que não possuem solução analítica, o método dos elementos finitos opera através da discretização do domínio analisado em elementos menores de geometria conhecida cujos vértices formam os nós deste elemento e toda a formulação se baseia nos deslocamentos nodais.

Como uma alternativa a esta abordagem, o método dos elementos finitos posicional que, ao contrário do método tradicional, utiliza como parâmetro a posição nodal. É possível que em um primeiro momento esta abordagem não represente uma grande alteração no procedimento pois, em essência, o deslocamento é função da posição. Na verdade, as implicações são substanciais.

O uso de coordenadas nodais (posição) na modelagem apresenta vantagens especialmente na análise de problemas de não linearidade geométrica onde são esperados grandes deslocamentos e rotações.

3.3 Energia Potencial Total e o Princípio da Estacionariedade

Para um sistema perfeitamente elástico a energia potencial total Π é obtida a partir da soma das energias potenciais associadas, sendo elas: Energia potencial das forças externas P , energia de deformação U , expresso pela equação 3.1.

$$\Pi = P + U \quad (3.1)$$

O princípio da estacionariedade da energia potencial afirma que uma estrutura em equilíbrio tem a menor energia potencial possível (a energia potencial é mínima), (ZHANG et al., 2020). No caso do elemento finito posicional, a energia potencial é estacionária em relação as posições dos nós do elemento. Isso significa que mudar as posições dos nós da estrutura quer seja ele através do deslocamento, rotação ou deformação, resultará na variação de sua energia potencial sendo a condição de equilíbrio alcançada em um mínimo

do funcional da energia potencial total, expresso pela equação 3.2.

$$\delta\Pi = \delta P + \delta U = 0 \quad (3.2)$$

Em termos práticos, a condição de equilíbrio pode ser determinada aplicando os testes das derivadas primeira e derivada segunda. Tais testes tem por objetivo determinar se, naquele ponto, uma função é um encontra-se em um máximo, um mínimo ou uma inflexão, determinando assim o estado de equilíbrio.

A primeira derivada da função energia potencial indicará se aquele ponto é um máximo ou mínimo local, condição esta que é verificada aplicando o teste da derivada segunda.

Uma vez que a energia é uma grandeza escalar e esta será derivada em função de um vetor, seu resultado não será mais um escalar e sim um vetor, no caso, um vetor gradiente. O princípio da estacionariedade é dado pela equação 3.3

$$\frac{\partial\Pi}{\partial Y_i} = \nabla\Pi = 0 \quad (3.3)$$

Para o teste da derivada segunda, um valor positivo indicará um ponto de mínimo enquanto um valor negativo indicará um valor de máximo, representando aqui equilíbrio estável e instável, respectivamente. Uma derivada segunda igual a zero indica equilíbrio indiferente. Uma vez que o objetivo é determinar a a menor energia potencial, calcula-se a derivada segunda da função energia potencial, resultando na chamada Matriz Hessiana, dada pela equação 3.4.

$$H_{ij}^{\alpha\beta} = \frac{\partial^2\Pi}{\partial Y_j^\beta \partial Y_i^\alpha} \quad (3.4)$$

Segundo Coda e Greco (2004), a matriz Hessiana é uma matriz quadrada, simétrica, que contém as segundas derivadas parciais da energia potencial . A matriz Hessiana pode ser usada para caracterizar o comportamento local da função a partir da análise de sua convexidade. Essa análise permite a determinação dos pontos críticos e os valores extremos. Sendo assim, o ponto de menor energia ocorrerá quando a matriz Hessiana for positiva definida .

O significado físico da matriz Hessiana ser positiva definida está relacionado com a curvatura da superfície formada pela função energia potencial. A curvatura mede o quanto a superfície se afasta de um plano tangente em cada ponto. Quanto maior a curvatura mais rápida é a variação da função. A matriz Hessiana pode ser vista como uma medida da taxa de variação do gradiente da função que, por sua vez, indica a direção de maior crescimento da função.

3.4 Não-Linearidade Geométrica

A não linearidade geométrica trata de estruturas submetidas à grandes deslocamentos e/ou rotações, resultando assim, em uma reproposta não linear com relação a estes parâmetros. Nestes casos, muito embora o deslocamento global seja grande, a deformação em um ponto qualquer é considerada pequena quando considerado um modelo linear elástico, neste trabalho Saint-Venant Kirchhoff. (GRECO, 2004) e (SOARES; PACCOLA; CODA, 2019).

Em tais problemas a análise é executada considerando a geometria deformada. Posto isso, é necessário, portanto, definir o referencial a ser adotado para descrever o movimento do sólido. Neste trabalho adota-se o referencial Lagrangiano total onde o referencial está sobre o sólido considerado (treliça), na sua configuração inicial. Há também outras formas de representar esse referencial, como na descrição Lagrangiana atualizada ou ainda Euleriana, porém, esses referenciais não serão tratados neste trabalho.

3.5 MÉTODO DOS ELEMENTOS FINITOS POSICIONAL APLICADO AO PROBLEMA DE TRELIÇAS PLANAS

O método dos elementos finitos posicional (MEFP) apresentado por Coda (2018) trata da solução de problemas utilizando o método dos elementos finitos a partir de uma abordagem energética considerando as posições nodais como graus de liberdade. Em essência o problema é formulado através de um funcional de energia potencial que tem como ponto de partida a posição inicial dos nós. Aplicando o princípio da estacionariedade é possível então resolver o problema através de métodos iterativos.

Uma das principais vantagens do MEFP é que ele permite tratar de forma natural e unificada os problemas que apresentam não linearidade geométrica. Isso se deve ao fato de que o MEFP utiliza as coordenadas nodais como variáveis independentes, e não os deslocamentos, como nos métodos convencionais, permitindo assim ao MEFP lidar com grandes rotações e deslocamentos sem perder a consistência e precisão, mantendo a objetividade das medidas de deformação, Liu et al. (2012)

A energia potencial devido a ação de forças externas em um sistema conservativo será dada pelo somatório do trabalho realizado por estas forças com relação a posição atual, é dada pela equação 3.5,

$$P = -\vec{F}_{\text{ext}} \cdot \vec{Y} \quad (3.5)$$

A energia potencial de deformação U_e , dada pela equação 3.6, em um referencial Lagrangiano, é calculada a partir de seu volume inicial V_0 :

$$U_e = \int_{V_0} u_e \cdot dV \quad (3.6)$$

sendo u_e a energia específica de deformação, dada pela equação 3.7

$$u_e = \frac{1}{2} E \cdot \mathbb{E}^2 \quad (3.7)$$

- E : Módulo de elasticidade longitudinal
- \mathbb{E} : Deformação de Green-Lagrange

A deformação de Green-Lagrange, ou simplesmente deformação de Green, leva em consideração os componentes de deformação lineares e não-lineares, o que a torna adequada para a análise de problemas que envolvem a não-linearidade geométrica, (KOJIĆ; MILOŠEVIĆ; ZIEMYS, 2023)

A deformação de Green pode ser escrita em termos dos comprimentos inicial (l_0) e final (l) de um elemento que sofre um deslocamento passando de uma configuração inicial X à uma configuração atual Y . Escrevendo a deformação de Green em termos dos comprimentos das barras de treliça, obtém-se:

$$\mathbb{E} = \frac{1}{2} \frac{l^2 - l_0^2}{l_0^2} \quad (3.8)$$

Sendo os comprimentos inicial (l_0) e final (l) reescritos em termos das coordenadas nodais (inicial e final), respectivamente representadas por X_i^j e Y_i^j , com i representando a direção e j o número do nó

$$l_0 = \sqrt{(X_1^2 - X_1^1)^2 + (X_2^2 - X_2^1)^2} \quad (3.9a)$$

$$l = \sqrt{(Y_1^2 - Y_1^1)^2 + (Y_2^2 - Y_2^1)^2} \quad (3.9b)$$

A energia potencial total do sistema será dada pela soma das parcelas de energia de deformação e trabalho realizado pelo carregamento pontual, conforme a equação 3.10

$$\Pi = \int_{V_0} \left(\frac{1}{2} E \cdot \mathbb{E}^2 \right) \cdot dV - F_i^\alpha Y_i^\alpha = \frac{1}{2} E A_0 l_0 \mathbb{E}^2 - F_i^\alpha Y_i^\alpha \quad (3.10)$$

Assim como o índice j , o índice α é um índice mudo e pode ser substituído por qualquer representação. Esse valor representa o nó do elemento.

3.5.1 Aplicação do princípio da estacionariedade

De acordo com o princípio da estacionariedade o equilíbrio ocorre em um ponto de mínimo local da energia potencial. Deve-se, portanto, derivar a energia total em função da posição atual.

$$\frac{\partial \Pi}{\partial Y_i^\alpha} = \frac{\partial U_e}{\partial Y_i^\alpha} + \frac{P}{\partial Y_i^\alpha} = 0 \quad (3.11)$$

Substituindo as equações 3.5 e 3.6 em 3.11,

$$\frac{\partial \Pi}{\partial \vec{Y}} = \frac{\partial}{\partial \vec{Y}} \left(\frac{1}{2} EA_0 l_0 \mathbb{E}^2 \right) + \frac{\partial}{\partial \vec{Y}} (-F_i^\alpha Y_i^\alpha) = 0 \quad (3.12)$$

e realizando as devidas manipulações, a equação 3.10 pode ser reescrita como 3.13

$$\frac{EA_0 \mathbb{E}}{l_0} (-1)^\beta (Y_j^2 - Y_j^1) - F_j^\beta = 0 \quad (3.13)$$

3.5.2 Aplicação do Método de Newton-Rahpson

O método de Newton-Rahpson é uma metodologia de previsão-correção de cálculo numérico, iterativo, utilizado para o cálculo das raízes de uma equação não linear. O método consiste em um processo de linearização que parte de um valor inicial arbitrário (previsão) onde é avaliada sua derivada. A intersecção desta reta com o eixo das abcissas fornece um novo valor inicial para que o processo recomece (correção), sendo repetido até que um critério de convergência seja alcançado. Considera-se que o equilíbrio é alcançado quando a função resíduo, dada pela equação 3.14, for zero ou o mais próximo possível deste valor, (KZAM, 2016)

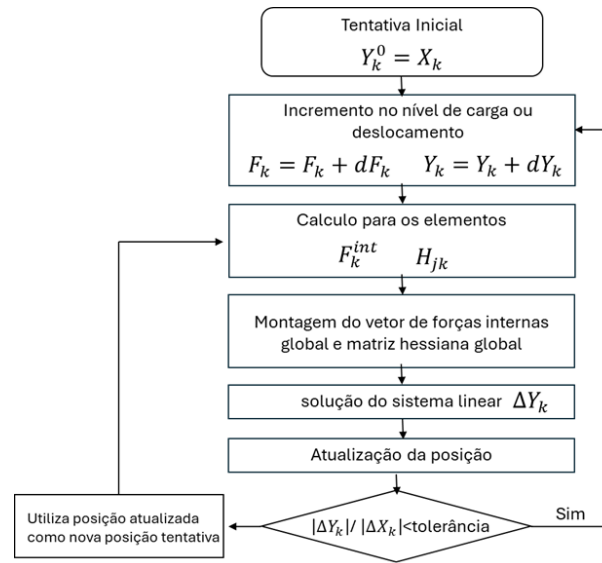
$$G_j^\beta(Y) = \frac{\partial \Pi}{\partial Y_j^\beta} = \frac{\partial U_e}{\partial Y_j^\beta} - F_j^\beta = F_j^{\beta int} - F_j^\beta = 0 \quad (3.14)$$

A fim de encontrar o termo de correção, a função resíduo é expandida em Série de Taylor até o termo de primeira ordem, dada pela equação 3.15

$$\Delta Y_i = - \left(\frac{\partial^2 U_e}{\partial Y_i^\alpha \partial Y_j^\beta} \right)_{Y_0}^{-1} G_j^\beta(Y_0) = - \left(H_{ij}^{\alpha\beta} \right)^{-1} G_j^\beta(Y_0) \quad (3.15)$$

O método é visto de forma sintetizada no fluxograma mostrado na figura 2

Figura 2 - Fluxograma para o método de Newton-Rahpson



Fonte: Adaptado de Coda (2018)

3.6 MÉTODO DOS ELEMENTOS FINITOS DE CHAPA PLANA

No MEFP, o domínio contínuo da chapa é discretizado em uma malha de elementos finitos. Para análise de chapas, elementos 2D são tipicamente usados representando a superfície média da placa. A geometria e os deslocamentos no plano são aproximados usando funções de forma (ou interpoladoras).

3.7 Relação tensão-deformação e estado plano de tensões

O estado plano de tensões caracteriza-se pela presença de tensões normais e cisalhantes em em direções contidas em um único plano enquanto as demais tensões que se manifestam na direção perpendicular a este plano são nulas ou desprezíveis, sendo validas, portando, para chapas com espessura muito menor que as demais dimensões, (SADD, 2009)

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{21} & \sigma_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \epsilon = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & 0 \\ \epsilon_{21} & \epsilon_{22} & 0 \\ 0 & 0 & \epsilon_{33} \end{bmatrix} \quad (3.16)$$

Os tensores de tensão e deformação se relacionam através de um mapeamento linear

$$\sigma = D\epsilon \quad (3.17)$$

onde D é a matriz de rigidez.

$$D = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1 - \nu) \end{bmatrix} \quad (3.18)$$

Para o estado plano de tensões, tem-se, portanto,

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1 - \nu) \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ 2\epsilon_{12} \end{Bmatrix} \quad (3.19)$$

3.7.1 Cinemática do elemento finito de chapa quadrilateral (Mapeamento)

A cinemática de um elemento quadrilateral tem por base a definição das funções de forma responsáveis por criar o mapeamento entre os domínios adimensional e físico do elemento, criando assim um paralelo entre a geometria deformada e a indeformada. Babu e Patel (2020)

O elemento finito de chapa quadrilateral, é um elemento bidimensional formado a partir de quatro nós, com dois graus de liberdade cada, dados por X_1 e X_2 , orientados nos eixos cartesianos, respectivamente. A geometria do elemento de chapa, tanto em sua configuração inicial (\mathbf{X}) quanto na atual (\mathbf{Y}), mostrados na figura 3, é aproximada usando coordenadas nodais e funções de forma. Para um elemento de chapa 2D, a superfície média pode ser mapeada a partir de um domínio adimensional (ξ_1, ξ_2) para o domínio físico, Coda (2018).

As coordenadas iniciais e atuais de qualquer ponto dentro de um elemento são expressas pelas equações 3.20 e 3.21, respectivamente

$$x_i(\xi_1, \xi_2) = \sum_{\lambda=1}^N \phi_\lambda(\xi_1, \xi_2) X_{i\lambda} \quad (3.20)$$

$$y_i(\xi_1, \xi_2) = \sum_{\lambda=1}^N \phi_\lambda(\xi_1, \xi_2) Y_{i\lambda} \quad (3.21)$$

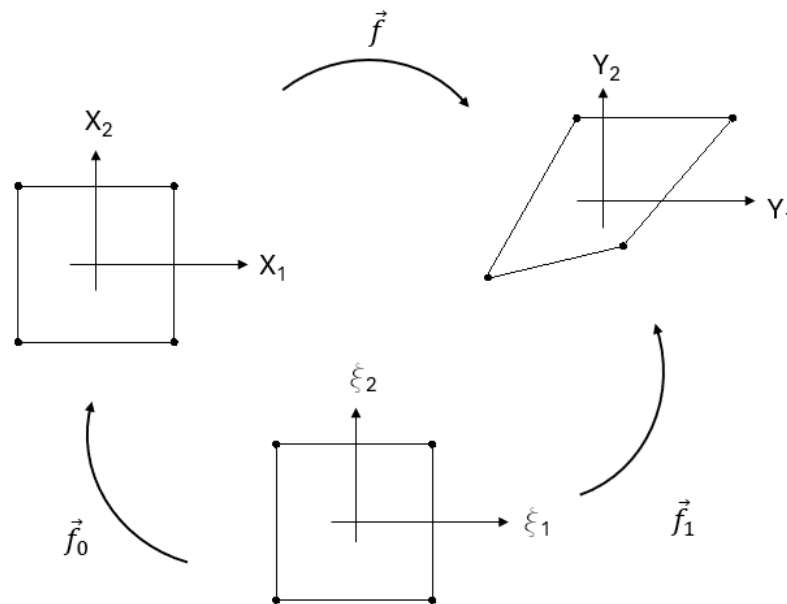
onde:

- $x_i(\xi_1, \xi_2)$ e $y_i(\xi_1, \xi_2)$ são os vetores de posição de um ponto nas configurações inicial e atual, respectivamente.
- $\phi_\lambda(\xi_1, \xi_2)$ são as funções de forma 2D (por exemplo, polinômios de Lagrange) para o nó λ .

- $X_{i\lambda}$ e $Y_{i\lambda}$ são as coordenadas globais inicial e atual do nó λ , respectivamente.
- N é o número de nós no elemento.
- i representa as componentes do vetor posição
- λ é o número de nós dependente do grau do polinômio aproximador da função de forma

Para esta explicação, considera-se um elemento 2D geral. A escolha das funções de forma dita a ordem de aproximação (por exemplo, linear, quadrática, cúbica). Para aproximações de ordem superior, mais nós internos são usados para capturar padrões de deformação complexos.

Figura 3 - Mapeamento cinemático do elemento finito de chapa quadrilateral



3.7.2 Tensor Gradiente da Função Mudança de Configuração

O tensor gradiente da função mudança de configuração $\mathbf{A} = \nabla f$ pode ser expresso como:

$$\mathbf{A} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \xi} \left(\frac{\partial \mathbf{x}}{\partial \xi} \right)^{-1} = \mathbf{A}_1 \mathbf{A}_0^{-1} \quad (3.22)$$

onde:

- $\frac{\partial \mathbf{Y}}{\partial \xi}$ é a matriz Jacobiana do mapeamento do domínio adimensional para a configuração atual, denotada como \mathbf{A}_1 .

- $\frac{\partial \mathbf{X}}{\partial \xi}$ é a matriz Jacobiana do mapeamento do domínio adimensional para a configuração inicial, denotada como \mathbf{A}_0 .

Os componentes de \mathbf{A}_0 e \mathbf{A}_1 são calculados a partir das derivadas parciais das funções de forma e das coordenadas nodais:

$$\mathbf{A}_0 = \frac{\partial x_i}{\partial \xi_j} = \sum_{\lambda=1}^N \frac{\partial \phi_\lambda}{\partial \xi_j} X_\lambda^i \quad (3.23)$$

$$\mathbf{A}_1 = \frac{\partial y_i}{\partial \xi_j} = \sum_{\lambda=1}^N \frac{\partial \phi_\lambda}{\partial \xi_j} Y_\lambda^i \quad (3.24)$$

Uma vez que \mathbf{A}_0 e \mathbf{A}_1 são determinados em cada ponto de integração dentro do elemento, o gradiente de deformação \mathbf{A} pode ser calculado. Essa abordagem permite que o gradiente de deformação, e conseqüentemente o tensor de deformações de Green-Lagrange, sejam expressos diretamente como funções das posições nodais atuais \mathbf{y}_λ .

3.8 Modelo Constitutivo para Elementos de Chapa (Modelo de Saint-Venant-Kirchhoff)

Para placas elásticas, uma escolha comum para o modelo constitutivo é o modelo de material de Saint-Venant-Kirchhoff. Este modelo assume uma relação linear entre a tensão de Piola-Kirchhoff de segunda espécie S , dado pela equação 3.25), e a deformação de Green-Lagrange, estendendo efetivamente a elasticidade linear para grandes deslocamentos.

$$S_{ij} = \mathbb{C}_{ijkl} E_{kl} \quad (3.25)$$

onde \mathbb{C}_{ijkl} é o tensor constitutivo de quarta ordem. Para materiais isotrópicos, \mathbb{C}_{ijkl} pode ser expresso em termos das constantes de Lamé (λ, μ) ou do módulo de Young e do coeficiente de Poisson.

Para elementos de Chapa, que tipicamente envolvem uma suposição de estado plano de tensões ou estado plano de deformações, o tensor constitutivo precisa ser adaptado.

3.8.1 Considerações de Estado Plano de Tensões / Estado Plano de Deformações

Para placas finas, a suposição de estado plano de tensões é frequentemente empregada, o que significa que as tensões normais à superfície média da placa são consideradas desprezíveis ($\sigma_{33} = \sigma_{13} = \sigma_{23} = 0$). Para placas espessas ou aplicações específicas, uma suposição de estado plano de deformações pode ser mais apropriada ($\epsilon_{33} = \epsilon_{13} = \epsilon_{23} = 0$).

Para o modelo Saint-Venant Kirchhoff (SVK) em estado plano de tensões, os componentes relevantes de S e E são aqueles no plano da placa (por exemplo, S_{11}, S_{22}, S_{12} e

E_{11}, E_{22}, E_{12}). A relação constitutiva então se torna:

$$\begin{bmatrix} S_{11} \\ S_{22} \\ S_{12} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{bmatrix} \quad (3.26)$$

Essa relação define a densidade de energia de deformação u_e para o elemento de placa.

3.9 Derivação das Forças Internas

As forças internas ($F_{\lambda i}^{\text{int}}$) em cada nó λ em cada direção i são obtidas derivando a energia de deformação total do elemento em relação à coordenada nodal correspondente. A energia de deformação total para um elemento é a integral da densidade de energia de deformação sobre seu volume inicial:

$$U_e = \int_{V_0} u_e dV_0 \quad (3.27)$$

Usando a regra da cadeia e a relação $u_e = S_{kl}E_{kl}$

$$F_{\lambda i}^{\text{int}} = \frac{\partial U_e}{\partial Y_{\lambda}^i} = \int_{V_0} S_{kl} \frac{\partial E_{kl}}{\partial Y_{\lambda}^i} dV_0 \quad (3.28)$$

O termo $\frac{\partial E}{\partial Y_{\lambda}^i}$ precisa ser derivado. Lembrando que $E = \frac{1}{2}(\mathbf{A}^T \mathbf{A} - \mathbf{I})$, então

$$\frac{\partial \mathbf{E}}{\partial Y_{\lambda}^i} = \frac{1}{2} \left[\mathbf{A}_0^{-T} \frac{\partial \mathbf{A}_1}{\partial Y_{\lambda}^i} \mathbf{A} + \mathbf{A} \mathbf{A}_0^{-T} \frac{\partial \mathbf{A}_1}{\partial Y_{\lambda}^i} \right] \quad (3.29)$$

Realizando as contrações de tensor necessárias obtém-se a expressão para as forças internas.

A integral sobre V_0 é transformada em uma integral sobre o domínio inicial adimensional usando o Jacobiano do mapeamento inicial:

$$F_{\lambda i}^{\text{int}} = \int_{\Omega_0} S_{kl} \frac{\partial E_{kl}}{\partial Y_{\lambda}^i} J_0 d\Omega_0 \quad (3.30)$$

onde $J_0 = \det(\mathbf{A}_0)$ é o Jacobiano do mapeamento inicial e Ω_0 é o adimensional. Essa integral é calculada numericamente usando quadratura de Gauss onde N_{gp} é o número de pontos de Gauss e w_j são seus respectivos pesos:

$$F_{\lambda i}^{\text{int}} \approx \sum_{j=1}^{N_{gp}} \left(S_{kl} \frac{\partial E_{kl}}{\partial Y_{\lambda}^i} J_0 \right)_j w_j \quad (3.31)$$

3.10 Derivação da Matriz Hessiana

A matriz Hessiana, também chamada de rigidez tangente (H), é essencial para resolver as equações de equilíbrio não lineares usando um método iterativo como o de Newton-Raphson. Ela representa a linearização das forças internas em relação às posições nodais:

$$H_{\lambda i, \mu k} = \frac{\partial F_{\lambda, i}^{\text{int}}}{\partial Y_{\mu}^k} \quad (3.32)$$

De modo a simplificar a notação apresenta-se as equações na sua forma implícita, retornando na sequência à notação indiciais.

Essa derivada pode ser decomposta em duas partes: a matriz de rigidez do material (H^M) e a matriz de rigidez geométrica (ou de tensão inicial) (H^G).

$$H = H^M + H^G \quad (3.33)$$

A matriz de rigidez do material surge da mudança na tensão devido às mudanças na deformação:

$$H_{\lambda i \mu k}^M = \int_{V_0} \frac{\partial \mathbf{S}}{\partial \mathbf{E}} : \frac{\partial \mathbf{E}}{\partial Y_{\lambda}^i} : \frac{\partial \mathbf{E}}{\partial Y_{\mu}^k} dV_0 \quad (3.34)$$

E a matriz de rigidez geométrica surge da mudança na medida de deformação devido às mudanças nas posições nodais, com a tensão atual mantida constante:

$$H_{\lambda i \mu k}^G = \int_{V_0} \mathbf{S} : \frac{\partial^2 \mathbf{E}}{\partial Y_{\lambda}^i \partial Y_{\mu}^k} dV_0 \quad (3.35)$$

Ambas as integrais são avaliadas numericamente usando quadratura de Gauss, similarmente às forças internas. A segunda derivada do tensor de deformações de Green-Lagrange precisa ser cuidadosamente derivada, frequentemente envolvendo produtos de \mathbf{A}_1 , \mathbf{A}_0^{-1} , e derivadas de funções de forma.

4 REPRESENTAÇÃO DO MEIO CONTÍNUO UTILIZANDO ESTRUTURA TRELIÇADA

4.1 Apresentação do capítulo

Neste capítulo é apresentada a metodologia e as estruturas utilizadas nos exemplos de validação. Aqui é demonstrada a aplicação do processo buscando a equivalência mecânica à deformação de elementos em estado plano de tensão.

4.2 Representação do meio contínuo utilizando estruturas treliçadas

A utilização de estruturas treliçadas para representar meios contínuos busca aproximar o comportamento de materiais ou estruturas contínuas através de redes discretas de elementos interconectados. Estudos como o de Zheng et al. (2023) mostram que essas redes formadas por barras conectadas por nós de treliça permitem replicar propriedades mecânicas específicas, como rigidez, densidade e anisotropia. Por exemplo, a modelagem generativa tem sido aplicada para projetar metamateriais com propriedades otimizadas explorando vastos espaços de design por meio de algoritmos de aprendizado de máquina .

4.3 Homogeneização

Entende-se por homogeneização a técnica matemática empregada para derivar e determinar as propriedades macroscópicas efetivas oriundas de uma combinação de materiais e/ou geometrias presentes na microescala. O objetivo principal de empregar o processo de homogeneização é simplificar significativamente a análise e a compreensão de materiais que possuem estruturas intrincadas e complexas, permitindo assim a média de suas diversas propriedades em relação a um elemento de volume representativo (RVE) que se apresenta como um elemento fundamental do processo de homogeneização. (SOMNIC; JO, 2022)

A utilização de um modelo equivalente para representar o meio contínuo através de uma estrutura treliçada não é algo novo remontando aos estudos de Lagrange em meados do século XVIII (CHALLAMEL et al., 2023). Mais recentemente, trabalhos como os de Noor (1988) e Nayfeh e Hefzy (1978) abordaram esta problemática nas últimas décadas do século passado.

Em uma época onde o poder de processamento dos computadores era bastante limitado, o trabalho de Noor (1988) busca uma solução para o dimensionamento de estruturas treliçadas aplicadas em projetos desenvolvidos pela NASA. A solução considerada

foi a de buscar um meio contínuo equivalente de tal modo que uma solução analítica fosse possível. Graças ao aumento da capacidade computacional associada a técnicas mais eficientes como, por exemplo, o MEFP, é possível traçar o caminho inverso, analisando estruturas contínuas complexas através da utilização de um arranjo periódico treliçado.

A utilização de uma estrutura treliçada para representar de modo equivalente o meio contínuo não apenas simplifica sua representação mas também permite uma melhor compreensão do comportamento mecânico sob diversas condições de carregamento. Para uma maior precisão nos dados é de fundamental importância considerar a não-linearidade geométrica uma vez que permite que o modelo capture os efeitos associados à grandes deslocamentos. (LIU; LV, 2017)

Para Hassani e Hinton (1999) o processo de homogenização busca transformar as equações que representam um meio heterogêneo e, por consequência, possuem variáveis que oscilam rapidamente em equações constantes ou com variáveis que oscilam mais lentamente de modo que as respostas sejam consistentes com a equação inicial, obtendo assim um conjunto de propriedades efetivas.

No contexto deste trabalho, o processo de homogenização visa seguir o caminho inverso, buscando modelar estruturas planas bidimensionais submetida a um estado plano de tensão ou deformação através de uma rede de barras que estão dispostas em uma estrutura treliçada, sendo, portando, elementos unidimensionais uma vez que estão submetidos a solicitações puramente axiais.

Para este estudo serão adotadas três células regulares arrançadas em um arranjo periódico.

4.4 Materiais

4.4.1 Gerador de Malha

O gerador de malha foi desenvolvido de modo que, fornecida uma célula unitária qualquer, ele seja capaz de gerar estrutura treliçada dentro de um polígono desejado. Uma vez gerada a malha o processo de homogenização é automaticamente aplicado buscando a equivalência mecânica entre as propriedades de uma chapa com a geometria fornecida e as propriedades necessárias à treliça para que ela possua um comportamento mecânico equivalente. O código em linguagem Fortran é apresentado no Apêndice A.

4.4.2 Otimização do Código em Fortran Para Elementos Treliçados

O código inicialmente utilizado na análise das treliça pelo método dos elementos finitos posicional foi revisado. Foram implementadas as diretrizes OpenMP de modo a

otimizar o desempenho através do processamento paralelo. Esta medida resultou em um ganho de desempenho significativo pois conforme o número de elementos cresce, não faz sentido a adoção de um processamento sequencial de loops e funções.

Além da paralelização do código, outro elemento adotado em favor da eficiência de processamento foi a adoção do Método do Gradiente Conjugado devido à sua eficiência na solução de matrizes esparsas como as resultantes deste trabalho.

4.4.3 Modelagem no software ABAQUS

Por se tratar de uma análise em estado plano de tensões foi utilizado o elemento CPS8, um elemento bidimensional quadrilateral de 8 nós e formulação isoparamétrica quadrática, e nós com dois graus de liberdade de deslocamento cada. Este elemento é particularmente adequado para a modelagem de chapas, membranas e estruturas planas sujeitas a carregamentos no próprio plano. Por utilizar funções de interpolação de segunda ordem, o CPS8 consegue representar de forma mais precisa campos de deslocamentos e deformações quando comparado a elementos lineares Dassault Systèmes (2025).

Em todos os modelos adotou-se um módulo de elasticidade longitudinal $E=200$ GPa e coeficiente de Poisson $\nu=0,3$. Por se tratar de problemas não lineares nas configurações de *step* o procedimento adotado foi o chamado *Static, Riks* com o modo de não linearidade geométrica ativado *Nlgeom On*.

4.4.4 Células unitárias propostas

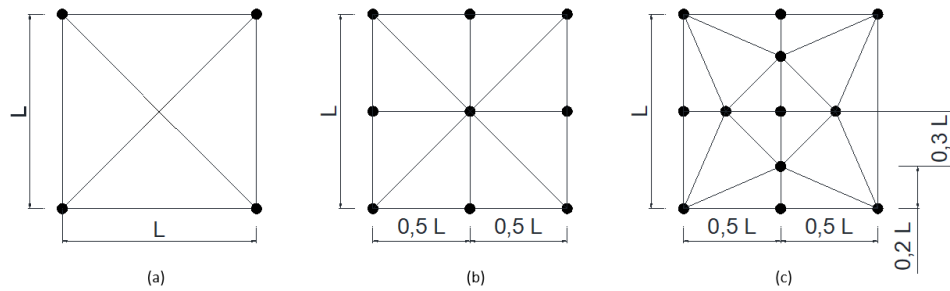
A proposta deste trabalho passa pela verificação de possíveis configurações de células unitárias, buscando a que melhor aproxima o comportamento estrutura treliçada ao de uma chapa plana equivalente. Considera-se como célula unitária um conjunto formado por nós e barras que se repetem de forma periódica ao longo da estrutura.

Como será apresentado mais adiante, o modelo adota a premissa de que o material modelado é isotrópico. Sendo assim, de modo a buscar tal comportamento, as três células adotadas são biximétricas. Neste trabalho foram consideradas três células unitárias distintas identificadas como Célula Unitária Contraventada (a) que é nomeada de forma simplificada como "X", E1 (b) e E2 (c), mostradas na figura 4.

4.5 Metodologia

O processo de representação do meio contínuo através de uma estrutura treliçada consiste em buscar a equivalência energética de modo que, quando sob efeito de um carregamento externo, a deformação da estrutura treliçada plana seja equivalente à do

Figura 4 - Células unitárias propostas



Fonte: Autor

meio contínuo equivalente em estado plano de tensão. Este processo é realizado nas seguintes etapas

1. Definição geométrica do problema,
2. Escolha da célula unitária,
3. Cálculo dos coeficientes de forma associados à célula unitária
4. Determinação da área das barras da treliça a partir da equivalência energética treliça/contínuo
5. Geração da malha
6. Simulação através do MEFP

4.5.1 Aplicação de carregamentos ou deslocamentos

A realização das simulações pode ser feita a partir de duas entradas distintas e mutuamente excludentes. Dado um número de passos (steps) para a aplicação da condição estabelecida, o carregamento ou o deslocamento total são obtidos através de um incremento deste parâmetro. Em ambos os casos é definida uma tolerância como critério de parada. A possibilidade de escolher o tipo de entrada é importante pois o código utiliza o método de Newton-Rahpson, não conseguindo, portanto, capturar os efeitos de *snap-back* e *snap-through* que, para Bathe (2006), são condições comuns em problemas não lineares onde uma estrutura ao atingir um ponto crítico sofre uma mudança abrupta para uma nova posição de equilíbrio.

4.6 Determinação do elemento treliçado equivalente

O dimensionamento das barras é executado sobre uma única célula unitária, buscando a área da seção transversal necessária às barras para que se obtenha uma energia de

deformação equivalente à do elemento contínuo de prova (chapa de mesmas dimensões). Para isso foram adotadas as seguintes hipóteses para a célula unitária

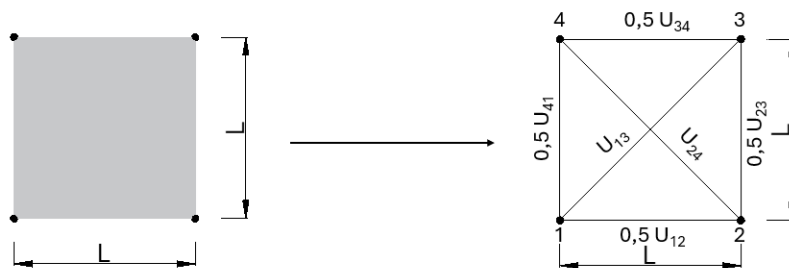
- Todas as barras são prismáticas de seção circular
- As barras possuem o mesmo módulo de rigidez da chapa,
- Apenas a deformação longitudinal dos elementos é considerada,
- Os nós são rotulas perfeitas, resultando em dois graus de liberdade por nó.
- As barras coincidentes com as arestas da chapa computam apenas metade da energia de deformação que, posteriormente, é corrigida por um fator de forma associado ao problema analisado.

Por outro lado, para a chapa foram consideradas as seguintes hipóteses

- A chapa encontra-se em estado plano de tensão,
- É modelada como um material isotrópico,

Dos itens citados acima, provavelmente o que merece maior atenção neste momento é o relativo à energia de deformação das barras coincidentes com as arestas da chapa de prova.

Figura 5 - Elementos compartilhados



Fonte: Autor

Como a análise é executada sobre uma única célula, figura 5, dois pontos devem ser considerados. Primeiro e mais óbvio é o efeito sobreposição que deve ser levado em consideração. Sendo assim, a análise é feita considerando apenas metade da energia de deformação nas barras que são compartilhadas. Além do compartilhamento de barras por células adjacentes, o efeito das demais barras da estrutura deve ser contabilizada, o que é feito adicionando o que será chamado de coeficiente de forma da estrutura.

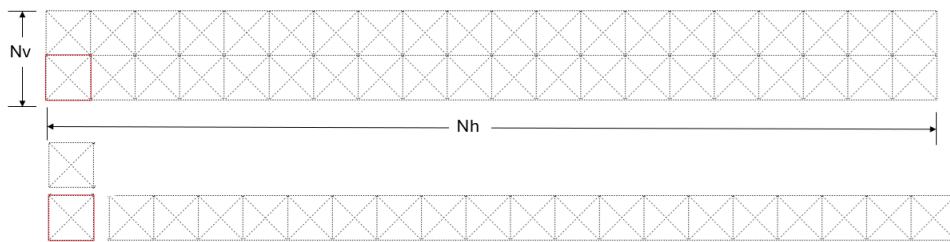
4.6.1 Propriedades do material

Todos os exemplos foram rodados considerando o aço como material constituinte com módulo de elasticidade $E = 200GPa$ e coeficiente de *poisson* $\nu = 0,3$

4.6.2 Coeficiente de forma da estrutura(f_k)

Este é provavelmente um dos pontos chaves na proposta apresentada. Como a análise é feita sobre uma única célula unitária, esta célula deve refletir, de alguma maneira, o comportamento das demais células que compõem a malha. Isto é feito adicionando um coeficiente para majorar a energia de deformação das barras periféricas. Esta análise é realizada tomando uma célula como referência e a separando do restante da malha, conforme mostrado na figura 6. Feito isto, são contabilizadas as células nas direções vertical (N_v) e horizontal (N_h) bem como suas projeções nestas direções.

Figura 6 - Decomposição da malha para o cálculo dos coeficientes



Fonte: Autor

Para a determinação dos coeficientes os seguintes passos devem ser executados:

1. Separar a célula da malha
2. Contabilizar as projeções das barras internas integralmente e as externas (livres) com valor 1/2. O mesmo é feito para linhas e colunas conectadas à célula.
3. Fazer a razão entre o numero de barras na célula e o número de barras na coluna ou fileira associada

O processo de equivalência energética assume que as barras compartilhadas em uma célula unitária devem ter seu valor computado pela metade. Nas células posicionadas nos limites da geometria analisada isso cria um desbalanceamento energético que é corrigido através da adoção de um fator de forma aplicado sobre a célula unitária.

O coeficiente de forma f , dado pela equação (4.1) depende, portando, da geometria da célula e da geometria do problema.

$$f = \frac{m}{(m-1)(N_{dir}-1)+1} \quad (4.1)$$

onde N_{dir} é o número de células em uma dada direção (N_v :direção vertical e N_h :direção horizontal) e m é um valor característico da célula unitária e deve ser avaliado em cada uma de suas direções e representa o somatório da projeção das barras em uma determinada direção, matematicamente representado pelas eq. 4.2 e 4.3

$$m_h = \frac{\sum \Delta x_{internas} + \frac{1}{2} \sum \Delta x_{externas}}{L_{célula}} \quad (4.2)$$

$$m_v = \frac{\sum \Delta y_{internas} + \frac{1}{2} \sum \Delta y_{externas}}{L_{célula}} \quad (4.3)$$

A tabela 1 apresenta os coeficientes m para as células utilizadas neste trabalho, observadas na fig. 4

Tabela 1 - Fatores m

Célula	m_h	m_v
X	3,0	3,0
E1	4,0	4,0
E2	6,0	6,0

Fonte: Autor

4.7 Equivalência energética treliça/contínuo

Uma vez que a célula unitária é definida e seus coeficientes calculados, o próximo passo é calcular a área necessária à seção transversal das barras da treliça para que sua energia de deformação ($U_{treliça}$) seja igual a energia de deformação de uma chapa de espessura unitária (U_{chapa}) e geometria equivalente.

4.7.1 Energia de deformação da chapa

A energia de deformação é dada pela equação 4.4:

$$U_{chapa} = \frac{1}{2} t L^2 \{\sigma\}^T \{\varepsilon\} \quad (4.4)$$

Sendo as tensões calculadas pela Lei de Hook, equação 4.5

$$\{\sigma\} = [D]\{\varepsilon\} \quad (4.5)$$

onde $[D]$ é a matriz de rigidez do material para o estado plano de tensão, dependente do Módulo de Young E e do Coeficiente de Poisson ν do material da chapa. Para facilitar a manipulação, convém definir $C = \frac{E}{(1+\nu)(1-2\nu)}$

$$\{\sigma\} = [D]\{\varepsilon\} = C \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ -\nu\varepsilon_x \\ \gamma_{xy} \end{Bmatrix} = C \begin{bmatrix} (1-\nu-\nu^2)\varepsilon_x \\ \nu(\nu-1)\varepsilon_x \\ \frac{1-2\nu}{2}\gamma_{xy} \end{bmatrix}$$

Operando o produto , $\{\sigma\}^T\{\varepsilon\}$

$$\{\sigma\}^T\{\varepsilon\} = [C(1-\nu-\nu^2)\varepsilon_x] \varepsilon_x + [C\nu(\nu-1)\varepsilon_x] (-\nu\varepsilon_x) + \left[C\frac{1-2\nu}{2}\gamma_{xy} \right] \gamma_{xy}$$

a agrupando os termos comuns,

$$= C \left[(1-\nu-2\nu^2+\nu^3)\varepsilon_x^2 + \frac{1-2\nu}{2}\gamma_{xy}^2 \right] \quad (4.6)$$

Feito isso, a energia , deformação da chapa 4.7 é obtida e substituindo 4.6 em 4.4

$$U_{\text{chapa}} = \frac{1}{2}tL^2C \left[(1-\nu-2\nu^2+\nu^3)\varepsilon_x^2 + \frac{1-2\nu}{2}\gamma_{xy}^2 \right] \quad (4.7)$$

4.7.2 Energia de deformação da treliça

A energia total de deformação da treliça que forma célula unitária será dada pela soma da energia necessária à deformação de cada uma de suas barras. Nas barras externas a energia deve ser computada pela metade acrescida de seu fator de forma uma vez que células adjacentes possuem barras compartilhadas.

A energia de deformação da k -ésima barra é dada pela equação 4.8

$$U_k = f_k \cdot \frac{1}{2} \cdot EAL_k\varepsilon_k^2 \quad (4.8)$$

Onde f_k é o coeficiente de forma, responsável pela correção energética. Deve ser aplicado a cada barra da seguinte maneira:

- $f_k = 1$ para barras internas.

- $f_k = (0.5 + f_{k,h})$ para barras horizontais nas arestas.
- $f_k = (0.5 + f_{k,v})$ para barras verticais nas arestas.

A energia de deformação da treliça será igual ao somatório da energia de deformação das barras, dada por 4.9:

$$U_{treliça} = \sum_{k=1}^{n_{barras}} f_k \cdot EAL_k \varepsilon_k^2 \quad (4.9)$$

A equação 4.9 pode ser reescrita em função dos nós e da deformação imposta inicialmente pelo problema.

Para a k-ésima barra formada pelos nós i e j , seu comprimento total é dado por 4.10:

$$L_k = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (4.10)$$

A deformação longitudinal ε_k , em cada uma das barras será, portanto:

$$\varepsilon_k = \frac{(u_{x,j} - u_{x,i}) \cos \theta_k + (u_{y,j} - u_{y,i}) \sin \theta_k}{L_k} \quad (4.11)$$

com os cossenos diretores dados por,

$$\cos \theta_k = \frac{x_j - x_i}{L_k}, \quad \sin \theta_k = \frac{y_j - y_i}{L_k}$$

e os deslocamentos dados por,

$$u_{x,j} - u_{x,i} = \varepsilon_x(x_j - x_i) + \frac{\gamma_{xy}}{2}(y_j - y_i)$$

$$u_{y,j} - u_{y,i} = -\nu\varepsilon_x(y_j - y_i) + \frac{\gamma_{xy}}{2}(x_j - x_i)$$

É possível determinar a deformação ao longo do eixo longitudinal ε_k das barras em função das deformações ε_x e γ_{xy} da chapa impostas como condição de contorno

$$\varepsilon_k = \frac{[\varepsilon_x(x_j - x_i) + \frac{\gamma_{xy}}{2}(y_j - y_i)] \cos \theta_k + [-\nu\varepsilon_x(y_j - y_i) + \frac{\gamma_{xy}}{2}(x_j - x_i)] \sin \theta_k}{L_k}$$

$$\varepsilon_k = \frac{\varepsilon_x(x_j - x_i) \cdot \frac{x_j - x_i}{L_k} + \frac{\gamma_{xy}}{2}(y_j - y_i) \cdot \frac{x_j - x_i}{L_k} - \nu\varepsilon_x(y_j - y_i) \cdot \frac{y_j - y_i}{L_k} + \frac{\gamma_{xy}}{2}(x_j - x_i) \cdot \frac{y_j - y_i}{L_k}}{L_k}$$

$$= \varepsilon_x \frac{(x_j - x_i)^2}{L_k^2} + \frac{\gamma_{xy} (x_j - x_i)(y_j - y_i)}{2 L_k^2} - \nu \varepsilon_x \frac{(y_j - y_i)^2}{L_k^2} + \frac{\gamma_{xy} (x_j - x_i)(y_j - y_i)}{2 L_k^2}$$

$$\varepsilon_k = \varepsilon_x (\cos^2 \theta_k - \nu \sin^2 \theta_k) + \gamma_{xy} \sin \theta_k \cos \theta_k \quad (4.12)$$

A energia total da treliça devido à deformação equivalente a da chapa será dada pela equação 4.9, reescrita em como 4.13

$$U_{\text{treliça}} = EA \sum_{k=1}^{n_{\text{bars}}} f_k L_k [\varepsilon_x (\cos^2 \theta_k - \nu \sin^2 \theta_k) + \gamma_{xy} \sin \theta_k \cos \theta_k]^2 \quad (4.13)$$

4.7.3 Aplicação da equivalência energética

A área necessária à seção transversal das barras que resulta em uma equivalência entre o comportamento mecânico do elemento de chapa e o de uma estrutura treliçada é obtida igualando as energias de deformação dadas por 4.7 e 4.13

$$U_{\text{chapa}} = U_{\text{treliça}} = A \cdot \dot{U}_{\text{treliça}} \quad (4.14)$$

Onde, por uma questão de simplificação na representação, $\dot{U}_{\text{treliça}}$ é a energia de deformação da treliça dividida pela área da seção transversal das barras.

$$\dot{U}_{\text{treliça}} = E \sum_{k=1}^{n_{\text{bars}}} f_k L_k [\varepsilon_x (\cos^2 \theta_k - \nu \sin^2 \theta_k) + \gamma_{xy} \sin \theta_k \cos \theta_k]^2 \quad (4.15)$$

A área será dada pela equação 4.16

$$A = \frac{U_{\text{chapa}}}{\dot{U}_{\text{treliça}}} \quad (4.16)$$

4.8 Fechamento

Neste capítulo foi possível estabelecer uma relação entre a energia de deformação de uma chapa em estado plano de tensão e um elemento treliçado. A partir desta relação obtêm-se a área necessária às barras da célula unitária que será então replicada formando uma malha capaz de representar o meio contínuo. Sua aplicabilidade será demonstrada no próximo capítulo através de exemplos

5 EXEMPLOS E ANÁLISE DOS RESULTADOS

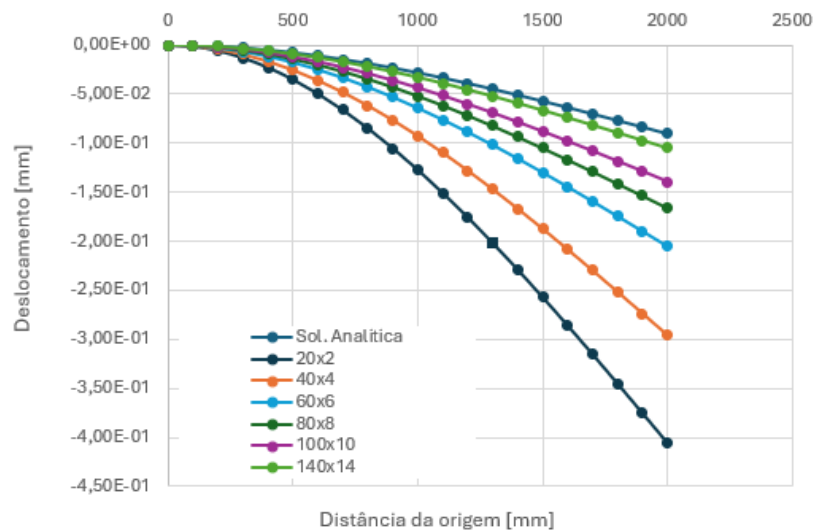
5.1 Apresentação do Capítulo

Este capítulo trata da apresentação dos exemplos propostos bem como dos resultados obtidos durante o desenvolvimento da pesquisa. Na sequência são apresentados exemplos com o intuito de validar o modelo proposto.

5.2 Resultados obtidos pelo método iterativo

Em um primeiro momento foram propostas vigas com configurações e carregamentos afim de testar a viabilidade de se adotar uma estrutura treliçada para representar o meio contínuo bidimensional.

Figura 7 - Convergência devido ao refinamento da malha em viga engastada



Fonte: Autor

5.2.1 Viga engastada com carga concentrada

O primeiro caso analisado foi o da viga engastada. Foram testadas seis malhas distintas, com o elemento proposto em dimensões variando 100x100 mm (20x2) a 14,28x14,28 mm (140x14). Este teste teve por objetivo verificar o comportamento da viga quanto a variação da rigidez dos elementos de barra.

Inicialmente, a abordagem foi sobre vigas com pequenas deformações. Uma vez fixada a área da seção transversal das barras o valor do módulo de elasticidade foi variado

até que uma condição de contorno qualquer fosse atendida, permitindo assim verificar o comportamento do restante da viga.

Este teste mostrou que deve haver, de alguma forma, uma relação entre área e rigidez para um dado elemento que produzirá um comportamento equivalente, ou muito próximo, ao comportamento esperado para o meio contínuo em condições de geometria e carregamento equivalentes.

5.3 Resultados obtidos com o processo de equivalência energética

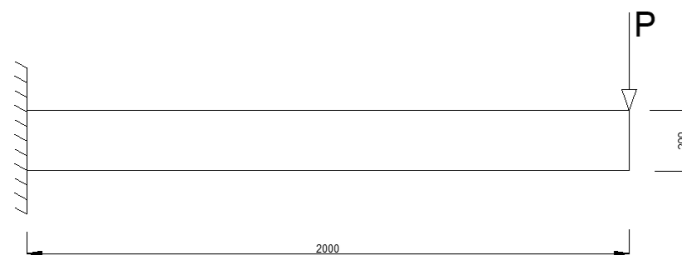
Partindo do pressuposto de que é possível representar o meio contínuo bidimensional através de uma estrutura treliçada a metodologia apresentada no capítulo 3 foi aplicada para as seguintes situações:

- Viga em balanço com carga na extremidade livre
- Pórtico de Lee
- Arco
- Coluna submetida à carregamento excêntrico

5.3.1 Exemplo 1 - Viga engastada com carga na extremidade livre

O primeiro exemplo é o caso de uma viga em balanço com uma carga concentrada na extremidade livre (Fig. 8).

Figura 8 - Viga em balanço com carga na extremidade



Fonte: Autor

As vigas foram simuladas considerando dois carregamentos distintos, sendo o primeiro $P=0,5$ KN e o segundo $P=100$ KN. Para cada uma das configurações de células unitárias propostas foram analisadas as malhas compostas por células unitárias de lado 100 mm, 50 mm, 25 mm, 10 mm e 5 mm. Como referencia foi adotado o elemento finito

posicional de chapa para elementos quadrados de segunda ordem com 10mm de lado, utilizando integração Gaussiana com iguais condições de contorno.

As tabelas 2 e 3 apresentam, para fins de comparação, informações como a quantidade de células utilizadas, total de elementos gerados e quantitativo de nós presentes em cada malha.

Tabela 2 - Dados das malhas 100 mm, 50 mm e 25 mm para a viga engastada

Célula	100 mm			50 mm			25 mm		
	40 Células			160 Células			640 Células		
	A(mm ²)	Nós	Elem.	A(mm ²)	Nós	Elem.	A(mm ²)	Nós	Elem.
X	32,49400	63	182	24,368	205	684	15,234	729	2648
E1	24,70103	205	524	16,82109	729	2008	9,68802	2737	7856
E2	19,06288	325	924	12,11899	1209	3608	6,66377	4657	14256

Fonte: Autor

Tabela 3 - Dados das malhas 10 mm e 5 mm para a viga engastada

Célula	10 mm			5 mm		
	4.000 Células			16.000 Células		
	A(mm ²)	Nós	Elem.	A(mm ²)	Nós	Elem.
X	7,04383	4221	16220	3,70480	16441	64440
E1	4,20598	16441	48440	2,16032	64881	192880
E2	2,80556	28441	88480	1,42573	112881	352880

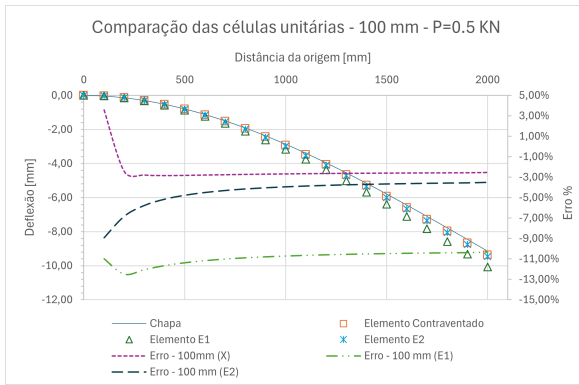
Fonte: Autor

A deflexão das diferentes malhas em comparação à viga de referência, todas sob um carregamento de 0,5 KN e 100KN são demonstradas nas figuras 9a à 9e e 10a à 10e.

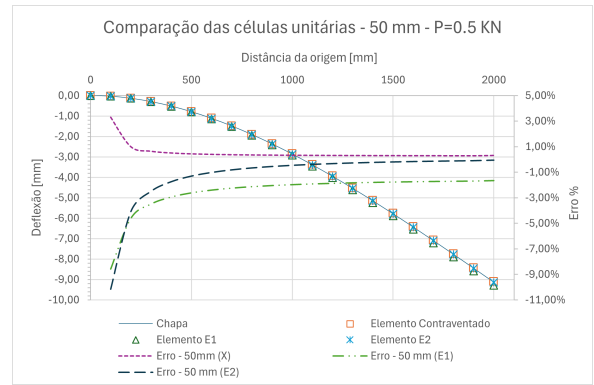
É possível identificar alguns padrões comuns a todas as células. Em termos de valores absolutos, o erro próximo ao engaste tende a ser superior aos demais pontos analisados. Em uma distância de aproximadamente 10%, muito semelhante ao efeito de Saint-Venant, do comprimento da viga, o erro tende a se tornar constante convergindo mais rapidamente para uma célula contraventada frente as demais células analisadas.

As figuras 9f à 9h e 10f à 10h comparam a influência do tamanho da célula, e por consequência a densidade da malha, sobre o erro com relação à viga de referência. Observa-se aqui que um aumento da densidade da malha não resultará necessariamente em melhor precisão, indicando, portanto, que deve haver um erro associado à geometria da célula unitária.

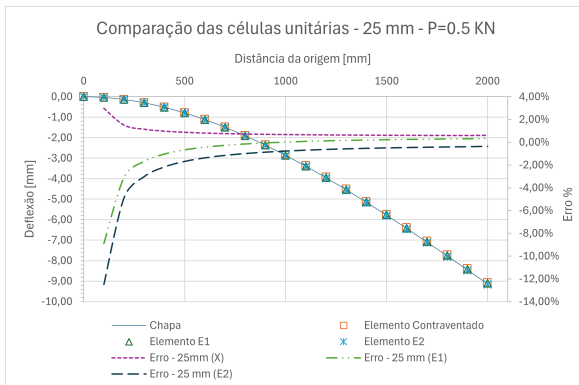
Figura 9 - Comparação das células unitárias para viga em balanço - P=0,5 KN



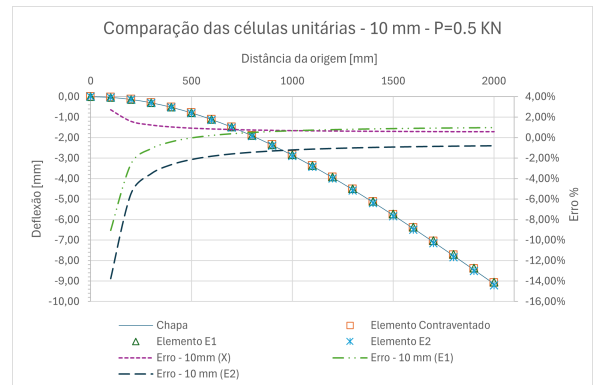
(a)



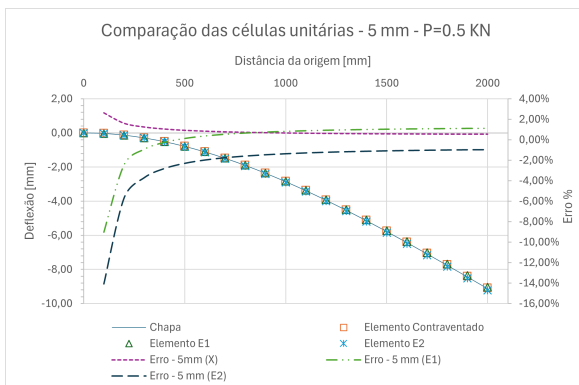
(b)



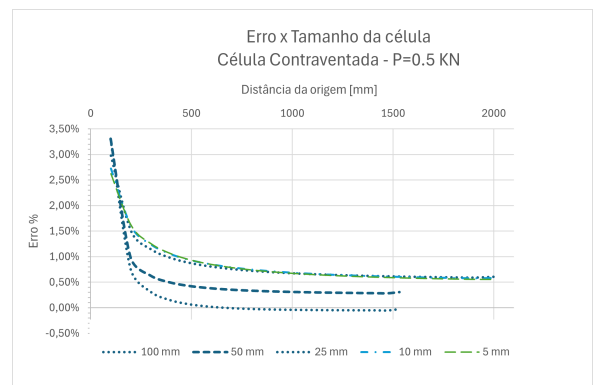
(c)



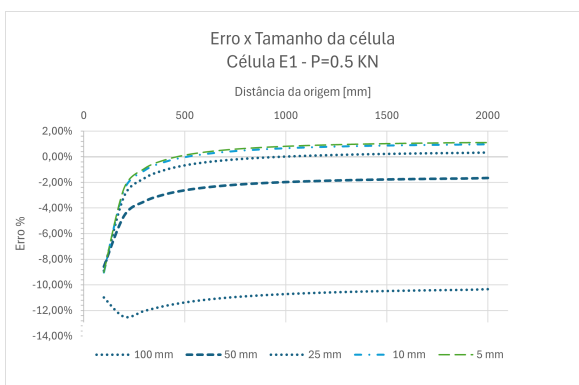
(d)



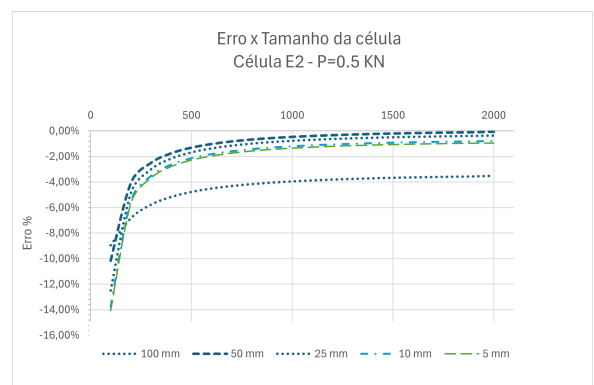
(e)



(f)

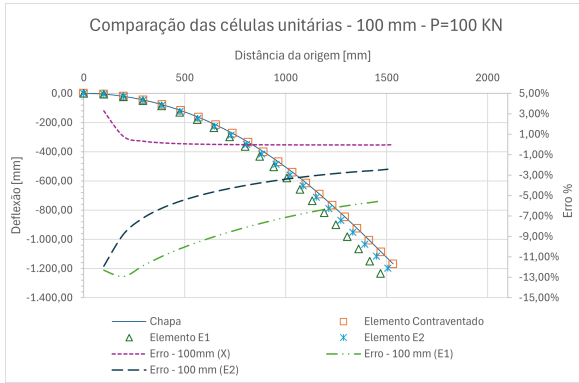


(g)

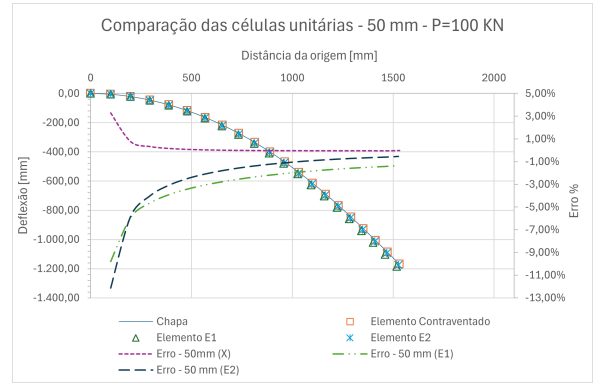


(h)

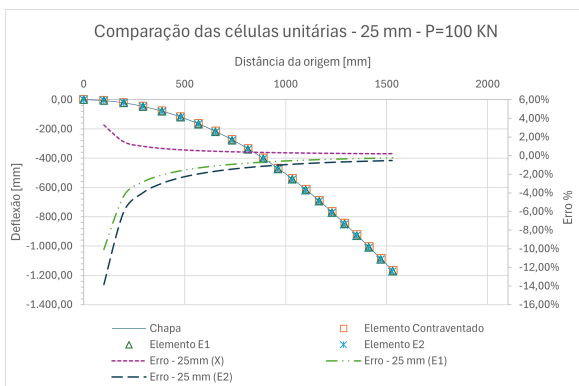
Figura 10 - Comparação das células unitárias para viga em balanço - P=100 KN



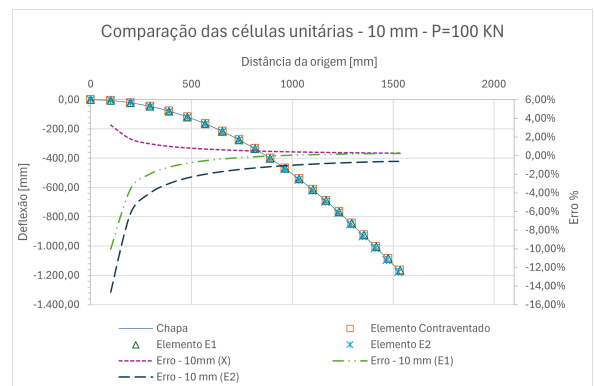
(a)



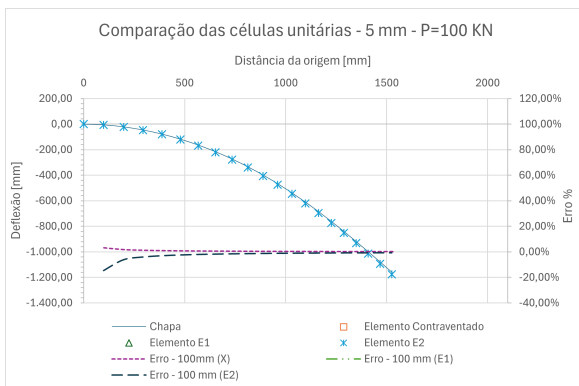
(b)



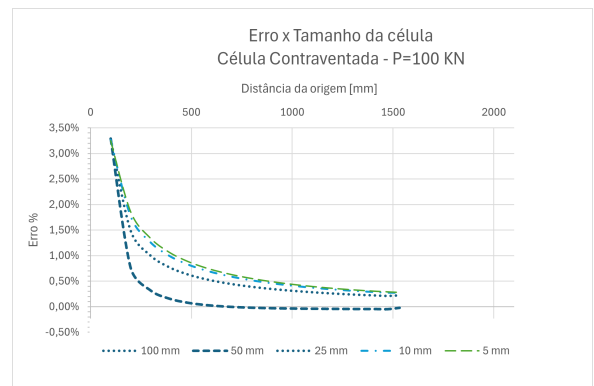
(c)



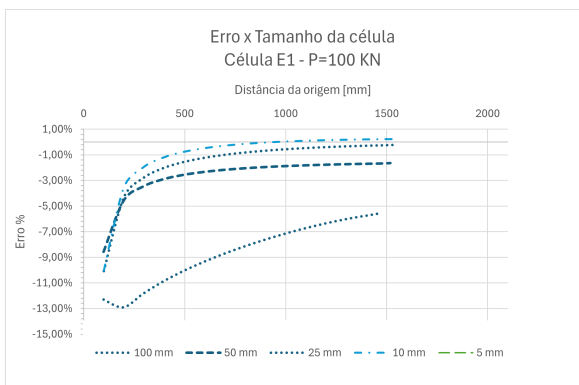
(d)



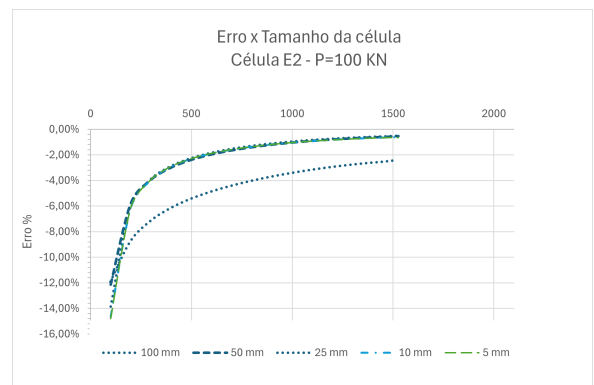
(e)



(f)



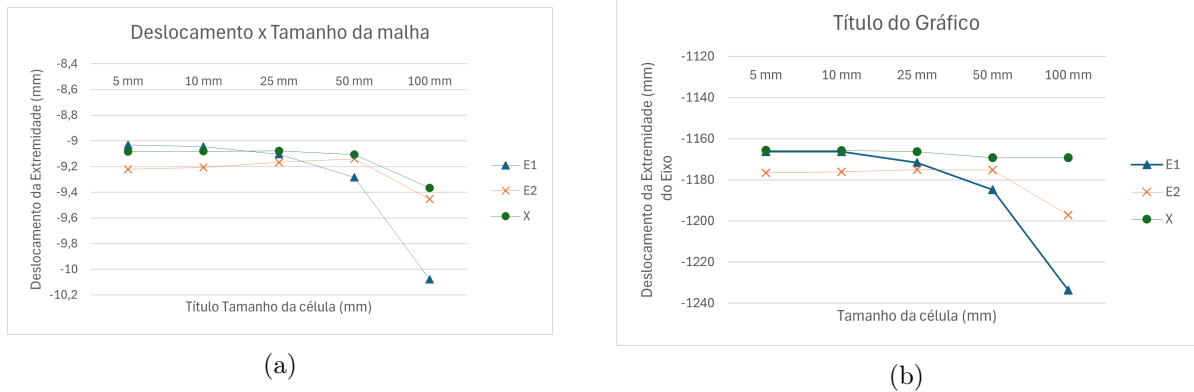
(g)



(h)

Tomando um ponto aleatório, neste caso a extremidade livre da viga, é possível comparar as células quanto sua convergência em função da densidade da malha, o que é mostrado nas figuras 11a e 11b.

Figura 11 - Convergência em função da densidade de malha



Fonte: Autor

5.3.2 Exemplo 2 - Pórtico de Lee

Para o segundo exemplo foi escolhido o pórtico de Lee, mostrado na figura 12. Os parâmetros adotados nas simulações são apresentados na tabela 4, sendo adotados os valores de 200GPa e 0,3 para o módulo de elasticidade e coeficiente de Poisson, respectivamente.

Para este exemplo foram adotadas as células unitárias já mencionadas, em uma estrutura com malha de 10mm. O resultado das simulações são demonstrados nas figuras 13a e 13b

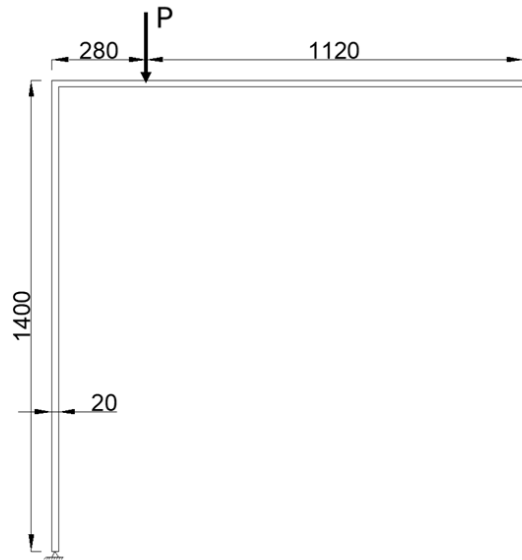
O pórtico de lee foi escolhido por representar os fenômenos de instabilidade por ponto limite (snap-back e snap-trough) , vistos nas figuras 13a 13b) .

Tabela 4 - Parâmetros adotados para o Pórtico de Lee

Célula	Step de		Step de	
	Carga		Deslocamento	
	N° de Steps	Valor do Step (N)	N° de Steps	Valor do Step (mm)
X	75	-50	125	-10
E1	75	-50	625	-2
E2	75	-50	625	-2

Fonte: Autor

Figura 12 - Pórtico de Lee



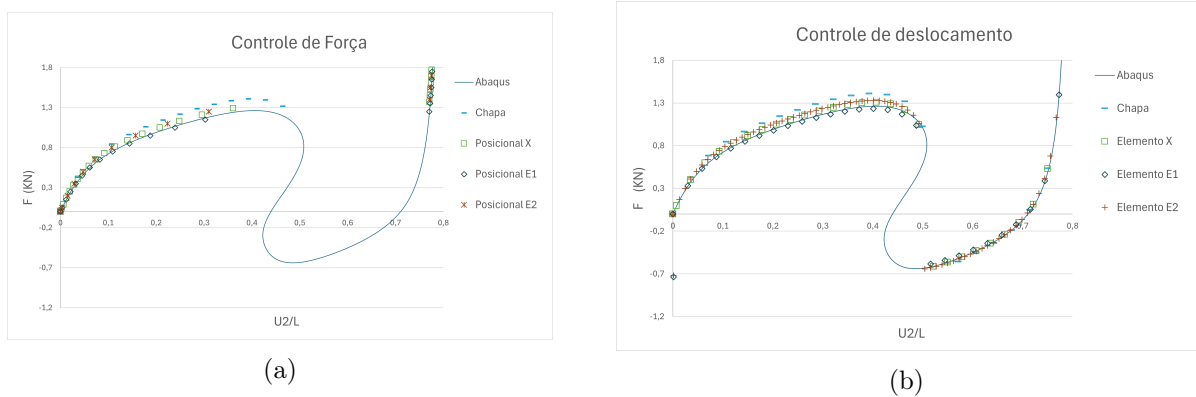
No gráfico da figura 13a fez-se uma análise considerando o controle de força no método de Newton-Rahpson. O controle de força consiste na aplicação gradativa de carregamento de forma quase estática. Verifica-se no gráfico que o controle de força não consegue representar os trechos pós ponto crítico pois trata-se uma característica do método de solução de Newton-Rahpson.

No gráfico da figura 13b fez-se um controle de deslocamento, que consiste na aplicação de deslocamentos prescritos de maneira também gradativa. Nesta estratégia de análise é possível verificar que o controle do deslocamento consegue superar o ponto limite associado ao snap-through, porém, não ao de snap-back, sendo necessário adotar outras estratégia de solução como o de controle de arco (arc-length).

É possível concluir que mesmo com a presença destes fenômenos de instabilidade complexas o modelo proposto foi capaz de reproduzir a trajetória de equilíbrio da estrutura. Na comparação entre as malhas a malha formada por células do tipo E1 foi a que apresentou menor desvio com relação à referência.

Quando aplicado o controle de força, em steps de 50N, os resultados foram consistentes entre os modelos, tanto pré quanto pós snap-back, reforçando a validade do modelo proposto.

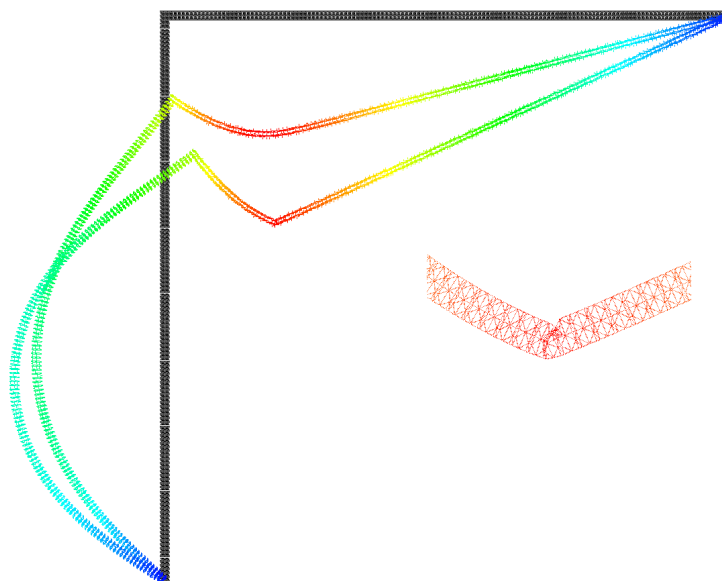
Figura 13 - Pórtico de Lee - Gráfico força x deslocamento



Fonte: Autor

No caso do controle de deslocamento, um fato chama à atenção nos elementos E1 e E2. Em passos maiores do que 1,5mm surge uma instabilidade local, resultando em deformações inesperadas em uma região próxima ao ponto de aplicação do carregamento(14). Em passos inferiores a 1,5mm este fenômeno não foi mais observado, porém, pode justificar o desvio observado para o elemento E2 após o snap-trought uma vez que o modelo tem dificuldade em reestabelecer o equilíbrio na posição esperada. É possível que exista alguma relação entre a geometria da célula e a taxa de deformação, o que explicaria este comportamento depender do passo.

Figura 14 - Colapso local devido ao passo de carregamento

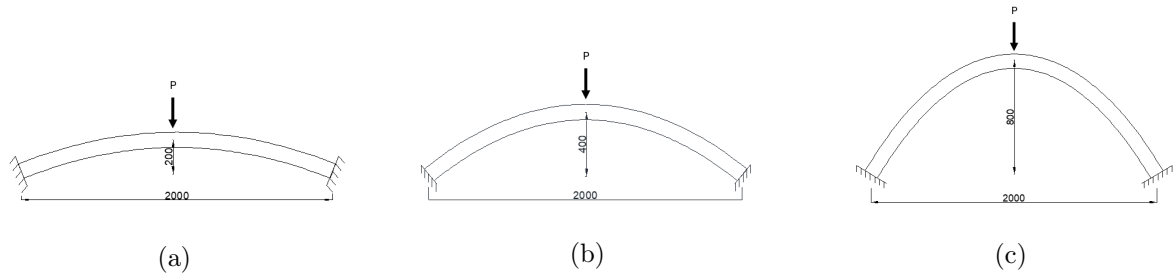


Fonte: Autor

5.3.3 Exemplo 3 - Arco

Foram testados três arcos distintos, identificados como H1(15a), H2(15b) e H3(15c). Em todos eles foram realizadas simulações com passos de carga e deslocamento, considerando as três células propostas.

Figura 15 - Arcos testados



Fonte: Autor

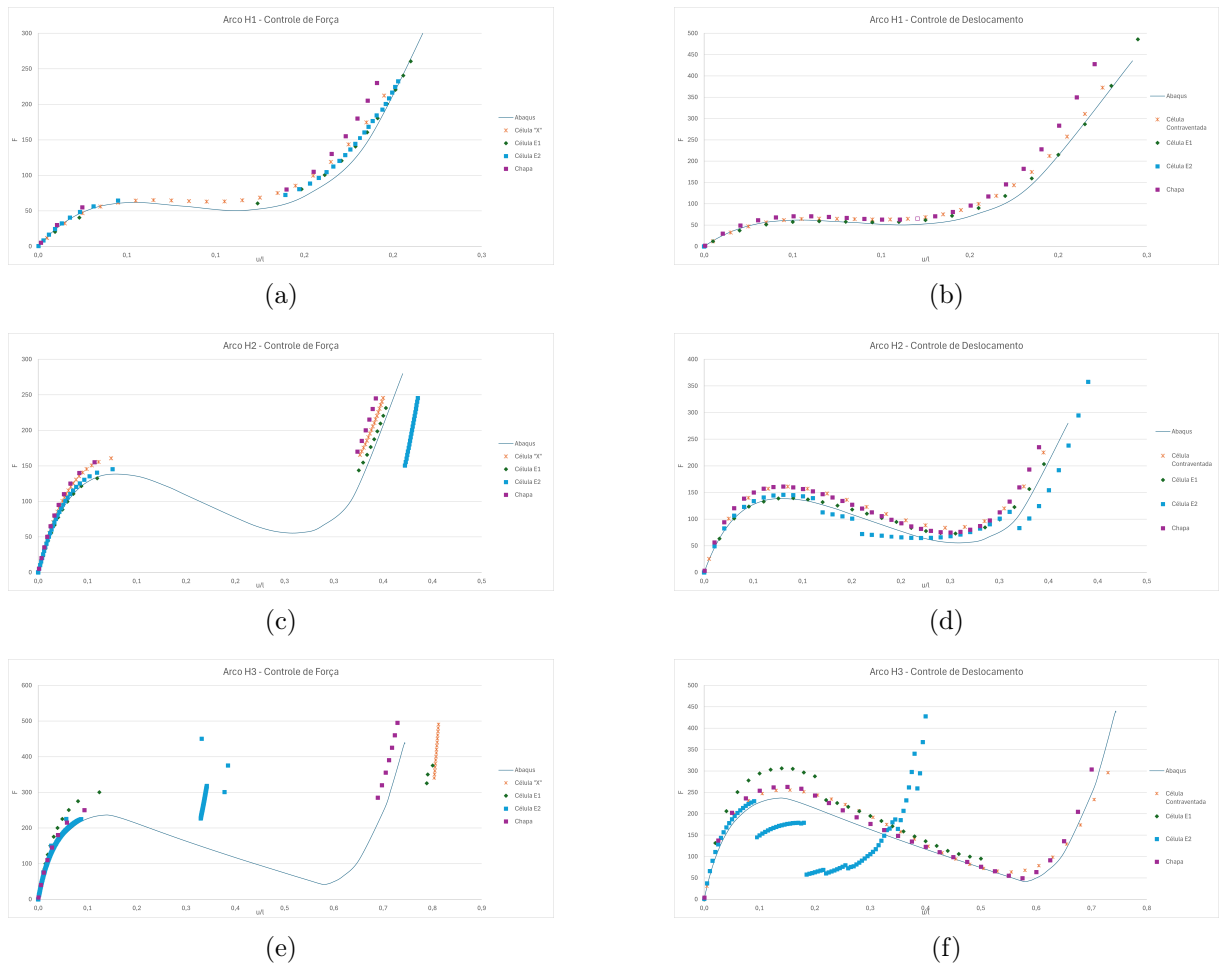
Assim como o pórtico, o arco foi escolhido para verificar o comportamento estrutural de arcos abatidos e não abatidos. Nesse sentido foram modeladas três alturas diferentes e iguais a 200 mm, 400 mm e 800 mm, todos formado a partir de uma viga de 2000 mm x 100 mm de espessura 1 mm, conforme as figuras 15a, 15b e 15c, respectivamente. Uma vez que, ao ser curvada, a viga tem seu comprimento aumentado, o número de células é ajustado de modo que elas não sejam deformadas no processo, mantendo assim uma malha homogênea. A seguir apresenta-se a trajetória de equilíbrio obtida a partir da análise numérica utilizando as células propostas juntamente com o elemento de chapa e modelo gerado pelo software *Abaqus*, para validação dos resultados.

Tabela 5 - Dados das malhas geradas para os Arcos

Célula	H1			H2			H3		
	$A(mm^2)$	Nós	Elem.	$A(mm^2)$	Nós	Elem.	$A(mm^2)$	Nós	Elem.
X	12.20270	415	1398	12.20391	446	1502	12.20683	542	1826
E1	8.41850	1449	4008	8.41902	1596	4414	8.42026	1941	5369
E2	6.06323	2477	7388	6.06347	2660	7934	6.13567	3233	9221

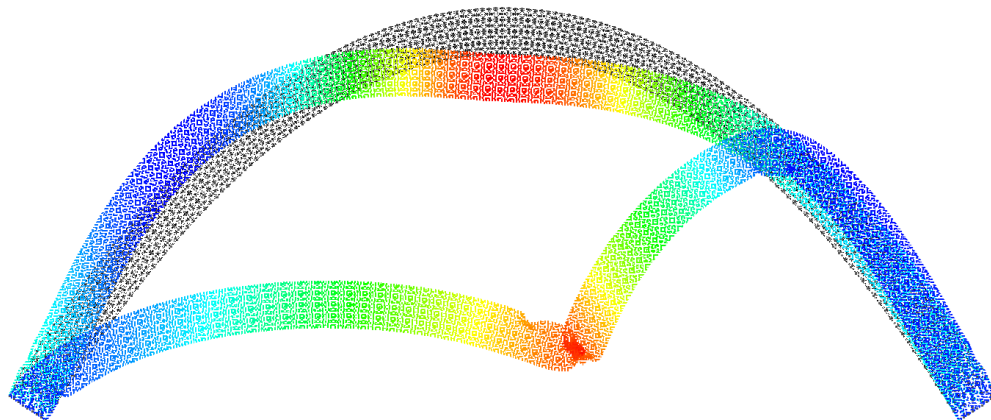
Fonte: Autor

Figura 16 - Comparação das células unitárias para os arcos H1, H2 e H3



Fonte: Autor

Figura 17 - Colapso do arco H3 de malha E2



Fonte: Autor

As trajetórias de equilíbrio para passos de força são apresentadas nas figuras 16a, 16c e 16e , enquanto que as trajetórias de equilíbrio considerando passos de deslocamento são vistos nas figuras 16b,16d e 16f

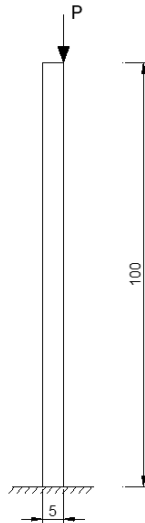
Os testes mostraram que tanto a célula E1 quanto a célula E2 apresentaram o mesmo fenômeno de colapso localizado (fig. 17, próximo ao ponto de aplicação do carregamento, comportamento semelhantemente ao observado no pórtico.

É possível que, neste caso, o resultado obtido esteja relacionado à forma com que a malha dos arcos foi gerada. Estas malhas tem origem em um viga que é então parametrizada, causando distorções nas células unitárias, e portanto, afetando seus coeficientes de forma uma vez que a malha deixa de ser periódica.

5.3.4 Exemplo 4 - Coluna com carregamento excêntrico

O último caso simulado é o da coluna com um carregamento excêntrico mostrada na figura 18). Sobre uma coluna engastada na base é aplicada uma excêntrica de intensidade P , em *steps* de 30N. A coluna possui módulo de elasticidade $E=12\text{Gpa}$ e $\nu=0,3$.

Figura 18 - Coluna submetida a carregamento excêntrico



Fonte: Autor

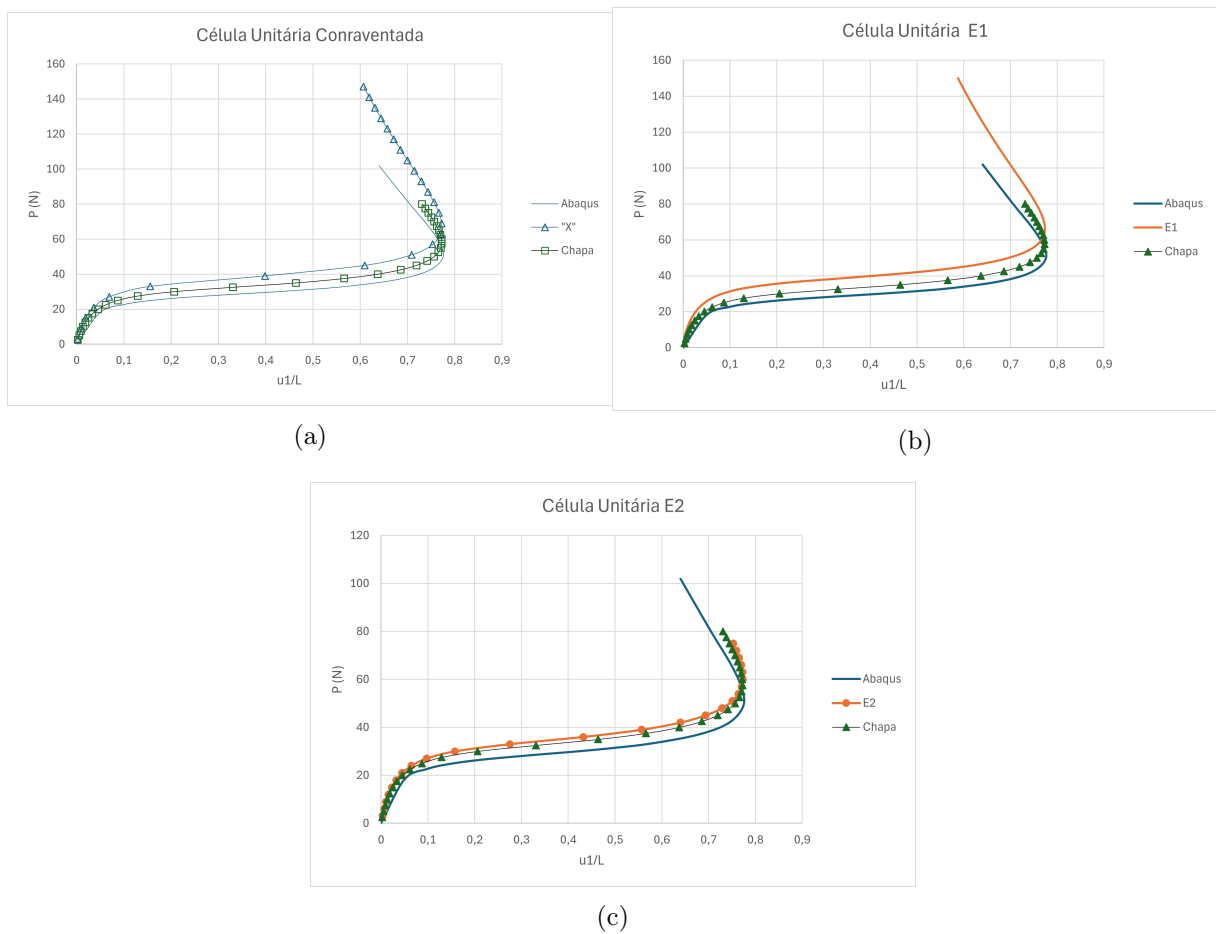
Assim como no pórtico de Lee, o elemento que apresentou pior desempenho foi a célula unitária E1 enquanto as células contraventada e E2 apresentaram desempenho próximo ao esperado para este tipo de carregamento.

Tabela 6 - Dados das malhas geradas para a coluna

Célula	N° de Células	A (mm ²)	N° de Nós	N° Elem.
X	2000	0,36720	2211	8210
E1	2000	0,21489	8421	24420
E2	2000	0,14207	14421	44420

Fonte: Autor

Figura 19 - Deslocamento horizontal normalizado em função da carga



Fonte: Autor

5.3.5 Fechamento

As células foram testadas admitindo a orientação apresentada na figura 4. Com base no observado fica evidente que, para materiais isotrópicos, é necessário que a célula unitária seja simétrica com relação aos eixos que passam pelo seu centróide.

6 CONCLUSÃO

Este trabalho partiu da premissa de que seria possível representar o meio contínuo através de um arranjo periódico de elementos lineares de barra na forma de uma estrutura treliçada.

Os primeiros testes apontaram que, através de uma combinação de parâmetros como o tamanho da célula unitária e a área da seção transversal de suas barras, seria possível obter um comportamento mecânico equivalente entre uma estrutura contínua e uma estrutura treliçada. Em face deste comportamento, optou-se por um modelo que considera a equiparação energética entre contínuo e discreto.

Foram propostas três células unitárias com quantidade de nós e barras distintas bem como suas orientações. Em todos os casos foi possível determinar a partir do modelo proposto um valor específico para a área da seção transversal destas barras de modo que o comportamento do meio contínuo se fizesse representado por elas. Ainda que com dimensões variadas, foi possível através dos chamados parâmetros de forma, manter os resultados consistentes em diferentes escalas.

Vale reforçar que a grande vantagem do modelo proposto reside em sua relativa simplicidade, tanto por sua formulação quanto por seu processamento pois, a partir de uma única célula unitária, foi possível representar o meio contínuo bidimensional conforme proposto inicialmente. O código gerado é capaz de analisar malhas densas a um baixo custo computacional, podendo ser aprimorado através do processamento em GPU ao invés de CPU.

Posto isso, convém apresentar algumas sugestões para trabalhos futuros: Colapso progressivo da estrutura, novos modelos para determinar os coeficientes de forma, problemas tridimensionais. Investigar a relação entre taxa de carregamento e geometria da célula unitária.

REFERÊNCIAS

- BABU, B.; PATEL, B. P. An improved quadrilateral finite element for nonlinear second-order strain gradient elastic Kirchhoff plates. *Meccanica*, v. 55, n. 1, p. 139–159, 1 2020. ISSN 0025-6455.
- BATHE, K. J. *Finite Element Procedures*. Prentice Hall, 2006. ISBN 9780979004902. Disponível em: <<https://books.google.com.br/books?id=rWvefGICfO8C>>.
- CHALLAMEL, N. et al. Discrete and continuous models of linear elasticity: history and connections. *Continuum Mechanics and Thermodynamics*, v. 35, n. 2, p. 347–391, 3 2023. ISSN 0935-1175.
- CODA, H. B. *O método dos elementos finitos posicional: sólidos e estruturas - não linearidade geométrica e dinâmica*. São Carlos: EESC/USP, 2018.
- CODA, H. B.; GRECO, M. A simple FEM formulation for large deflection 2D frame analysis based on position description. *Computer Methods in Applied Mechanics and Engineering*, North-Holland, v. 193, n. 33-35, p. 3541–3557, 8 2004. ISSN 0045-7825.
- Dassault Systèmes. *ABAQUS Analysis User's Guide*. 2025. Disponível em: <https://help.3ds.com/2025x/English/DSDoc/FrontmatterMap/DSDocHome.htm?contextscope=cloud&redirect_lang=English>.
- GASPARETTO, V.; ELSAYED, M. Multiscale Modelling and Mechanical Anisotropy of Periodic Cellular Solids with Rigid-Jointed Truss-Like Microscopic Architecture. *Applied Mechanics*, v. 2, n. 2, p. 331–355, 6 2021. ISSN 2673-3161.
- GEORGES, H.; MITTELSTEDT, C.; BECKER, W. RVE-based grading of truss lattice cores in sandwich panels. *Archive of Applied Mechanics*, v. 93, n. 8, p. 3189–3203, 8 2023. ISSN 0939-1533.
- GLAESENER, R. N. et al. Continuum representation of nonlinear three-dimensional periodic truss networks by on-the-fly homogenization. *International Journal of Solids and Structures*, Elsevier Ltd, v. 206, p. 101–113, 12 2020. ISSN 00207683.
- Grand View Research. *Metamaterial Market Size, Share & Trends Analysis Report By Product (Electromagnetic, Terahertz, Photonic, Tunable, Frequency Selective Surface, Non-linear), By Application, End-use, And Segment Forecasts, 2018 - 2025*. 2016.
- GRECO, M. *Análise de problemas de contato/impacto em estruturas de comportamento não linear pelo método dos elementos finitos*. Tese (Doutorado), 4 2004.
- GU, H.; SHTERENLIKHT, A.; PAVIER, M. Brittle fracture of three-dimensional lattice structure. *Engineering Fracture Mechanics*, Pergamon, v. 219, p. 106598, 10 2019. ISSN 0013-7944. Disponível em: <<https://doi-org.ez350.periodicos.capes.gov.br/10.1016/j.engfracmech.2019.106598>>.
- GUAN, C. et al. A homogenization method for natural frequencies and damping analysis of composite pyramidal lattice truss sandwich structures based on representative volume elements. *Journal of Sandwich Structures & Materials*, v. 25, n. 5, p. 572–590, 6 2023. ISSN 1099-6362.

HABERLAND, S. et al. Representative Volume Element Approximations in Elastoplastic Spring Networks. *Multiscale Modeling & Simulation*, v. 22, n. 1, p. 588–638, 3 2024. ISSN 1540-3459.

HASSANI, B.; HINTON, E. *Homogenization and Structural Topology Optimization*. London: Springer London, 1999. ISBN 978-1-4471-1229-7.

JAGIELLO, E.; MUÑOZ-ROJAS, P. A. An Extended Multiscale Finite Element Method (EMsFEM) Analysis of Periodic Truss Metamaterials (PTMM) Designed by Asymptotic Homogenization. *Latin American Journal of Solids and Structures*, v. 18, n. 2, 2021. ISSN 1679-7825.

KALAMKAROV, A. L.; ANDRIANOV, I. V.; DANISHEVS'KYY, V. V. Asymptotic Homogenization of Composite Materials and Structures. *Applied Mechanics Reviews*, v. 62, n. 3, 5 2009. ISSN 0003-6900.

KOJIĆ, M.; MILOŠEVIĆ, M.; ZIEMYS, A. Smearred finite element formulation for mechanics. *Computational Models in Biomedical Engineering*, Academic Press, p. 65–103, 1 2023.

KZAM, A. K. L. *Análise da instabilidade estrutural global e local pelo MEF posicional com determinação de pontos críticos na trajetória de equilíbrio*. Tese (Doutorado), 2016.

LIU, H.; LV, J. An equivalent continuum multiscale formulation for 2D geometrical nonlinear analysis of lattice truss structure. *Composite Structures*, v. 160, p. 335–348, 1 2017. ISSN 02638223.

LIU, J. et al. Three-Node Euler-Bernoulli Beam Element Based on Positional FEM. *Procedia Engineering*, No longer published by Elsevier, v. 29, p. 3703–3707, 1 2012. ISSN 1877-7058.

MIAO, X. et al. Review on mechanical properties of metal lattice structures. *Composite Structures*, v. 342, p. 118267, 8 2024. ISSN 02638223.

MOHAMMADI, A. et al. Asymptotic homogenization of tesseract lattice structures. *European Journal of Mechanics - A/Solids*, v. 107, p. 105368, 9 2024. ISSN 09977538.

NAYFEH, A. H.; HEFZY, M. S. Continuum modeling of three-dimensional truss-like space structures. *AIAA Journal*, v. 16, n. 8, p. 779–787, 8 1978. ISSN 0001-1452.

NOOR, A. K. Continuum Modeling for Repetitive Lattice Structures. *Applied Mechanics Reviews*, v. 41, n. 7, p. 285–296, 7 1988. ISSN 0003-6900.

PARK, M.; VENTER, M. P.; PLESSIS, A. D. Simulation of the Compression Testing of Additively Manufactured Lattice Structures Using Inputs from Microcomputed Tomography. *Material Design & Processing Communications*, v. 2023, p. 1–15, 11 2023. ISSN 2577-6576.

PHLIPOT, G. P.; KOCHMANN, D. M. A quasicontinuum theory for the nonlinear mechanical response of general periodic truss lattices. *Journal of the Mechanics and Physics of Solids*, v. 124, p. 758–780, 3 2019. ISSN 00225096.

- RÁŽ, K.; CHVAL, Z.; PEREIRA, M. Lattice Structures—Mechanical Description with Respect to Additive Manufacturing. *Materials*, v. 17, n. 21, p. 5298, 10 2024. ISSN 1996-1944.
- REIS, F. D.; GANGHOFFER, J. Homogenized elastoplastic response of repetitive 2D lattice truss materials. *Computational Materials Science*, v. 84, p. 145–155, 3 2014. ISSN 09270256.
- ROMIJN, N. E.; FLECK, N. A. The fracture toughness of planar lattices: Imperfection sensitivity. *Journal of the Mechanics and Physics of Solids*, Pergamon, v. 55, n. 12, p. 2538–2564, 12 2007. ISSN 0022-5096. Disponível em: <<https://doi-org.ez350.periodicos.capes.gov.br/10.1016/j.jmps.2007.04.010>>.
- SADD, M. H. *Elasticity: Theory, Applications, and Numerics*. Academic Press, 2009. ISBN 9780080922416. Disponível em: <<https://books.google.co.ao/books?id=Kl7wLSF4730C>>.
- SOARES, H. B.; PACCOLA, R. R.; CODA, H. B. Unconstrained Vector Positional Shell FEM formulation applied to thin-walled members instability analysis. *Thin-Walled Structures*, Elsevier, v. 136, p. 246–257, 3 2019. ISSN 0263-8231.
- SOMNIC, J.; JO, B. W. Status and Challenges in Homogenization Methods for Lattice Materials. *Materials*, v. 15, n. 2, p. 605, 1 2022. ISSN 1996-1944.
- SPINA, R. et al. Homogenization simulation of material extruded lattice structures. *Heliyon*, v. 9, n. 2, p. e13539, 2 2023. ISSN 24058440.
- TELES, D. V. C.; AMORIM, D. L. N. F.; LEONEL, E. D. *MECÂNICA DO DANO CONCENTRADO EXPANDIDA PARA MEIOS BIDIMENSIONAIS: ABORDAGEM GEOMETRICAMENTE EXATA LUMPED DAMAGE MECHANICS EXPANDED FOR TWO-DIMENSIONAL DOMAINS: GEOMETRICALLY EXACT APPROACH*. [S.l.], 2023. Disponível em: <<http://www.ipd.ufs.br>>.
- VIGLIOTTI, A.; PASINI, D. Linear multiscale analysis and finite element validation of stretching and bending dominated lattice materials. *Mechanics of Materials*, v. 46, p. 57–68, 3 2012. ISSN 01676636.
- XU, S.; ZHANG, W. Energy-based homogenization method for lattice structures with generalized periodicity. *Computers & Structures*, v. 302, p. 107478, 10 2024. ISSN 00457949.
- YANG, H. et al. Mechanical properties of hierarchical lattice via strain gradient homogenization approach. *Composites Part B: Engineering*, v. 271, p. 111153, 2 2024. ISSN 13598368.
- ZHANG, J. et al. Multiscale topology optimization of anisotropic multilayer periodic structures based on the isogeometric analysis method. *Meccanica*, v. 59, n. 11, p. 1939–1959, 11 2024. ISSN 0025-6455.
- ZHANG, Q. et al. Ultralight X-type lattice sandwich structure (II): Micromechanics modeling and finite element analysis. *Science in China Series E: Technological Sciences*, v. 52, n. 9, p. 2670–2680, 9 2009. ISSN 1006-9321.

ZHANG, X.; O'BRIEN, D. J.; GHOSH, S. Parametrically homogenized continuum damage mechanics (PHCDM) models for unidirectional composites with nonuniform microstructural distributions. *Journal of Computational Physics*, Academic Press Inc., v. 435, 6 2021. ISSN 10902716.

ZHANG, Y. et al. Design and characterization of multi-stable mechanical metastructures with level and tilted stable configurations. *Extreme Mechanics Letters*, v. 34, p. 100593, 1 2020. ISSN 23524316.

ZHENG, L. et al. Unifying the design space and optimizing linear and nonlinear truss metamaterials by generative modeling. *Nature Communications*, v. 14, n. 1, p. 7563, 11 2023. ISSN 2041-1723.

ZSCHERNACK, C.; WADEE, M. A.; VÖLLMECKE, C. Nonlinear buckling of fibre-reinforced unit cells of lattice materials. *Composite Structures*, v. 136, p. 217–228, 2 2016. ISSN 02638223.

APÊNDICE A – Código para geração de malha e Discretização

A.1 Gerador de malha

```

1 program Gera_malha
2     implicit none
3
4     !
5     ↪ =====
6     ! DECLARAÇÃO DE VARIÁVEIS
7     !
8     ↪ =====
9
10    real, allocatable :: coordenadas_base(:, :), poligono_pontos(:, :),
11    ↪ coordenadas_malha(:, :), nos_unicos(:, :)
12    integer, allocatable :: incidencias_base(:, :),
13    ↪ incidencias_malha(:, :), mapeamento(:), incidencias_unicas(:, :)
14    integer :: num_coord, num_incid, num_poligono_pontos, i, j, k, io,
15    ↪ unit_out, total_nos, total_incidencias
16    integer :: nx, ny, num_unicos, num_incid_unicas, m, p, cell_id,
17    ↪ node_id1, node_id2, engaste
18    real :: x, y, z, tamanho_x_celula, tamanho_y_celula, min_x_poligono,
19    ↪ max_x_poligono, min_y_poligono, max_y_poligono
20    real :: tol, mid_x, mid_y, m_h, m_v
21    character(len=50) :: entrada
22    logical :: found, incidencia_existe
23    real, allocatable :: nos_sobre_aresta(:, :)
24    integer :: num_nos_sobre_aresta, edge_start, edge_end, Nh, Nv, ln_no,
25    ↪ count_ln, sn
26    integer, dimension(21) :: l_n=0
27    real(8) :: area_calculada
28    real(8) :: L_mm, eps_x, gamma_xy, coef_v, coef_h
29
30    !
31    ↪ =====
32    ! ENTRADA DOS DADOS DE CÉLULA E GEOMETRIA

```

```

25 !
    ↪ =====
26
27 ln_no=0
28 count_ln=1
29
30 print *, '=====|'
31 print *, 'INICIO DO PROCESSO INTEGRADO|'
32 print *, '=====|'
33 print *, ' '
34     print *, '--- GEOMETRIA DA CELULA BASE / TRELICA ---|'
35 print *, 'Entre com os nos da celula base (ID X Y Z) e apos "exit"
    ↪ para encerrar:'
36 num_coord = 0
37 do
38     read(*, '(A)') entrada
39     if (trim(entrada) == 'exit') exit
40     num_coord = num_coord + 1
41     if (.not. allocated(coordenadas_base)) then
42         allocate(coordenadas_base(num_coord, 4))
43     else if (size(coordenadas_base, 1) < num_coord) then
44         call resize_real_array(coordenadas_base, num_coord + 10, 4)
45     end if
46     read(entrada, *, iostat=io) i, x, y, z
47     if (io /= 0) then
48         print *, 'Erro na entrada. Tente novamente.'
49         num_coord = num_coord - 1
50         cycle
51     end if
52     coordenadas_base(num_coord, 1) = real(i)
53     coordenadas_base(num_coord, 2) = nint(x * 100.0) / 100.0
54     coordenadas_base(num_coord, 3) = nint(y * 100.0) / 100.0
55     coordenadas_base(num_coord, 4) = z
56 end do
57
58 print *, 'Entre com as incidências da célula base (Nó1 Nó2) e após
    ↪ "exit" para encerrar:'
59 num_incid = 0
60 do

```

```

61     read(*, '(A)') entrada
62     if (trim(entrada) == 'exit') exit
63     num_incid = num_incid + 1
64     if (.not. allocated(incidencias_base)) then
65         allocate(incidencias_base(num_incid, 2))
66     else if (size(incidencias_base, 1) < num_incid) then
67         call resize_int_array(incidencias_base, num_incid + 10, 2)
68     end if
69     read(entrada, *, iostat=io) i, j
70     if (io /= 0) then
71         print *, 'Erro na entrada. Tente novamente.'
72         num_incid = num_incid - 1
73         cycle
74     end if
75     incidencias_base(num_incid, 1) = i
76     incidencias_base(num_incid, 2) = j
77 end do

78
79 print *, ' '
80 print *, '--- GEOMETRIA DO POLIGONO DE CONTORNO ---'
81 print *, 'Entre com as coordenadas do polígono (ID X Y Z) ou "exit"
82     ↪ para encerrar:'
83 num_poligono_pontos = 0
84 do
85     read(*, '(A)') entrada
86     if (trim(entrada) == 'exit') exit
87     num_poligono_pontos = num_poligono_pontos + 1
88     if (.not. allocated(poligono_pontos)) then
89         allocate(poligono_pontos(num_poligono_pontos, 4))
90     else if (size(poligono_pontos, 1) < num_poligono_pontos) then
91         call resize_real_array(poligono_pontos, num_poligono_pontos +
92             ↪ 10, 4)
93     end if
94     read(entrada, *, iostat=io) i, x, y, z
95     if (io /= 0) then
96         print *, 'Erro na entrada. Tente novamente.'
97         num_poligono_pontos = num_poligono_pontos - 1
98         cycle
99     end if

```

```

98     poligono_pontos(num_poligono_pontos, 1) = real(i)
99     poligono_pontos(num_poligono_pontos, 2) = nint(x * 100.0) / 100.0
100    poligono_pontos(num_poligono_pontos, 3) = nint(y * 100.0) / 100.0
101    poligono_pontos(num_poligono_pontos, 4) = z
102  end do
103  print *, '--- PARAMETROS PARA O CALCULO DA AREA ---'
104
105  print *, 'Deseja entrar com os coeficientes manualmente? Sim (1) /Nao
106  ↪ (2)'
107  read*,sn
108  if (sn==1) then
109      print *, 'Digite os coeficientes'
110      read*,coef_h,coef_v
111      goto 1
112  endif
113
114  print *, ' '
115  print *, '--- CALCULO AUTOMATICO DOS FATORES m_h e m_v ---'
116  call calcula_mh_mv(coordenadas_base, num_coord, incidencias_base,
117  ↪ num_incid, m_h, m_v)
118  print *, 'Fatores calculados -> m_h:', m_h, ' m_v:', m_v
119  print *, ' '
120
121  print *, 'Digite Nh e Nv (numero de celulas na direcao H e V):'
122  read *, Nh, Nv
123
124  coef_h=(m_h)/((m_h-1)*(Nv-1)+1)
125  coef_v=(m_v)/((m_v-1)*(Nh-1)+1)
126
127  1 print *, 'Digite o lado da chapa (em mm):'
128  read *, L_mm
129
130  print *, 'Digite as deformações (eps_x, gamma_xy):'
131  read *, eps_x, gamma_xy
132
133  print *, ' '
134  !
135  ↪ =====

```

```

134 ! CÁLCULO DA ÁREA DA SEÇÃO TRANSVERSAL ATRAVÉS DO PROCESSO DE
    ↳ EQUIVALÊNCIA ENERGÉTICA
135 !
    ↳ =====
136 call calcula_area_equivalente(area_calculada, L_mm, eps_x, gamma_xy,
    ↳ coef_h, coef_v, &
137                                coordenadas_base, num_coord,
    ↳                                incidencias_base, num_incid)
138
139 !
    ↳ =====
140 ! GERAÇÃO DA MALHA
141 !
    ↳ =====
142 print *, ' '
143 print *, '=====|'
144 print *, 'INICIO DA GERACAO DA MALHA|'
145 print *, '=====|'
146
147 tol = 1e-2 ! Tolerância para 2 casas decimais
148
149 ! Calcular tamanho da célula base
150 tamanho_x_celula = maxval(coordenadas_base(1:num_coord, 2)) -
    ↳ minval(coordenadas_base(1:num_coord, 2))
151 tamanho_y_celula = maxval(coordenadas_base(1:num_coord, 3)) -
    ↳ minval(coordenadas_base(1:num_coord, 3))
152 if (tamanho_x_celula <= 0.0 .or. tamanho_y_celula <= 0.0) then
153     print *, 'Erro: Célula base com tamanho inválido.'
154     stop
155 end if
156
157 ! Calcular dimensões do polígono
158 min_x_poligono = minval(poligono_pontos(1:num_poligono_pontos, 2))
159 max_x_poligono = maxval(poligono_pontos(1:num_poligono_pontos, 2))
160 min_y_poligono = minval(poligono_pontos(1:num_poligono_pontos, 3))
161 max_y_poligono = maxval(poligono_pontos(1:num_poligono_pontos, 3))
162
163 ! Calcular número de células em cada direção
164 nx = ceiling((max_x_poligono - min_x_poligono) / tamanho_x_celula)

```

```

165 ny = ceiling((max_y_poligono - min_y_poligono) / tamanho_y_celula)
166
167 ! Estimar total máximo de nós e incidências
168 total_nos = num_coord * (nx + 1) * (ny + 1)
169 total_incidencias = num_incid * nx * ny + total_nos
170
171 ! Alocar arrays da malha
172 allocate(coordenadas_malha(total_nos, 7))
173 allocate(incidencias_malha(total_incidencias, 2))
174 allocate(mapeamento(total_nos))
175
176 ! Gerar nós da malha com eliminação de duplicatas
177 total_nos = 0
178 allocate(nos_unicos(0, 7))
179 num_unicos = 0
180 do j = 0, ny
181     do i = 0, nx
182         do k = 1, num_coord
183             x = min_x_poligono + i * tamanho_x_celula +
184                 ↪ (coordenadas_base(k, 2) -
185                 ↪ minval(coordenadas_base(1:num_coord, 2)))
186             y = min_y_poligono + j * tamanho_y_celula +
187                 ↪ (coordenadas_base(k, 3) -
188                 ↪ minval(coordenadas_base(1:num_coord, 3)))
189             x = nint(x * 100.0) / 100.0
190             y = nint(y * 100.0) / 100.0
191             if (x <= max_x_poligono + tol .and. y <= max_y_poligono +
192                 ↪ tol .and. &
193                 ↪ point_in_polygon(x, y, poligono_pontos,
194                 ↪ num_poligono_pontos)) then
195                 total_nos = total_nos + 1
196                 coordenadas_malha(total_nos, 1) = real(total_nos)
197                 coordenadas_malha(total_nos, 2) = x
198                 coordenadas_malha(total_nos, 3) = y
199                 coordenadas_malha(total_nos, 4) = 0.0
200                 coordenadas_malha(total_nos, 5) = 0.0
201                 coordenadas_malha(total_nos, 6) = 0.0
202                 coordenadas_malha(total_nos, 7) = real((i + j * (nx +
203                 ↪ 1)) * num_coord + k)

```

```

197
198         found = .false.
199         do m = 1, num_unicos
200             if (abs(x - nos_unicos(m, 2)) < tol .and. abs(y -
                ↪ nos_unicos(m, 3)) < tol) then
201                 mapeamento(total_nos) = m
202                 found = .true.
203                 exit
204             end if
205         end do
206         if (.not. found) then
207             num_unicos = num_unicos + 1
208             if (num_unicos > size(nos_unicos, 1)) call
                ↪ resize_real_array(nos_unicos, num_unicos +
                ↪ 100, 7)
209             nos_unicos(num_unicos, 1) = real(num_unicos)
210             nos_unicos(num_unicos, 2:4) =
                ↪ coordenadas_malha(total_nos, 2:4)
211             nos_unicos(num_unicos, 5:7) = 0.0
212             mapeamento(total_nos) = num_unicos
213         end if
214     end if
215 end do
216 end do
217 end do
218
219 ! Gerar incidências da malha
220 total_incidencias = 0
221 do j = 0, ny - 1
222     do i = 0, nx - 1
223         do k = 1, num incid
224             node_id1 = (i + j * (nx + 1)) * num_coord +
                ↪ incidencias_base(k, 1)
225             node_id2 = (i + j * (nx + 1)) * num_coord +
                ↪ incidencias_base(k, 2)
226
227             m = 0; p = 0
228             do io = 1, total_nos

```

```

229         if (abs(coordenadas_malha(io, 7) - real(node_id1)) <
↪ 1e-5) m = mapeamento(io)
230         if (abs(coordenadas_malha(io, 7) - real(node_id2)) <
↪ 1e-5) p = mapeamento(io)
231         if (m > 0 .and. p > 0) exit
232     end do
233
234     if (m > 0 .and. p > 0 .and. m /= p) then
235         mid_x = (nos_unicos(m, 2) + nos_unicos(p, 2)) / 2.0
236         mid_y = (nos_unicos(m, 3) + nos_unicos(p, 3)) / 2.0
237         if (point_in_polygon(mid_x, mid_y, poligono_pontos,
↪ num_poligono_pontos)) then
238             incidencia_existe = .false.
239             do io = 1, total_incidencias
240                 if ((incidencias_malha(io, 1) == m .and.
↪ incidencias_malha(io, 2) == p) .or. &
241                     (incidencias_malha(io, 1) == p .and.
↪ incidencias_malha(io, 2) == m)) then
242                     incidencia_existe = .true.
243                     exit
244                 end if
245             end do
246             if (.not. incidencia_existe) then
247                 total_incidencias = total_incidencias + 1
248                 if (total_incidencias >
↪ size(incidencias_malha, 1)) then
249                     call resize_int_array(incidencias_malha,
↪ total_incidencias + 100, 2)
250                 end if
251                 incidencias_malha(total_incidencias, 1) = m
252                 incidencias_malha(total_incidencias, 2) = p
253             end if
254         end if
255     end if
256 end do
257 end do
258 end do
259
260 ! Verificar e criar incidências para nós sobre arestas

```

```

261 allocate(nos_sobre_aresta(num_unicos, 8))
262 do edge_start = 1, num_poligono_pontos
263     edge_end = mod(edge_start, num_poligono_pontos) + 1
264     num_nos_sobre_aresta = 0
265     do i = 1, num_unicos
266         x = nos_unicos(i, 2)
267         y = nos_unicos(i, 3)
268         if (point_on_edge(x, y, poligono_pontos(edge_start, 2),
269             ↪ poligono_pontos(edge_start, 3), &
270                 poligono_pontos(edge_end, 2),
271                 ↪ poligono_pontos(edge_end, 3))) then
272             num_nos_sobre_aresta = num_nos_sobre_aresta + 1
273             nos_sobre_aresta(num_nos_sobre_aresta, 1) = nos_unicos(i,
274                 ↪ 1)
275             nos_sobre_aresta(num_nos_sobre_aresta, 2:7) =
276                 ↪ nos_unicos(i, 2:7)
277             nos_sobre_aresta(num_nos_sobre_aresta, 8) = sqrt((x -
278                 ↪ poligono_pontos(edge_start, 2))**2 + &
279                 (y -
280                 ↪ poligono_ponto
281                 ↪ 3))**2)
282         end if
283     end do
284
285     if (num_nos_sobre_aresta > 1) call sort_nodes(nos_sobre_aresta,
286         ↪ num_nos_sobre_aresta)
287
288     do i = 1, num_nos_sobre_aresta - 1
289         node_id1 = nint(nos_sobre_aresta(i, 1))
290         node_id2 = nint(nos_sobre_aresta(i + 1, 1))
291         if (node_id1 /= node_id2) then
292             incidencia_existe = .false.
293             do k = 1, total_incidencias
294                 if ((incidencias_malha(k, 1) == node_id1 .and.
295                     ↪ incidencias_malha(k, 2) == node_id2) .or. &
296                     (incidencias_malha(k, 1) == node_id2 .and.
297                     ↪ incidencias_malha(k, 2) == node_id1)) then
298                 incidencia_existe = .true.
299             end if
300             exit

```

```

290         end if
291     end do
292     if (.not. incidencia_existe) then
293         total_incidencias = total_incidencias + 1
294         if (total_incidencias > size(incidencias_malha, 1))
295             ↪ then
296                 call resize_int_array(incidencias_malha,
297                     ↪ total_incidencias + 100, 2)
298             end if
299             incidencias_malha(total_incidencias, 1) = node_id1
300             incidencias_malha(total_incidencias, 2) = node_id2
301         end if
302     end if
303 end do
304
305 ! Copiar incidências únicas
306 allocate(incidencias_unicas(total_incidencias, 2))
307 num_incid_unicas = total_incidencias
308 incidencias_unicas(1:num_incid_unicas, :) =
309     ↪ incidencias_malha(1:num_incid_unicas, :)
310
311 ! Escrever no arquivo de saída (saida.txt)
312 open(newunit=unit_out, file='saida.txt', status='replace', iostat=io)
313 if (io /= 0) then
314     print *, 'Erro ao criar saida.txt!'
315     stop
316 end if
317
318 write(unit_out, '(A)') 'POS-PROCESSAMENTO - ACADVIEW'
319 write(unit_out, '(A)') ' '
320 write(unit_out, '(A)') 'NNOS NELEM NLISTAS'
321 write(unit_out, '(A)') '# '
322 write(unit_out, '(I10, I10, I10)') num_unicos, num_incid_unicas, 0
323 write(unit_out, '(A)') ' '
324 write(unit_out, '(A)') 'COORDX COORDY COORDZ DESLX DESLY DESLZ'
325 write(unit_out, '(A)') '# '
326 do m = 1, num_unicos

```

```

325     write(unit_out,
      ↪ '(1x,f15.2,5x,f15.2,5x,f15.2,1x,f15.2,1x,f15.2,1x,f15.2)' ) &
326         nos_unicos(m, 2), nos_unicos(m, 3), nos_unicos(m, 4), 0.0,
      ↪ 0.0, 0.0
327 end do
328 write(unit_out, '(A)') ' '
329 write(unit_out, '(A)') 'tipo, grau, no1 no2...non'
330 write(unit_out, '(A)') '#'
331 do i = 1, num_incid_unicas
332     write(unit_out, '(I8, I6, I6, I6)' ) 1, 1, incidencias_unicas(i,
      ↪ 1), incidencias_unicas(i, 2)
333 end do
334 close(unit_out)
335 print *, 'Arquivo "saida.txt" gerado com sucesso!'
336
337 ! Escrever no arquivo malha.txt
338 open(newunit=unit_out, file='malha.txt', status='replace', iostat=io)
339 if (io /= 0) then
340     print *, 'Erro ao criar malha.txt!'
341     stop
342 end if
343
344 write(unit_out, '(A)') 'General parameters
      ↪ -----
345 write(unit_out, '(A)') ' '
346 write(unit_out, '(A16,1x,I0)') 'Number of steps: ', 10
347 write(unit_out, '(A27,1x,I0)') 'Number max of interactions: ', 100
348 write(unit_out, '(A11,es7.1)') 'Tolerance: ', 1.0E-05
349 write(unit_out, '(A)') ' '
350 write(unit_out, '(A)') 'Global parameters
      ↪ -----
351 write(unit_out, '(A)') ' '
352 write(unit_out, '(A,I0)') 'Number of nodes: ', num_unicos
353 write(unit_out, '(A,I0)') 'Number of elements: ', num_incid_unicas
354 write(unit_out, '(A,I0)') 'Number of properties: ', 1
355 write(unit_out, '(A)') ' '
356 write(unit_out, '(A)') 'Property_Id, A, E
      ↪ -----
357 write(unit_out, '(A)') ' '

```

```

358 write(unit_out, '(1x,i5,2x,F15.5,F15.5)' ) 1, real(area_calculada *
    ↪ 1.0d6, 4), 200E3
359 write(unit_out, '(A)' ) ' '
360 write(unit_out, '(A)' ) 'Node_Id, x1, x2, x3
    ↪ -----
361 write(unit_out, '(A)' ) ' '
362 do m = 1, num_unicos
363     write(unit_out, '(1x,i9,5x,f15.2,5x,f15.2,1x,f15.2)' ) m,
    ↪ nos_unicos(m, 2), nos_unicos(m, 3), nos_unicos(m, 4)
364     if ((nos_unicos(m, 3)==100).and.(nos_unicos(m, 2)==ln_no)) then
365         l_n(count_ln)=m
366         count_ln=count_ln+1
367         ln_no=ln_no+100
368     endif
369 end do
370 write(unit_out, '(A)' ) ' '
371 write(unit_out, '(A)' ) 'Element_Id, Property_Id, Incidence
    ↪ -----
372 write(unit_out, '(A)' ) ' '
373 do i = 1, num_incid_unicas
374     write(unit_out, '(i9,5x,i9,5x,i9,5x,i9)' ) i, 1,
    ↪ incidencias_unicas(i, 1), incidencias_unicas(i, 2)
375 end do
376 write(unit_out, '(A)' ) ' '
377 write(unit_out, '(A)' ) 'Boundary conditions: Nodal position
    ↪ -----
378 write(unit_out, '(A)' ) ' '
379 write(unit_out, '(A,I0)' ) 'Number of nodal position: ', num_unicos
380 write(unit_out, '(A)' ) ' '
381 write(unit_out, '(A)' ) 'Node_Id, dx1_Id, dx2_Id, dx3_Id, dx1, dx2,
    ↪ dx3'
382 write(unit_out, '(A)' ) ' '
383 do m = 1, num_unicos
384     !if (abs(nos_unicos(m, 2)) < tol) then
385     if (nos_unicos(m, 3)==0) then
386         engaste = 1
387     else
388         engaste = 0
389     endif

```

```

390     write(unit_out,
      ↪ '(i9,5x,i9,5x,i9,5x,i9,5x,f15.5,5x,f15.5,5x,f15.5)' ) m,
      ↪ engaste, engaste, 1, 0.00000, 0.00000, 0.00000
391 end do
392 write(unit_out, '(A)' ) ' '
393 write(unit_out, '(A)' ) 'Boundary conditions: Nodal load
      ↪ -----'
394 write(unit_out, '(A)' ) ' '
395 write(unit_out, '(A,I0)' ) 'Number of nodal load: ', 1
396 write(unit_out, '(A)' ) ' '
397 write(unit_out, '(A)' ) 'Node_Id, fx1_Id, fx2_Id, fx3_Id, fx1, fx2,
      ↪ fx3-----'
398 write(unit_out, '(A)' ) ' '
399 write(unit_out, '(i5,5x,i5,5x,i5,5x,i5,5x,f15.5,5x,f15.5,5x,f15.5)' )
      ↪ l_n(count_ln-1), 0, 1, 0, 0.0, -50.0, 0.0
400 close(unit_out)
401 print *, 'Arquivo "malha.txt" gerado com sucesso!'
402
403 ! Liberação de memória
404 if(allocated(coordenadas_base)) deallocate(coordenadas_base)
405 if(allocated(incidencias_base)) deallocate(incidencias_base)
406 if(allocated(poligono_pontos)) deallocate(poligono_pontos)
407 if(allocated(coordenadas_malha)) deallocate(coordenadas_malha)
408 if(allocated(incidencias_malha)) deallocate(incidencias_malha)
409 if(allocated(nos_unicos)) deallocate(nos_unicos)
410 if(allocated(mapeamento)) deallocate(mapeamento)
411 if(allocated(incidencias_unicas)) deallocate(incidencias_unicas)
412 if(allocated(nos_sobre_aresta)) deallocate(nos_sobre_aresta)
413
414 print *, 'integer, dimension(21) :: l_n=(/', l_n, '/')'
415 contains
416
417 !
      ↪ =====
418 ! SUB-ROTINA: CÁLCULO AUTOMÁTICO DE m_h E m_v
419 !
      ↪ =====
420 subroutine calcula_mh_mv(coordenadas_base, num_coord,
      ↪ incidencias_base, num_incid, m_h_out, m_v_out)

```

```

421     implicit none
422     ! Parâmetros de entrada e saída
423     integer, intent(in) :: num_coord, num_incid
424     real, intent(in) :: coordenadas_base(:, :)
425     integer, intent(in) :: incidencias_base(:, :)
426     real, intent(out) :: m_h_out, m_v_out
427
428     ! Definições de precisão e tolerância
429     integer, parameter :: DP = selected_real_kind(15, 300)
430     real(DP), parameter :: TOL_MH = 1.0D-7
431
432     ! Variáveis locais
433     integer :: i
434     integer, allocatable :: id(:), ei(:), ej(:)
435     real(DP), allocatable :: x(:), y(:)
436     real(DP) :: xmin, xmax, ymin, ymax, wcell, hcell
437     real(DP) :: Ih, Eh, Iv, Ev
438
439     ! Extrair dados da célula para os formatos esperados pelas
440     ↪ sub-rotinas
441     allocate(id(num_coord), x(num_coord), y(num_coord))
442     allocate(ei(num_incid), ej(num_incid))
443
444     do i = 1, num_coord
445         id(i) = nint(coordenadas_base(i, 1))
446         x(i) = dble(coordenadas_base(i, 2))
447         y(i) = dble(coordenadas_base(i, 3))
448     end do
449
450     do i = 1, num_incid
451         ei(i) = incidencias_base(i, 1)
452         ej(i) = incidencias_base(i, 2)
453     end do
454
455     ! BBox da célula
456     call bbox_mh(x, y, num_coord, xmin, xmax, ymin, ymax)
457     wcell = xmax - xmin
458     hcell = ymax - ymin
459     if (wcell <= 0.0D0 .or. hcell <= 0.0D0) then

```

```

459         print *, 'Erro em calcula_mh_mv: BBox degenerado.'
460         stop
461     end if
462
463     ! Somas de comprimentos projetados
464     call soma_projecoeh_mh(x, y, id, num_coord, ei, ej, num_incid,
465         ↪ xmin, xmax, ymin, ymax, TOL_MH, Ih, Eh, Iv, Ev)
466
467     ! Cálculo de m_h e m_v
468     m_h_out = (Ih + 0.5D0*Eh) / wcell
469     m_v_out = (Iv + 0.5D0*Ev) / hcell
470
471     ! Liberação de memória
472     deallocate(id, x, y, ei, ej)
473
474 end subroutine calcula_mh_mv
475
476 subroutine bbox_mh(x, y, n, xmin, xmax, ymin, ymax)
477     integer, parameter :: DP = selected_real_kind(15, 300)
478     real(DP), intent(in) :: x(:), y(:)
479     integer, intent(in) :: n
480     real(DP), intent(out) :: xmin, xmax, ymin, ymax
481     integer :: i
482     xmin = x(1); xmax = x(1)
483     ymin = y(1); ymax = y(1)
484     do i = 2, n
485         if (x(i) < xmin) xmin = x(i)
486         if (x(i) > xmax) xmax = x(i)
487         if (y(i) < ymin) ymin = y(i)
488         if (y(i) > ymax) ymax = y(i)
489     end do
490
491 end subroutine bbox_mh
492
493 logical function on_edge_h_mh(y1, y2, ymin, ymax, tol)
494     integer, parameter :: DP = selected_real_kind(15, 300)
495     real(DP), intent(in) :: y1, y2, ymin, ymax, tol
496     on_edge_h_mh = (abs(y1 - ymin) <= tol .and. abs(y2 - ymin) <=
497         ↪ tol) .or. &
498         (abs(y1 - ymax) <= tol .and. abs(y2 - ymax) <= tol)

```

```

496     end function on_edge_h_mh
497
498     logical function on_edge_v_mh(x1, x2, xmin, xmax, tol)
499         integer, parameter :: DP = selected_real_kind(15, 300)
500         real(DP), intent(in) :: x1, x2, xmin, xmax, tol
501         on_edge_v_mh = (abs(x1 - xmin) <= tol .and. abs(x2 - xmin) <=
502             ↪ tol) .or. &
503             (abs(x1 - xmax) <= tol .and. abs(x2 - xmax) <= tol)
504     end function on_edge_v_mh
505
506     subroutine soma_projecoeh_mh(x, y, id, nn, ei, ej, ne, xmin, xmax,
507     ↪ ymin, ymax, tol, Ih, Eh, Iv, Ev)
508         integer, parameter :: DP = selected_real_kind(15, 300)
509         real(DP), intent(in) :: x(:), y(:)
510         integer, intent(in) :: id(:), nn, ei(:), ej(:), ne
511         real(DP), intent(in) :: xmin, xmax, ymin, ymax, tol
512         real(DP), intent(out):: Ih, Eh, Iv, Ev
513         integer :: e, p1, p2, n1, n2
514         real(DP) :: x1, y1, x2, y2, dx, dy
515
516         Ih = 0.0D0; Eh = 0.0D0
517         Iv = 0.0D0; Ev = 0.0D0
518
519     do e = 1, ne
520         n1 = ei(e); n2 = ej(e)
521         p1 = find_node_pos_mh(id, nn, n1)
522         p2 = find_node_pos_mh(id, nn, n2)
523         if (p1 <= 0 .or. p2 <= 0) cycle
524
525         x1 = x(p1); y1 = y(p1)
526         x2 = x(p2); y2 = y(p2)
527
528         dx = abs(x2 - x1)
529         dy = abs(y2 - y1)
530
531         ! Direcao H: soma |dx|; interfaces: barras inteiras nas bordas
532         ↪ y=ymin ou y=ymax
533         if (on_edge_h_mh(y1, y2, ymin, ymax, tol)) then
534             Eh = Eh + dx

```

```

532     else
533         Ih = Ih + dx
534     end if
535
536     ! Direcao V: soma |dy|; interfaces: barras inteiras nas bordas
537     ↪ x=xmin ou x=xmax
538     if (on_edge_v_mh(x1, x2, xmin, xmax, tol)) then
539         Ev = Ev + dy
540     else
541         Iv = Iv + dy
542     end if
543 end do
544 end subroutine soma_projecoes_mh
545
546 integer function find_node_pos_mh(id, n, key)
547     integer, intent(in) :: id(:), n, key
548     integer :: i
549     find_node_pos_mh = -1
550     do i = 1, n
551         if (id(i) == key) then
552             find_node_pos_mh = i
553             return
554         end if
555     end do
556 end function find_node_pos_mh
557
558 !
559 ↪ =====
560 ! SUB-ROTINA: CÁLCULO DA ÁREA EQUIVALENTE
561 !
562 ↪ =====
563 subroutine calcula_area_equivalente(area_saida, L_mm_in, eps_x_in,
564     ↪ gamma_xy_in, coef_h_in, coef_v_in, &
565     nos_celula, num_nos,
566     ↪ incidencias_celula,
567     ↪ num_incidencias)
568
569     implicit none
570     real(8), intent(out) :: area_saida

```

```

564     real(8), intent(in) :: L_mm_in, eps_x_in, gamma_xy_in, coef_h_in,
      ↪     coef_v_in
565     real, intent(in) :: nos_celula(:, :)
566     integer, intent(in) :: num_nos
567     integer, intent(in) :: incidencias_celula(:, :)
568     integer, intent(in) :: num_incidencias
569     real(8) :: L_m, t, E, nu_plate, eps_y, U_plate_energy,
      ↪     U_truss_energy, A_truss
570     real(8) :: D_plate(3,3), sigma_plate(3), eps_plate(3)
571     real(8), allocatable :: x_nodes_truss(:), y_nodes_truss(:)
572     integer :: i, j, k
573     real(8) :: L_bar, cos_theta, sin_theta, x_min, x_max, y_min,
      ↪     y_max
574     logical :: is_vertical_edge, is_horizontal_edge
575     real(8) :: energy_factor, delta_u_x, delta_u_y, delta_u_axial,
      ↪     eng_strain

576
577     print *, ' '
578     print *, '|--- EXECUTANDO CALCULO DA AREA EQUIVALENTE ---|'
579     E = 200.0d9
580     nu_plate = 0.3d0
581     t = 0.001d0
582     L_m = L_mm_in / 1000.0d0
583     eps_y = -nu_plate * eps_x_in
584
585     allocate(x_nodes_truss(num_nos), y_nodes_truss(num_nos))
586     do i = 1, num_nos
587         x_nodes_truss(i) = dble(nos_celula(i, 2)) / 1000.0d0
588         y_nodes_truss(i) = dble(nos_celula(i, 3)) / 1000.0d0
589     end do

590
591     D_plate = 0.0d0
592     D_plate(1,1) = (1.0d0 - nu_plate) / ((1.0d0 + nu_plate) * (1.0d0
      ↪     - 2.0d0 * nu_plate))
593     D_plate(1,2) = nu_plate / ((1.0d0 + nu_plate) * (1.0d0 - 2.0d0 *
      ↪     nu_plate))
594     D_plate(2,1) = D_plate(1,2)
595     D_plate(2,2) = D_plate(1,1)

```

```

596     D_plate(3,3) = 0.5d0 * (1.0d0 - 2.0d0 * nu_plate) / (1.0d0 +
      ↪ nu_plate)
597     D_plate = D_plate * E
598
599     eps_plate(1) = eps_x_in
600     eps_plate(2) = eps_y
601     eps_plate(3) = gamma_xy_in
602     sigma_plate = matmul(D_plate, eps_plate)
603     U_plate_energy = 0.5d0 * (t * L_m * L_m) *
      ↪ dot_product(sigma_plate, eps_plate)
604
605     x_min = minval(x_nodes_truss); x_max = maxval(x_nodes_truss)
606     y_min = minval(y_nodes_truss); y_max = maxval(y_nodes_truss)
607
608     U_truss_energy = 0.0d0
609     do k = 1, num_incidencias
610         i = incidencias_celula(k,1); j = incidencias_celula(k,2)
611         L_bar = sqrt((x_nodes_truss(j) - x_nodes_truss(i))**2 +
      ↪ (y_nodes_truss(j) - y_nodes_truss(i))**2)
612         if (L_bar < 1.0d-9) cycle
613         cos_theta = (x_nodes_truss(j) - x_nodes_truss(i)) / L_bar
614         sin_theta = (y_nodes_truss(j) - y_nodes_truss(i)) / L_bar
615
616         delta_u_x = (eps_x_in * (x_nodes_truss(j) - x_min) + 0.5d0 *
      ↪ gamma_xy_in * (y_nodes_truss(j) - y_min)) - &
617             (eps_x_in * (x_nodes_truss(i) - x_min) + 0.5d0 *
      ↪ gamma_xy_in * (y_nodes_truss(i) - y_min))
618         delta_u_y = (eps_y * (y_nodes_truss(j) - y_min) + 0.5d0 *
      ↪ gamma_xy_in * (x_nodes_truss(j) - x_min)) - &
619             (eps_y * (y_nodes_truss(i) - y_min) + 0.5d0 *
      ↪ gamma_xy_in * (x_nodes_truss(i) - x_min))
620
621         delta_u_axial = delta_u_x * cos_theta + delta_u_y * sin_theta
622         eng_strain = delta_u_axial / L_bar
623
624         is_vertical_edge = (abs(x_nodes_truss(i) - x_min) < 1.0d-6
      ↪ .and. abs(x_nodes_truss(j) - x_min) < 1.0d-6) .or. &

```

```

625         (abs(x_nodes_truss(i) - x_max) < 1.0d-6
           ↪ .and. abs(x_nodes_truss(j) - x_max) <
           ↪ 1.0d-6)
626     is_horizontal_edge = (abs(y_nodes_truss(i) - y_min) < 1.0d-6
           ↪ .and. abs(y_nodes_truss(j) - y_min) < 1.0d-6) .or. &
627         (abs(y_nodes_truss(i) - y_max) < 1.0d-6
           ↪ .and. abs(y_nodes_truss(j) - y_max)
           ↪ < 1.0d-6)
628
629     if (is_vertical_edge) then
630         energy_factor = 0.5d0 * (0.5d0 + coef_v_in)
631     else if (is_horizontal_edge) then
632         energy_factor = 0.5d0 * (0.5d0 + coef_h_in)
633     else
634         energy_factor = 0.5d0
635     end if
636
637     U_truss_energy = U_truss_energy + energy_factor * E * L_bar *
           ↪ eng_strain**2
638 end do
639
640 if (abs(U_truss_energy) < 1.0d-12) then
641     print *, 'Erro: Energia da treliça é zero. Impossível
           ↪ calcular a área.'
642     stop
643 end if
644 A_truss = U_plate_energy / U_truss_energy
645
646 print *, ' '
647 print *, '--- RESULTADOS DO CALCULO DA AREA ---'
648 print *, 'Deformacao calculada eps_y: ', eps_y
649 print *, 'Energia da chapa (J): ', U_plate_energy
650 print *, 'Energia da trelica por unidade de area (J/m^2): ',
           ↪ U_truss_energy
651 print *, 'Área da secao transversal das barras (mm^2): ', A_truss
           ↪ * 1.0d6
652 print *, 'Coeficientes da celula: ', m_h, '/', m_v
653 print *, 'Coeficientes: ', coef_h, ' / ', coef_v
654

```

```

655
656     area_saida = A_truss
657     deallocate(x_nodes_truss, y_nodes_truss)
658 end subroutine calcula_area_equivalente
659
660
661 logical function point_in_polygon(x, y, poly, n) result(inside)
662     real, intent(in) :: x, y
663     real, intent(in) :: poly(:, :)
664     integer, intent(in) :: n
665     integer :: i, j
666     real :: xi, yi, xj, yj, denom, tol_local
667     logical :: intersect
668
669     tol_local = tol
670     inside = .false.
671     do i = 1, n
672         if (abs(x - poly(i, 2)) < tol_local .and. abs(y - poly(i, 3))
673             ↪ < tol_local) then
674             inside = .true.
675             return
676         end if
677     end do
678
679     j = n
680     do i = 1, n
681         xi = poly(i, 2); yi = poly(i, 3)
682         xj = poly(j, 2); yj = poly(j, 3)
683         denom = (xj - xi) * (y - yi) - (yj - yi) * (x - xi)
684         if (abs(denom) < tol_local) then
685             if ((min(xi, xj) - tol_local <= x .and. x <= max(xi, xj)
686                 ↪ + tol_local) .and. &
687                 (min(yi, yj) - tol_local <= y .and. y <= max(yi, yj)
688                 ↪ + tol_local)) then
689                 inside = .true.
690                 return
691             end if
692         end if
693     end do
694     j = i

```

```

691     end do
692
693     j = n
694     do i = 1, n
695         xi = poly(i, 2); yi = poly(i, 3)
696         xj = poly(j, 2); yj = poly(j, 3)
697         intersect = ((yi > y) .neqv. (yj > y)) .and. &
698                     (x < (xj - xi) * (y - yi) / (yj - yi) + xi)
699         if (intersect) inside = .not. inside
700         j = i
701     end do
702 end function point_in_polygon
703
704 logical function point_on_edge(x, y, x1, y1, x2, y2) result(on_edge)
705     real, intent(in) :: x, y, x1, y1, x2, y2
706     real :: denom, tol_local
707     tol_local = tol
708     on_edge = .false.
709     denom = (x2 - x1) * (y - y1) - (y2 - y1) * (x - x1)
710     if (abs(denom) < tol_local) then
711         if ((min(x1, x2) - tol_local <= x .and. x <= max(x1, x2) +
712             ↪ tol_local) .and. &
713             (min(y1, y2) - tol_local <= y .and. y <= max(y1, y2) +
714             ↪ tol_local)) then
715             on_edge = .true.
716         end if
717     end if
718 end function point_on_edge
719
720 subroutine sort_nodes(arr, n)
721     implicit none
722     real, intent(inout) :: arr(:, :)
723     integer, intent(in) :: n
724     real :: temp(size(arr, 2))
725     integer :: i, j
726     if (n <= 1) return
727     do i = 1, n - 1
728         do j = i + 1, n
729             if (arr(i, 8) > arr(j, 8)) then

```

```

728             temp = arr(i, :)
729             arr(i, :) = arr(j, :)
730             arr(j, :) = temp
731         end if
732     end do
733 end do
734 end subroutine sort_nodes
735
736 subroutine resize_real_array(arr, new_size, cols)
737     implicit none
738     real, allocatable, intent(inout) :: arr(:, :)
739     integer, intent(in) :: new_size, cols
740     real, allocatable :: temp(:, :)
741     integer :: old_size
742     if (.not. allocated(arr)) then
743         allocate(arr(new_size, cols))
744         return
745     end if
746     old_size = size(arr, 1)
747     allocate(temp(new_size, cols))
748     temp = 0.0
749     if (old_size > 0) then
750         temp(1:min(old_size, new_size), :) = arr(1:min(old_size,
751             ↪ new_size), :)
752     end if
753     call move_alloc(temp, arr)
754 end subroutine resize_real_array
755
756 subroutine resize_int_array(arr, new_size, cols)
757     implicit none
758     integer, allocatable, intent(inout) :: arr(:, :)
759     integer, intent(in) :: new_size, cols
760     integer, allocatable :: temp(:, :)
761     integer :: old_size
762     if (.not. allocated(arr)) then
763         allocate(arr(new_size, cols))
764         return
765     end if
766     old_size = size(arr, 1)

```

```
766     allocate(temp(new_size, cols))
767     temp = 0
768     if (old_size > 0) then
769         temp(1:min(old_size, new_size), :) = arr(1:min(old_size,
770             ↪ new_size), :)
771     end if
772     call move_alloc(temp, arr)
773 end subroutine resize_int_array
774 end program Gera_malha
```