

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Diego Cuitiño Manso

Otimização topológica para problemas de condução de calor usando elementos finitos poligonais

Trabalho de Conclusão de Curso

Departamento de Engenharia Mecânica
Programa de Graduação em Engenharia Mecânica

Anderson Pereira, D.Sc.

Orientador
Departamento de Engenharia Mecânica – PUC-Rio

Rio de Janeiro, 14 de dezembro, 2016

You don't have to be great to start,
but you have to start to be great. – Zig Zagler

Agradecimentos

Primeiramente agradeço aos meus pais por terem sempre me encorajado a tomar minhas próprias decisões, escolher meus próprios caminhos, me incentivando ao longo de toda a jornada. Eles me permitiram viver, errar e aprender por conta própria, pois esses são os melhores aprendizados. Agradeço por terem aceitado minha escolha de faculdade e investido em minha formação e educação ao longo de todos esses anos. Posso não deixar isso evidente a todos os momentos, porém sou muito grato, pois sei os sacrifícios que são tomados para permitir isso.

Agradeço ao meu avô por todas as nossas discussões e conversas sobre engenharia, presentes em minha vida desde que eu me entendo por gente, e um dos motivos pelos quais segui em frente nessa escolha de carreira, apesar da dificuldade e de meus inúmeros tropeços.

Agradeço ao professor Ivan F. M. Menezes que me recebeu sempre com muita simpatia me ajudando na escolha de um tema e de um orientador, apesar da minha procura ter sido bastante tardia.

Ao Dr. Anderson Pereira que se tornou meu orientador, idealizou o que viria a ser esse trabalho, e sempre se mostrou disponível para sanar quaisquer que fossem minhas dúvidas. Obrigado pela paciência, por sempre procurar me estimular e principalmente por me mostrar que seria possível a realização desse trabalho, apesar do curto espaço de tempo e de todas as minhas outras obrigações.

Aos meus irmãos não biológicos, amigos que a vida me deu dentro e fora da faculdade e que carrego no peito, aos amigos que não mais seguem nesse plano por alguma fatalidade da vida, vocês foram os pilares nos quais me apoiei quando mais precisava, sempre me encorajaram a seguir em frente e nunca desistir de meus objetivos, me ajudando a me reerguer quando caía e vibrando com minhas conquistas.

Resumo

Esse trabalho visa a realização de simulações em algoritmos de MATLAB para a otimização topológica de problemas de condução de calor, tendo como meta replicar as tradicionais topologias ramificadas conhecidas na literatura por serem os caminhos condutivos ótimos na condução de calor 2D. Em primeiro momento se discute o algoritmo educacional compacto de 99 linhas de Ole Sigmund, conhecido como ponto de acesso à otimização topológica estrutural em malha estrutura, e em seguida se adapta o código para a otimização topológica da condução de calor em meio isotrópico e homogêneo. Para ilustrar o funcionamento do algoritmo, diferentes otimizações foram simuladas ao se variar os parâmetros passados na chamada da função, variando do número de elementos ao tamanho do filtro. Aplicações gerais e as limitações desse código compacto são discutidas para então seguir com a extensão do estudo em malhas não estruturadas arbitrárias com elementos finitos poligonais. Foram adotadas mudanças ao já conhecido e validado código `PolyTop` de Talischi et al., rotina em código MATLAB para otimização topológica estrutural que inclui uma rotina de elementos finitos baseada em elementos poligonais isoparamétricos, para também adequá-la a condução de calor. A partir da geração de um novo domínio de projeto, foi simulado diferentes condições iniciais para se comparar com os resultados obtidos com o código adaptado de 99 linhas de Sigmund e com a literatura.

Palavras-chave: Otimização topológica, condução de calor, código MATLAB.

Topology optimization for heat conduction problems using polygonal finite elements

Abstract

This work is focused on topology optimization of heat transfer in two dimensional problems (2D). Its aim was to perform simulations on different MATLAB routines for the topological optimization of pure heat conduction, trying to emulate the traditional “tree-like” optimized conductive pathway commonly seen in literature. Firstly, we discuss the 99-line educational algorithm created by Ole Sigmund and known till this day as the access point for the learning and implementing of topology optimization for compliance minimization of statically loaded structures. Afterwards, this script was altered to adapt it to topology optimization for heat conduction in isotropic and homogeneous domains. To further illustrate the algorithm’s functioning, different optimizations were conducted by altering specific parameters, such as the total number of mesh elements and the filter’s size. General applications and limitations of this method were discussed. To conclude the work, `PolyTop`, an efficient MATLAB code for structural topology optimization using unstructured meshes with polygonal finite elements were introduced and discussed, followed by its alteration for single physics topology optimization of heat transfer. Also, as part of the validation process, different initial conditions were simulated and resulting optimized branched structures compared with topologies obtained with Sigmund's adapted 99-line.

Keywords: Topology optimization, heat conduction, MATLAB code.

Sumário

1. Introdução.....	11
1.1. Objetivos.....	11
1.2. Motivação	11
1.3. Revisão Bibliográfica	13
1.3.1. Método dos Elementos Finitos.....	13
1.3.2. Otimização	14
1.3.3. Otimização Estrutural e Topológica.....	15
1.3.4. Transferência de Calor.....	17
1.4. Organização do Trabalho.....	19
2. Método dos Elementos Finitos	21
2.1. Introdução	21
2.2. Conceitos Fundamentais.....	22
2.3. Graus de Liberdade	23
2.4. Formulação Isoparamétrica.....	25
2.4.1. Sistema de Referência Global e Local.....	25
2.4.2. Elementos Finitos Poligonais.....	26
2.4.3. Funções de Forma	27
2.4.4. Cálculo da Matriz de Rigidez.....	30
3. Otimização Topológica	32
3.1. Introdução	32
3.2. A 99 Line Topology Optimization Code.....	33
3.2.1. Considerações Iniciais	33
3.2.2. Formulação do Problema	35
3.2.3. Implementação do Código	37
3.2.4. Adaptação do Código	40
4. Equacionamento Para o Problema de Condução de Calor	43
4.1. Introdução	43
4.2. Equações Governantes do Problema	45
4.3. Formulação do Problema.....	49
5. Resultados.....	53
6. Conclusão	63
Referências	65

Lista de Figuras

FIGURA 1.1 - MODOS DE TRANSFERÊNCIA DE CALOR	17
FIGURA 1.2 - TRANSFERÊNCIA DE CALOR UNIDIMENSIONAL POR CONDUÇÃO	19
FIGURA 2.1 - MALHA DE ELEMENTOS FINITOS PARA PROBLEMAS NO PLANO.	22
FIGURA 2.2 - DIFERENTES TIPOS DE ELEMENTOS FINITOS.....	23
FIGURA 2.3 - GRAUS DE LIBERDADE DE UM PONTO E DE UM CORPO RÍGIDO.....	24
FIGURA 2.4 - TRANSFORMAÇÃO ENTRE OS SISTEMAS DE REFERÊNCIA GLOBAL E LOCAL.....	25
FIGURA 2.5 - TRANSFORMAÇÃO ENTRE OS SISTEMAS DE REFERÊNCIA UTILIZANDO FUNÇÕES DE FORMA	26
FIGURA 2.6 – TRIANGULAÇÃO DE DELAUNAY EM CÉLULA DE VORONOI	27
FIGURA 2.7 – ELEMENTO BIDIMENSIONAL QUADRANGULAR LINEAR	28
FIGURA 2.8 – DEMONSTRAÇÃO VISUAL DO FUNCIONAMENTO DAS FUNÇÕES DE FORMA DE ELEMENTOS QUADRANGULARES LINEARES	29
FIGURA 2.9 – ÁREAS TRIANGULARES UTILIZADAS NA DEFINIÇÃO DE α_i	30
FIGURA 3.1 – CATEGORIAS DE OTIMIZAÇÃO ESTRUTURAL	32
FIGURA 3.2 – CORPO BIDIMENSIONAL COM DOMÍNIO Ω E CONTORNO Γ	34
FIGURA 3.3 – DOMÍNIO APROXIMADO POR DIFERENTES MALHAS.....	34
FIGURA 3.4 – MODELO DE ELEMENTOS FINITOS E MODELO DE ELEMENTOS FINITOS REFINADO .	35
FIGURA 3.5 – NUMERAÇÃO DOS ELEMENTOS DA MALHA.....	36
FIGURA 3.6 – NUMERAÇÃO DOS NÓS DA MALHA.....	36
FIGURA 3.7 – OTIMIZAÇÃO TOPOLÓGICA DA VIGA MBB.....	38
FIGURA 3.8 – DISTRIBUIÇÃO DE MATERIAL NO INTERIOR DE UM DOMÍNIO E AO LONGO DE ALGUMAS ITERAÇÕES	39
FIGURA 3.9 – OTIMIZAÇÃO TOPOLÓGICA DA CONDUÇÃO DE CALOR	41
FIGURA 4.1 – O EFEITO DO TABULEIRO DE XADREZ	44
FIGURA 4.2 – ELEMENTO DIFERENCIAL COM FLUXO DE CALOR ATRAVESSANDO SEU CONTORNO	45
FIGURA 4.3 – MODELO DE ELEMENTOS FINITOS DE DUAS DIMENSÕES NA FORMA FRACA PARA O PROBLEMA DE CONDUÇÃO DE CALOR.....	48
FIGURA 4.4 – DOMÍNIO DE PROJETO 2D PARA A OTIMIZAÇÃO TOPOLÓGICA DA CONDUÇÃO DE CALOR PURA.....	50
FIGURA 5.1 – SIMULAÇÃO DO CÓDIGO DE 91 LINHAS PARA OT DE PROBLEMAS DE CONDUÇÃO DE CALOR. CHAMADA DA FUNÇÃO: TOPH(40,40,0.4,3.0,1.2)	53
FIGURA 5.2 – SIMULAÇÃO DO CÓDIGO DE 91 LINHAS PARA OT DE PROBLEMAS DE CONDUÇÃO DE CALOR. CHAMADA DA FUNÇÃO: TOPH(40,40,0.5,3.0,1.2).....	53
FIGURA 5.3 – SIMULAÇÃO DO CÓDIGO DE 91 LINHAS PARA OT DE PROBLEMAS DE CONDUÇÃO DE CALOR. CHAMADA DA FUNÇÃO: TOPH(80,80,0.4,3.0,1.2).....	54
FIGURA 5.4 – SIMULAÇÃO DO CÓDIGO DE 91 LINHAS PARA OT DE PROBLEMAS DE CONDUÇÃO DE CALOR. CHAMADA DA FUNÇÃO: TOPH(80,80,0.5,3.0,1.2).....	55
FIGURA 5.5 – SIMULAÇÃO DO CÓDIGO DE 91 LINHAS PARA OT DE PROBLEMAS DE CONDUÇÃO DE CALOR. CHAMADA DA FUNÇÃO: TOPH(100,100,0.4,3.0,1.2)	55
FIGURA 5.6 – SIMULAÇÃO DO CÓDIGO DE 91 LINHAS PARA OT DE PROBLEMAS DE CONDUÇÃO DE CALOR. CHAMADA DA FUNÇÃO: TOPH(100,100,0.5,3.0,1.2)	56

FIGURA 5.7 – SQUAREDOMAIN COM MALHA NÃO ESTRUTURADA DE 10 ELEMENTOS POLIGONAIS (À ESQUERDA) E 100 ELEMENTOS POLIGONAIS (À DIREITA).....	57
FIGURA 5.8 – POLYSCRIPTHEAT(@SQUAREDOMAIN,1600,100).	58
FIGURA 5.9 – POLYSCRIPTHEAT(@SQUAREDOMAIN,6400,100).	59
FIGURA 5.10 – POLYSCRIPTHEAT(@SQUAREDOMAIN,10000,100).....	59
FIGURA 5.11 – COMPARATIVO DA OT DE MALHA NÃO ESTRUTURADA PARA CONDUÇÃO DE CALOR COM DIFERENTES NUMEROS DE ELEMENTOS.....	60
FIGURA 5.12 – INFLUÊNCIA DO RAI DO FILTRO NA OT DE MALHA NÃO ESTRUTURADA PARA CONDUÇÃO DE CALOR.	61
FIGURA 5.13 – INFLUÊNCIA DO VARIAÇÃO DA FRAÇÃO VOLUMÉTRICA NO LAYOUT OTIMIDADO DA CONDUÇÃO DE CALOR.....	62

Lista de Símbolos

Acrônimos

MEF	MÉTODO DOS ELEMENTOS FINITOS
FEM	FINITE ELEMENT METHOD
MEC	MÉTODO DOS ELEMENTOS DE CONTORNO
EF	ELEMENTOS FINITOS
OT	OTIMIZAÇÃO TOPOLÓGICA
TOP	TOPOLOGY OPTIMIZATION
OC	CRITÉRIO DE ÓTIMO (OPTIMALITY CRITERIA)
DOF	DEGREE OF FREEDOM
GDL	GRAU DE LIBERDADE
2D	BI-DIMENSIONAL
3D	TRIDIMENSIONAL
MBB	MESSERSCHMITT-BÖLKOW-BLOHM
ESO	EVOLUTIONARY STRUCTURAL OPTIMIZATION
SIMP	SOLID ISOTROPIC MATERIAL WITH PENALIZATION

Romanos de caixa-alta

B	MATRIX DE DEFORMAÇÃO
F	VETOR DE FORÇAS GLOBAIS
K	MATRIX DE RIGIDEZ GLOBAL
N	MATRIZ DE FUNÇÕES DE FORMA
N ..	NÚMERO DE ELEMENTOS USADO PARA DISCRETIZAR O DOMÍNIO DE PROJETO ($NELX * NELY$)
N_i ..	FUNÇÃO DE FORMA DE UM DADO ELEMENTO
U	VETOR DOS DESLOCAMENTOS GLOBAIS (GLOBAL DISPLACEMENT VECTOR)
D	MATRIX DE ELASTICIDADE
E	MÓDULO DE YOUNG
J	JACOBIANO
$V(x)$..	VOLUME MATERIAL
V_0 ..	VOLUME DO DOMÍNIO DE PROJETO
$V(p_i)$..	CÉLULA DE VORONOI

Romano de caixa-baixa

u_e	VETOR DE DESLOCAMENTO DO ELEMENTO
x	VETOR DE VARIÁVEIS DE PROJETO
x_{min} ..	VETOR COM AS DENSIDADES RELATIVAS MÍNIMAS
x_e	VARIÁVEIS DE PROJETO
k_e	MATRIZ DE RIGIDEZ DO ELEMENTO
k_{max} ..	MATERIAL DE ALTA CONDUTIVIDADE TÉRMICA
k_{min} ..	MATERIAL DE BAIXA CONDUTIVIDADE TÉRMICA

p	PENALIZATION POWER
c	COMPLIANCE DO PROJETO
R_{min}	RAIO DO FILTRO
f	FRAÇÃO VOLUMÉTRICA PRESCRITA (VOLFRAC)

Grego de caixa-alta

Ω	DOMÍNIO DE PROJETO (DESIGN DOMAIN)
$d\Omega$	DOMÍNIO DISCRETIZADO
Γ	CONTORNO DO DOMÍNIO

Grego de caixa-baixa

λ	MULTIPLICADOR DE LAGRANGE
σ	VETOR DE TENSÃO
ε	VETOR DE DEFORMAÇÃO
ν	COEFICIENTE DE POISSON
ρ	DENSIDADE
ρ_{min}	LIMITE INFERIOR DA DENSIDADE

Outros símbolos e termos

R^2	CONJUNTO DE PONTOS QUE DEFINEM O ESPAÇO PLANO
R^3	CONJUNTO DE PONTOS QUE DEFINEM O ESPAÇO TRIDIMENSIONAL
PAPER	PEQUENO ARTIGO CIENTÍFICO A RESPEITO DE UM TEMA PRÉ-DETERMINADO
COMPLIANCE	FLEXIBILIDADE
99 LINES	CÓDIGO DE OTIMIZAÇÃO TOPOLÓGICA CRIADO POR SIGMUND (1999)
POLYTOP & POLYSCRIPT	CÓDIGO DE OTIMIZAÇÃO TOPOLÓGICA CRIADO POR TALISCHI ET AL. (2012)

1. Introdução

1.1. Objetivos

Este trabalho se propõe inicialmente em analisar e compreender a rotina compacta e genérica de implementação em 99 linhas de código no software MATLAB da otimização topológica para minimização de flexibilidade (*compliance*) de estruturas estaticamente carregadas, proposta por Sigmund [1]. Posteriormente, dados os conhecimentos introdutórios adquiridos em Elementos Finitos e Otimização Topológica, pretende-se modificar a existente rotina com as sugestões discutidas por Bendsøe e Sigmund [2] para a adequação do código para a resolução de problemas de condução de calor, respeitadas as condições iniciais e de contorno empregadas no modelo. Por último pretende-se adaptar o algoritmo de otimização topológica de elementos finitos poligonais isoparamétricos desenvolvidos por Talischi et al. [3] para modelar o problema de condução de calor discutido acima e procurar assim, obter resultados igualmente satisfatórios.

1.2. Motivação

O Merriam-Webster, a referência Norte Americana e mundial de publicações de referência de cunho linguístico, define a engenharia como a arte da aplicação da ciência e da matemática através da qual as propriedades da matéria e as fontes de energia na natureza se tornam úteis para as pessoas. Em contrapartida, a *Accreditation Board for Engineering and Technology, Inc* (ABET), define a mesma como a profissão em que o conhecimento das ciências matemáticas e naturais, adquiridas pelo estudo, experiência e prática, são aplicados com juízo para desenvolver meios de utilizar, economicamente, os materiais e forças da natureza em prol da humanidade [4]. Independente de qual seja seu entendimento e definição da engenharia, é possível se argumentar, que esta prática não exista fora do domínio dos interesses da sociedade,

caminhando paralelamente com suas necessidades e ocasionando um impacto inerente e inevitável nela. Sendo assim, ao contrário das ciências, motivadas em grande parte pela curiosidade, a engenharia visa combinar a sede pelo entendimento e o conhecimento em sua pureza, com as necessidades entendidas da sociedade, encontrando soluções práticas para os problemas do mundo real à medida que estes são criados, e demandando uma contínua interação com a sociedade [5]. É inegável que existe uma vasta gama de engenharias e especialidades, porém, todas compartilham do mesmo objetivo que é a soluções dos problemas relacionados a uma determinada área da vida humana.

Dadas as crescentes necessidades de nosso meio, a complexidade desses problemas e as limitações da mente humana, engenheiros e matemáticos passaram a dividir sistemas em seus componentes individuais para facilitar a sua compreensão. Posteriormente, reconstruía-se o sistema original a partir desses elementos e se estudava o comportamento do todo. Problemas discretos se originaram, portanto, de uma divisão finita e bem definida dos componentes, enquanto os problemas cujas subdivisões continuavam de forma indefinida, levaram a implicação de equações diferenciais, ou seja, um número infinito de elementos, um sistema dito contínuo. No âmbito desses problemas realísticos de natureza contínua, cujas soluções eram obtidas apenas por aproximações que no limite se possivelmente se assemelham a real solução contínua, que foram se desenvolvendo métodos de “discretização” genéricos que eventualmente levaram às aproximações por diferenças finitas e ao então conhecido e muito utilizado, Método dos Elementos Finitos (MEF) [6].

Desde a sua criação, o MEF possibilitou e ainda possibilita à pesquisadores e engenheiros resolver por aproximações numéricas as equações diferenciais parciais que regem os problemas típicos de engenharia de análise de tensões, troca de calor, escoamento de fluidos [7]. Aliadas aos avanços das tecnologias de computação e dos processos de simulação, é possível se prever de forma bastante satisfatória o comportamento estrutural, mecânico,

térmico, elétrico e químico de análises de design e performance indispensáveis para nossos avanços tecnológicos.

Tratando-se da necessidade do uso eficiente de materiais dados seus custos, econômicos e sociais, “performáticos”, devido ao impacto ambiental da aquisição dessa matéria prima ou até mesmo devido a sua escassez, a otimização geométrica e topológica passou a ser cada vez mais estudada e priorizada. Tendo-se percebido o alto impacto em relação a performance de estruturas que é possível se obter com a otimização topológica, método que tem como base o MEF, foi observado um crescente investimento monetário e intelectual em pesquisas dessas áreas, cujas aplicações vão desde a distribuição de material para otimização de estruturas com aplicações mecânicas, largamente utilizada na engenharia civil, as aplicações de condutividade térmica em sólidos, de importância prática e óbvia em difusores de calor, aletas de refrigeração, canais de alta condutividade de micro componentes eletrônicos [8,9,10,11].

1.3. Revisão Bibliográfica

1.3.1 Método dos Elementos Finitos

Para a resolução de problemas complexos de relações entre tensões e deformações de um corpo no qual uma força é aplicada, torna-se necessário o uso de métodos numéricos, pois a resolução analítica já não é mais trivial. Estabelecendo-se um sistema de equações diferenciais válidas em uma determinada região, ou domínio, segue-se para a imposição de condições iniciais, e de contorno e por fim lança-se mão do método numérico escolhido. Obtém-se então, soluções aproximadas da equação diferencial que se propõe a resolver através da “discretização” do problema contínuo e através do uso de aproximações.

O MEF foi inicialmente desenvolvido para o uso da indústria aeroespacial nos anos de 1950 e envolveu grandes corporações como a Boeing e a Rolls Royce [7]. Sua primeira citação

na literatura ocorreu em 1956 com a publicação do paper acadêmico: *Stiffness and deflection analysis of complex structures*, tendo sua criação atribuída ao então professor de Berkley, Ray William Clough, por ter sido o primeiro a utilizar o termo *Finite Elements* [13]. Nos anos subsequentes o MEF foi ganhando a atenção da comunidade acadêmica, matemáticos e grandes corporações que reconheceram seus notáveis benefícios. Entre os anos de 1967-2005 foram publicados 578 livros sobre o tema e acredita-se que sejam gastos, anualmente, apenas nos Estados Unidos da América, valores excedentes a US\$1 bilhão com softwares e pesquisas relacionadas [7].

O MEF é um método dito de domínio, e envolve a formulação de uma malha que divide toda a região no qual está definido em elementos finitos ou simplesmente elementos, que por sua vez são conectados entre si através de nós. Apesar de não ser o único método de “discretização” existente, o MEF é atualmente o dominante no cenário dos problemas lineares com o MEC ficando logo atrás. Já para os problemas não lineares a preferência pela utilização do MEF é de maioria esmagadora [12].

1.3.2 Otimização

Otimização é o processo através do qual o objeto de estudo torna-se ótimo, isto é, uma busca por excelência que leva a escolha das melhores alternativas existentes. Em uma definição mais formal, fornecida pela *American Society for Engineering Education Conference Paper*, o objetivo da otimização é conseguir o “melhor” projeto relativo a um conjunto de critérios ou restrições priorizadas, que incluem a maximização de fatores tais como produtividade, longevidade, força, eficiência, confiabilidade e utilização [14,15]. Era de se esperar que a otimização fosse encontrada como objeto de estudo de pesquisadores e profissionais de variados campos. No setor administrativo de empresas, no planejamento estratégico, na gestão para redução de gastos e custos, logística, no estudo para aumento de produtividade ou vendas, na utilização eficiência do tempo dos funcionários, dos processos ou de máquinas, entre outros.

Matematicamente, para problemas de engenharia, deve-se primeiro construir modelos matemáticos que representem fielmente o sistema de estudo e chegar a uma função para representar a performance ou desempenho do mesmo. É justamente esta função que deve ser maximizada ou minimizada, seguindo critérios de desempenho previamente definidos para se obter a solução otimizada prática através de algoritmos, sendo estes os mais eficientes possível.

1.3.3 Otimização Estrutural e Topológica

A otimização estrutural pode ser datada ao século XIX com as teorias relativas ao design de estruturas de James Clerk Maxwell [16] e posteriormente com as aplicações de tais conceitos por Anthony Michell [17]. Maxwell focou seu estudo nas principais estruturas estudadas por engenheiros, as pontes. Engenheiros preocupavam-se em obter modelos precisos das tensões mecânicas nas configurações do projeto. Maxwell por sua vez, procurou aplicar os conceitos da teoria de elasticidade a problemas simples, afim de estudar o risco de falha de uma ponte com a menor quantidade de material possível. Maxwell sugeriu que a estrutura com menor quantidade de material e com cujas propriedades estruturais seriam preservadas seria uma onde as direções principais seriam compostas por elementos de treliça. Essa solução veio mais tarde a ser reconhecida como a solução ótima para a estrutura com a máxima rigidez e o menor peso, dado um carregamento único, minimizando a quantidade de material, o peso da estrutura e o custo associado a produção da mesma. Em suma, pode-se dizer que o projeto sem a otimização pode levar a superdimensionamentos.

Partindo originalmente da Otimização Estrutural e para aplicações de mecânica estrutural, a OT veio a ser reconhecida propriamente dita como um campo de estudo após a publicação de 1988 de Martin Philip Bense e Noboru Kikuchi [18]. Derivado das palavras gregas *topos*, “lugar” e de *logos*, “estudo”, a otimização topológica é um método computacional de distribuição de material para a síntese de estruturas sem qualquer forma preestabelecida, concedendo ao método a liberdade para encontrar “*layouts*” estruturais inovadores e de alta

performance [19]. O problema de OT típico pode ser definido como um problema de programação binária, onde “*loops*” iterativos de análises for elementos finitos, análises de sensibilidade e critérios de otimização são preenchidos para encontrar as porções ineficientes e, portanto, desnecessárias da estrutura. Para cada elemento é atribuída uma densidade individual relativa associada as propriedades físicas do material e são interpoladas entre 0 (vazio, ausência de material) e 1 (sólido) e são usados critérios de penalização para influenciar essa distribuição de material [1,2,20,21].

Kai Liu e Andrés Tovar [19] mapearam recente, além de obras de referência de renome, mais de 11000 publicações acadêmicas sobre o tema e rotinas educacionais para pronta implementação em ferramentas gráficas, tal como o MATLAB[19]. Duas dessas rotinas serão abordadas no presente trabalho; são elas, o programa `Top.m` de 99 linhas para otimização topológica em duas dimensões [1] e o programa `PolyTop` [2] de OT baseada em densidades que opta pelo uso de malhas de elementos finitos poligonais não estruturados, dessa forma, modelando de forma mais precisa a geometria do domínio de projeto e mantendo a eficiência de custo operacional [3].

O emprego de algoritmos computacionais como os discutidos acima leva a inevitáveis ganhos de tempo para projetistas, pois fornecem soluções otimizadas e atenuam as antigas necessidades exaustivas de verificação das melhores distribuições estruturais de material para um dado problema. Auxiliados por capacidades de processamento computacional cada vez maiores, e do uso do MEF, diversos métodos de OT têm sido desenvolvidos. Vale ressaltar o método de leiaute baseado na abordagem material de Bendøe e Kikuchi [18] o método ESO de design evolucionário idealizado por Xie e Steven [20], para mencionar apenas alguns.

1.3.4 Transferência de Calor

Calor é a energia térmica em trânsito devido a uma diferença de temperaturas no espaço, sendo assim, sempre que houver uma diferença entre meios, haverá, necessariamente,

transferência de calor [22]. Nos estudos de transferência de calor é comum a utilização do termo *modo* para identificar os diferentes tipos de processos de transferência de calor. *Condução* é o termo associado ao modo de transferência de calor que ocorre através de um meio estacionário (sólido ou fluido), quando nele existe um gradiente de temperatura. Quando uma superfície e um fluido em movimento estiverem a diferentes temperaturas, a transferência de calor que ocorrerá entre eles será intitulada como *convecção*. *Radiação*, ou *radiação térmica*, refere-se ao último modo de transferência de calor, que representa a energia líquida emitida na forma de ondas eletromagnéticas entre superfícies com temperaturas não nulas e ausência de um meio interposto participante [22]. Na *Figura 1.1* são exibidos os diferentes modos aqui exemplificados, sendo a *condução* o foco de nosso estudo.

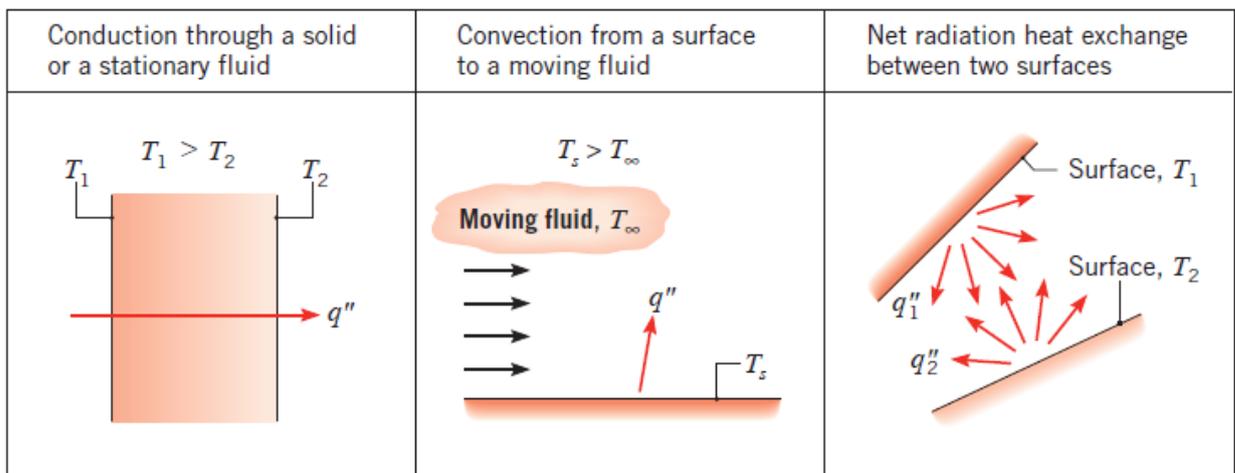


Figura 1.1 - Modos de transferência de calor: condução, convecção e radiação. Fonte: [22]

A *condução* pode ser vista como transferência de energia, a nível atômico e molecular, das partículas mais energéticas para as menos energéticas através de um meio, sendo devido apenas às interações entre partículas. Para ilustrar o conceito pode-se pensar em um gás, contido entre duas superfícies mantidas a temperaturas diferentes. Os conhecimentos de termodinâmica nos levam a confirmar que temperaturas mais elevadas estão associadas a uma maior energia molecular, energia que por sua vez está relacionada aos movimentos translacionais, rotacionais e vibracionais das moléculas. Partículas mais energéticas, relacionadas a proximidade da

superfície mais quente estarão naturalmente mais “agitadas” e ao colidirem com partículas vizinhas em seus movimentos aleatórios, haverá uma transferência de energia, da mais energética para a menos energética. Sendo assim, na presença de um gradiente de temperatura, a transferência de energia por condução deverá necessariamente ocorrer na direção da menor temperatura. Essa transferência de energia líquida devido ao movimento randômico de partículas é comumente chamado de difusão de energia. Na condução através de sólidos essa transferência ou difusão de energia ocorre de maneira similar, sendo a interação atômica das moléculas devido às vibrações dos retículos [22].

A transferência de calor é quantificada pelas equações de calor, que podem ser usadas para calcular a quantidade de energia transferido por unidade de tempo, sendo a equação para a condução de calor térmica conhecida como Lei de Fourier [22].

A equação da taxa de transferência de calor na direção x do meio unidimensional da *Figura 1.2*, com uma distribuição de temperaturas $T(x)$, por unidade de área perpendicular à sua direção de propagação, é dada por:

$$q''_x = -k \frac{dT}{dx} \quad (1.1)$$

Essa taxa, também conhecida como fluxo térmico ($\frac{W}{m^2}$), é proporcional ao gradiente de temperaturas, $\frac{dT}{dx}$ na direção de transferência. O parâmetro k da equação (1.1) é a *condutividade térmica* do meio material, uma propriedade de transporte, característica e singular para cada material. O sinal negativo na equação é devido à transferência de calor se dar da maior temperatura para a menor temperatura, num sentido decrescente [22]. Caso fosse de interesse encontrar a taxa de transferência de calor (W) através de uma superfície de área A , como a da *Figura 1.2*, a equação (1.1) seria reescrita como:

$$q_x = q''_x \cdot A \quad (1.2)$$

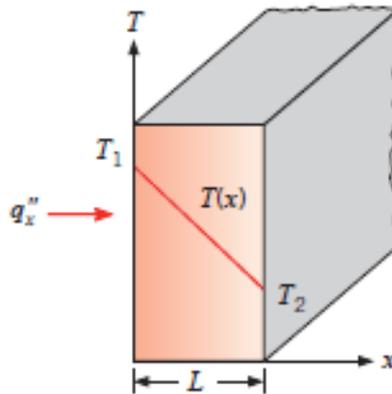


Figura 1.2 – Transferência de calor unidimensional por condução. Fonte: [22]

1.4. Organização do Trabalho

No Capítulo 2 é descrito o funcionamento do método de elementos finitos e todos os conceitos introdutórios necessários para seu entendimento e aplicação. São definidos e apresentados os conceitos de domínio, malha, elemento, nó, sistemas de referência, graus de liberdade, entre outros.

No Capítulo 3 se introduz a otimização topológica e a rotina MATLAB elaborada por Sigmund que foi utilizada para os primeiros entendimentos sobre o tema. Se apresenta também o problema de acesso típico para a OT baseada em densidade, sua formulação e a simulação de OT estrutural da viga MBB. Segue-se com algumas alterações na rotina do 99-Lines para adequá-lo aos problemas de condução de calor.

No Capítulo 4, é mostrado as principais desvantagens da rotina compacta de malha estruturada de 99 linhas de Sigmund e apresentada o programa `PolyTop` [3]; são enumeradas as principais vantagens da utilização de uma malha não estruturada arbitrária de elementos finitos poligonais. Assim como foi feito para problemas de minimização de *compliance* no Capítulo 3, mostra-se o MEF aplicado ao problema de condução de calor e todas as suas deduções.

No Capítulo 5, são apresentados os resultados e exemplos de simulações e são comparados os códigos `Toph.m` (99 lines de Sigmund adaptado para condução de calor) e o `PolyTopHeat` (código de Talischi et al. adaptado para a condução de calor).

O capítulo 6 é reservado para a conclusão, críticas e possíveis desenvolvimentos e melhorias futuras.

2. Método dos Elementos Finitos

2.1. Introdução

Difundido entre as engenharias devido a sua capacidade de analisar diversos tipos de problemas, o Método dos Elementos Finitos (MEF) permite, para citar apenas alguns, o estudo dos deslocamentos e tensões em peças e estruturas das mais diversas formas, fluxos de calor, pressões neutras, determinar percolação e adensamento [23]. Independente da análise a ser feita, do problema ou da aplicação, todas têm em comum o fato de estarem definidas em um determinado domínio, terem que satisfazer condições iniciais e de contorno impostas pelos projetistas para as variáveis fundamentais e necessitarem solucionar um problema modelado através de equações diferenciais parciais.

A ideia básica é “discretizar” o problema definido no espaço contínuo, que nada mais é que utilizar de aproximações para substituir o então infinito conjunto de números que representam as funções desconhecidas por um conjunto finito de parâmetros também desconhecidos que então são encontrados através de funções algébricas.

O sistema algébrico é da forma:

$$[A]\{x\} = \{b\} \quad (2.2)$$

Sendo $[A]$ uma matriz quadrada e não singular, $\{b\}$ o vetor independente e $\{x\}$ o vetor solução, com os parâmetros desconhecidos da solução aproximada [24]. Uma outra denominação frequentemente encontrada na literatura, principalmente no estudo de tensões e deslocamentos, e que será a adotada ao decorrer desse trabalho é:

$$[K][U] = [F] \quad (3.2)$$

Sendo K a matriz de rigidez global, U o vetor de deslocamento global e F o vetor com as forças ou carregamentos atuantes no sistema.

2.2. Conceitos Fundamentais

A ideia base do MEF é discretizar o domínio do problema contínuo, isso é, dividi-lo através de sub-regiões de geometrias simples, conhecidas como “elementos finitos”, como vieram a ser nomeadas por Ray Clough [6]. Esses elementos podem adotarem formatos de quadriláteros, cubos, triângulos, etc., de acordo com as dimensões do problema, buscando sempre representar o domínio da maneira mais fidedigna possível. Elementos conectam entre si através de *nós* ou *pontos nodais* para gerar a *malha* de elementos finitos, conforme pode ser observado na Figura 2.1 [25].

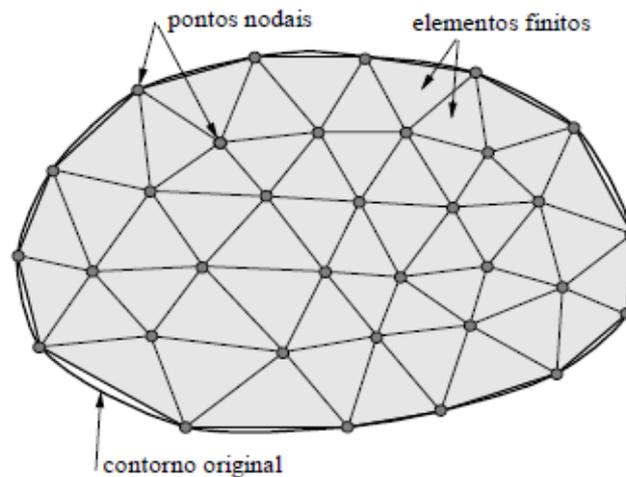


Figura 2.1 – Malha de Elementos Finitos para problemas no plano. Fonte: [25]

Na *Figura 2.2* a seguir, é possível observar diferentes tipos de elementos finitos, definidos de acordo com a sua geometria e número de nós. No caso dos problemas lineares, o número de parâmetros desconhecidos é igual ao número de nós, e dado à implementação via algoritmos computacionais, é rotineiro observar que a precisão da resposta aumenta à medida que se aumenta o número de elementos e nós. Um conceito importante quando se fala do MEF, é em relação a convergência dos resultados. Embora este seja um método de aproximações, é

possível se demonstrar que a medida que se eleva ao limite a quantidade de nós em uma malha consistente, isto é, quando o tamanho individual de cada elemento finito tende a zero, maior será a quantidade de elementos e consequentemente maior será a quantidade de nós na malha, que tenderá ao infinito. Sendo assim, é possível a obtenção de uma solução que converge para a solução exata do problema em questão. Todavia, apesar do aumento do número de nós e quantidade de elementos implicar um subsequente ganho de precisão, também acaba-se aumentando o tempo de máquina gasto, ou seja, o custo de processamento para a solução do problema [7,25].

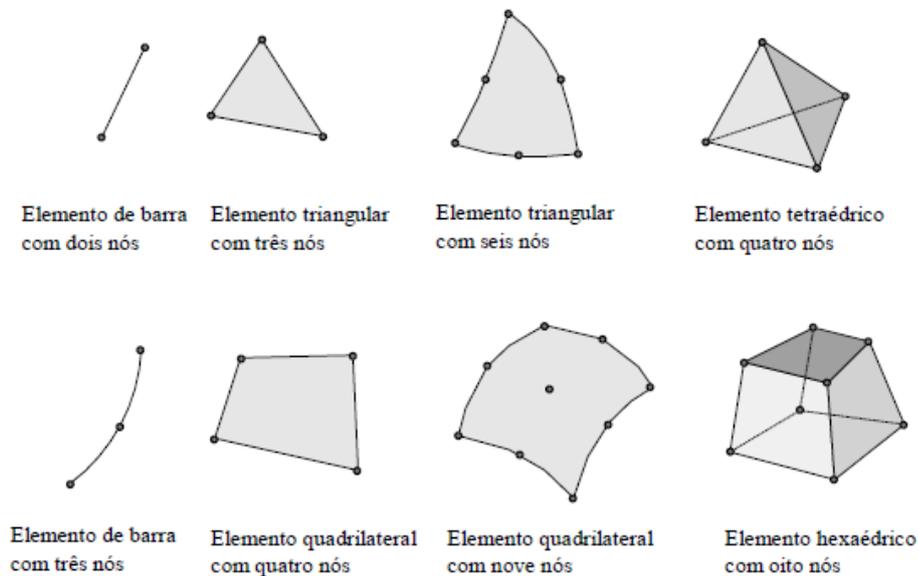


Figura 2.2 – Diferentes tipos de elementos finitos. Fonte: [25]

2.3. O Grau de Liberdade no MEF

O conceito de grau de liberdade (*degrees of freedom*) é um termo originado da Mecânica que explicita a permissividade de movimento de um ponto; as direções no qual este é livre para mover-se. No espaço \mathbb{R}^3 existem três possíveis direções de movimento atribuídas aos possíveis movimentos de translação, o que equivaleria a três graus de liberdade, enquanto no \mathbb{R}^2 , teríamos apenas dois GDL, ou DOF. Entretanto, para um corpo sólido no espaço

tridimensional, deve-se também levar em conta, além dos movimentos de translação, os movimentos de rotação em cada um dos eixos, que somados equivalem a seis graus de liberdade.

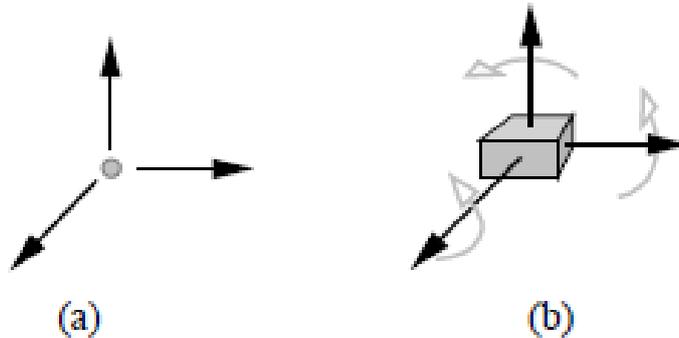


Figura 2.3 – (a) graus de liberdade de um ponto; (b) graus de liberdade de um corpo rígido. Fonte: [25]

Um dado problema, com o mesmo tipo de malha e elementos, pode, portanto, ser interpretado de diferentes formas e gerar comportamentos distintos de acordo com o número de graus de liberdade associados a cada nó e gerar conseqüentemente complexidades distintas. A atribuição do número de graus de liberdade vai depender da dimensão e do tipo do problema a ser analisado. Nos problemas típicos de comportamento mecânico e análises de tensões, os graus de liberdade dos nós replicam os possíveis movimentos que estes podem sofrer; no caso 2D são observadas duas possíveis translações, nos eixos X e Y do plano de coordenadas cartesianas, enquanto no 3D existem três translações, correspondentes aos eixos X, Y e Z. Nos problemas de condução de calor, apesar de não haver deslocamentos, os graus de liberdade seguem sendo utilizados, porém, nesse caso, para representar o valor do campo de temperatura em cada nó da malha.

2.4. Formulação Isoparamétrica

2.4.1. Sistema de Referência Global e Local

Para um elemento sólido tridimensional comum, é padrão a transformação entre o sistema de referência global definido no sistema cartesiano xyz para o sistema local definido com as componentes $\xi\eta\zeta$. Essa modificação traz como benefício a simplificação na hora dos cálculos das matrizes de rigidez e vetores de carregamento ao modificar os limites superior e inferior de integração [26].

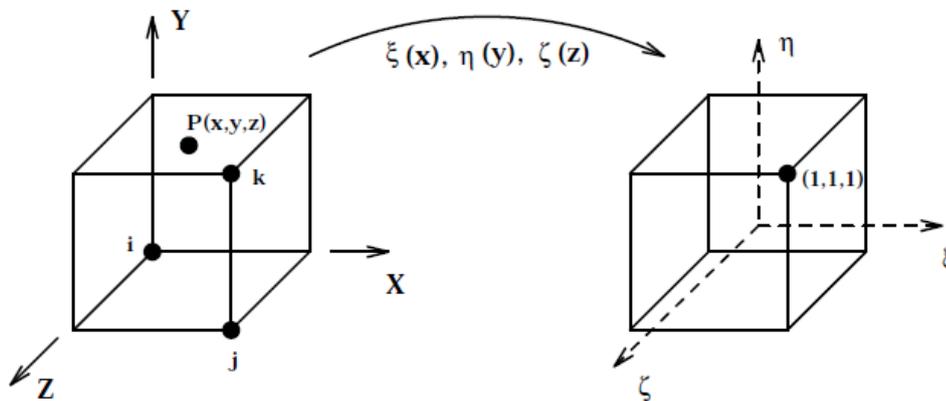


Figura 2.4 – Transformação entre sistemas de referência global e local. Fonte: [26]

A transformação de sistemas de referência também é bastante útil quando o elemento finito de estudo possui uma forma distorcida ou demasiadamente complexa no sistema global. Através de funções de forma é possível fazer com que este elemento fique com seus lados retos no sistema local, conforme pode ser visto na *Figura 2.5*. Além de tornar mais simples a representação e a visualização de elementos, esse novo contorno também ajuda a simplificar cálculos.

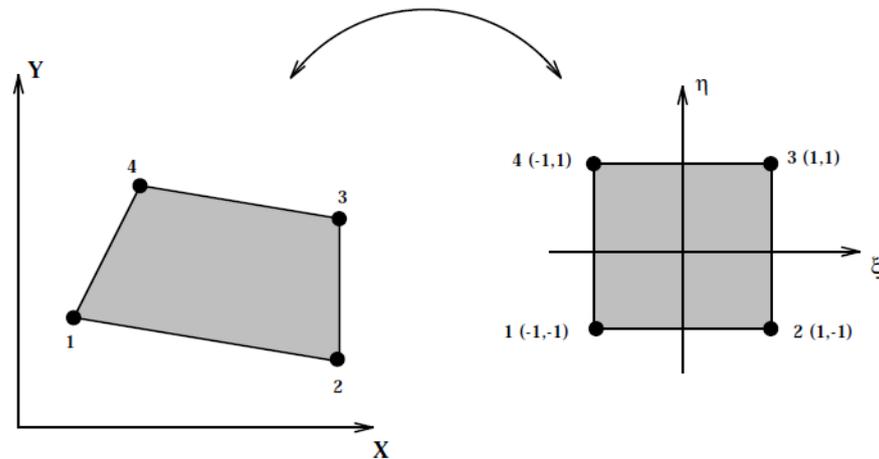


Figura 2.5 – Transformação entre sistemas de referência utilizando funções de forma. Fonte: [26]

2.4.2. Elementos Finitos Poligonais

Para os estudos e pesquisa de EF assim como na sua aplicação prática na indústria, é muito comum se observar o uso dos tradicionais elementos triangulares de três nós de tensão constante assim como os elementos quadrilaterais bilineares de quatro nós, porém, têm crescido a atenção dada aos elementos poligonais devido as suas vantagens. Os benefícios potenciais de sua utilização na computação de elementos finitos incluem: uma maior flexibilidade na geração de malhas de geometrias arbitrárias, uma maior precisão da solução numérica em comparação às soluções obtidas com malhas triangulares e quadrilaterais, capacidade de modelar materiais policristalinos, para mencionar apenas algumas [43]. Essa escolha adiciona versatilidade a uma rotina de elementos finitos, pois, as funções de forma dos elementos poligonais isoparamétricos que serão discutidos a seguir, permitem a construção de qualquer polígono convexo a partir dos polígonos regulares de n lados, incluindo os tradicionais polígonos lineares como os triângulos lineares ($n=3$) e quadriláteros bilineares ($n=4$) [3].

A interpolação dos polígonos se dá através do conceito dos diagramas de Voronoi. Considerando-se um domínio Ω em duas dimensões, descrito por um conjunto N de M nós

espalhados: $N = \{p_1, p_2, p_3, \dots, p_M\}$. O diagrama de Voronoi do conjunto N é a subdivisão do domínio em regiões $V(p_i)$ onde qualquer ponto em $V(p_i)$ está mais próximo do nó p_i do que de qualquer outro nó $p_j \in N$ ($j \neq i$). A Triangulação de Delaunay por sua vez é utilizada para conectar nós com uma lateral de Voronoi em comum, como pode ser observado na *Figura 2.6* para o conjunto nodal constituído de sete nós ($M=7$). O conceito pode ser mais aprofundado no trabalho original do autor [43].

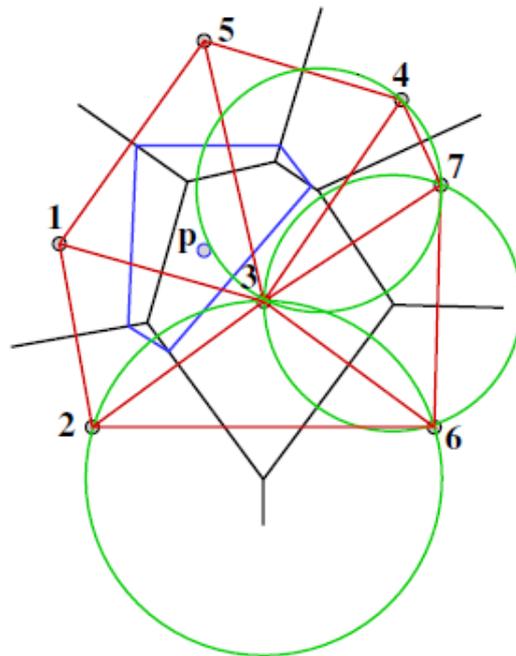


Figura 2.6 – Triangulação de Delaunay em célula de Voronoi composta por sete nós. Fonte: [43]

2.4.3. Funções de Forma

Para cada nó de um elemento finito deve-se associar uma função de forma que recebem o valor de 1 no nó e 0 nos demais. Para um elemento bidimensional, com m e n nós nas direções ξ e η , tal como os elementos utilizados no código `TOP.m` de Sigmund [1], tem-se um total de mn funções de forma, dadas por [26]:

$$N_1 = \frac{(1 - \xi) \cdot (1 + \eta)}{4} \quad (2.3)$$

$$N_2 = \frac{(1 + \xi) \cdot (1 + \eta)}{4} \quad (2.4)$$

$$N_3 = \frac{(1 + \xi) \cdot (1 - \eta)}{4} \quad (2.5)$$

$$N_4 = \frac{(1 - \xi) \cdot (1 - \eta)}{4} \quad (2.6)$$

As expressões acima podem variar de acordo com a forma com que os nós do elemento foram numerados e percorridos, sendo elas associadas ao elemento exibido abaixo. Com os nós do elemento sendo numerados de 1 a 4 e percorridos no sentido horário a partir do canto superior esquerdo [27].

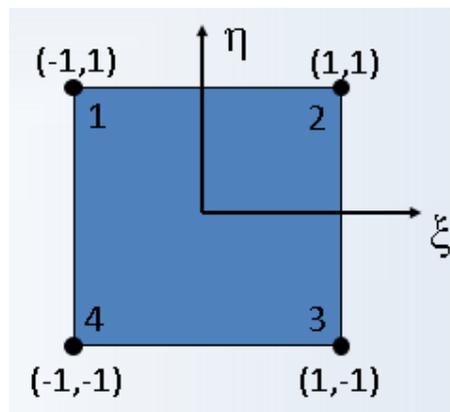


Figura 2.7 – Elemento bidimensional quadrangular linear. Fonte: [27]

A *Figura 2.7* ilustra o que foi explicado acima para o comportamento das funções de forma, com quatro casos particulares do elemento quadrangular, onde para cada caso, apenas um nó recebe o valor 1 enquanto os demais recebem o valor de 0, isto é, apenas um dos nós está selecionado. Sendo assim, as funções de forma não são apenas úteis quando se deseja interpolar uma geometria, ou a transformação entre sistemas de referência global e local, mas também são utilizados, como será visto mais à frente para os cálculos dos deslocamentos nodais.

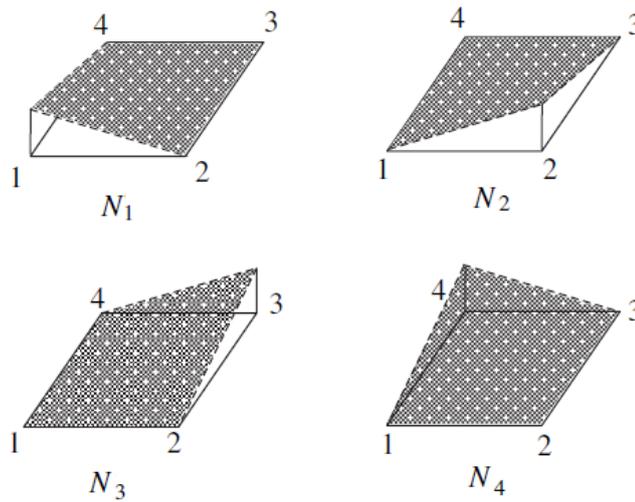


Figura 2.8 – Demonstração visual do funcionamento das funções de forma dos elementos quadrangulares lineares. Fonte: [7]

Para os elementos finitos poligonais utilizados pelo algoritmo `PolyTop` de Talischi et al.[3] as funções de forma são definidas de acordo com o trabalho de Sukumar [43], e serão utilizadas na função auxiliar `PolyShapeFnc`, que computa o conjunto de funções de forma lineares para o polígono de referência de n lados em um ponto interior ξ . A função de forma correspondente ao nó i do elemento finito poligonal, dado que $1 \leq i \leq n$, é definida a seguir [3]:

$$N_i(\xi) = \frac{\alpha_i(\xi)}{\sum_{j=1}^n \alpha_j(\xi)} \quad (2.7)$$

onde α_i são os interpolantes da forma, i o nó em questão e A a área do triângulo formado, conforme pode ser visto na *Figura 2.9*:

$$\alpha_i(\xi) = \frac{A(p_{i-1}, p_i, p_{i+1})}{A(p_{i-1}, p_i, \xi)A(p_i, p_{i+1}, \xi)} \quad (2.8)$$

Adotando a notação:

$$A_i(\xi) = A(p_{i-1}, p_i, \xi) \quad (2.9)$$

A equação (2.8) pode ser expressa em sua forma reduzida:

$$\alpha_i(\xi) = \frac{1}{A_i(\xi)A_{i+1}(\xi)} \quad (2.10)$$

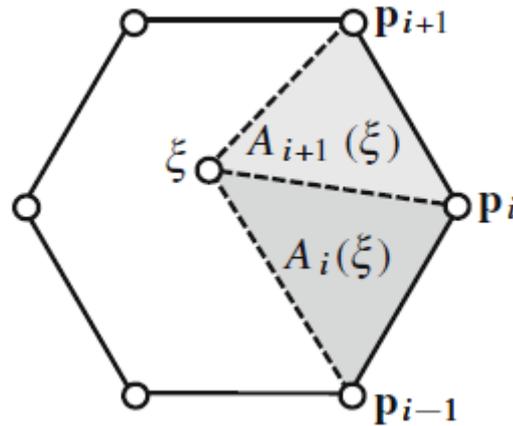


Figura 2.9 – Áreas utilizadas na definição de α_i (equação 2.8). Fonte: [3]

2.4.4. Cálculo da Matriz de Rigidez

O cálculo da matriz de rigidez dos elementos finitos envolve, através da matriz de deformação $[B]$, a derivada das funções de forma em relação às suas coordenadas globais.

Considerando para tanto, um nó i de um elemento qualquer,

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z} \frac{\partial z}{\partial \xi} \quad (2.11)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \eta} + \frac{\partial N_i}{\partial z} \frac{\partial z}{\partial \eta}$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \zeta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \zeta} + \frac{\partial N_i}{\partial z} \frac{\partial z}{\partial \zeta}$$

As equações apresentadas acima também podem ser expressas em sua forma matricial,

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} \quad (2.12)$$

onde $[J]$ é a matriz de transformação, ou simplesmente, Jacobiano de transformação. Nas coordenadas globais x,y,z as funções de forma da equação (2.12) ficam:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} \quad (2.13)$$

O Jacobiano é bastante utilizado para substituir os diferenciais de volume dV ou de área dA , no caso plano, tornando o cálculo das matrizes e vetores de carregamento dos elementos finitos em função das coordenadas locais ξ,η,ζ [26].

No caso geral, a matriz de rigidez do elemento é dada por [2]:

$$[K_e] = \int_V [B]^T [D][B]dV = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 [B]^T [D][B]|\det[J]|d\xi d\eta d\zeta \quad (2.14)$$

No caso plano dos elementos bidimensionais a equação (2.14) se reduz a:

$$[K_e] = \int_A [B]^T [D][B]dA = t \int_{-1}^1 \int_{-1}^1 [B]^T [D][B]|\det[J]|d\xi d\eta \quad (2.15)$$

Sendo $[B]$ a matriz de deformação, as vezes também denominada de matriz SN, $[D]$ a matriz de condutividade do material, $[J]$ a matriz do Jacobiano e t a espessura do elemento. Vale ressaltar que as integrações das equações (2.14) e (2.15) não poder ser realizadas analiticamente, tendo que para resolvê-las, utilizar de técnicas de integração numéricas, como a integração Gaussiana entre determinados pontos de Gauss e se empregando certos pesos neles (*weighting*) [2,7,27].

3. Otimização Topológica

3.1. Introdução

Bendsøe e Sigmund esclarecem os ingredientes básicos de como achar o *lay-out* ótimo do método de distribuição de material em estruturas linearmente elásticas em sua obra. Definindo para tanto, o *layout* de uma estrutura como uma combinação de informações que abrangem topologia, formato e tamanho. Na OT de estruturas sólidas determina-se por exemplo, o número, localização e formato dos furos na estrutura, assim como a conectividade do domínio. Nos problemas tradicionais, as únicas informações disponíveis são quanto ao volume da estrutura a ser construída, as condições de suporte, as cargas aplicadas e possivelmente alguma condição de restrição de design. Os problemas mais simples em relação às suas funções objetivas e de restrição que levam ao entendimento do tema são os problemas de projetos para minimização de “*compliance*”, ou seja, minimizando a flexibilidade para a maximização da rigidez estrutural global [2].

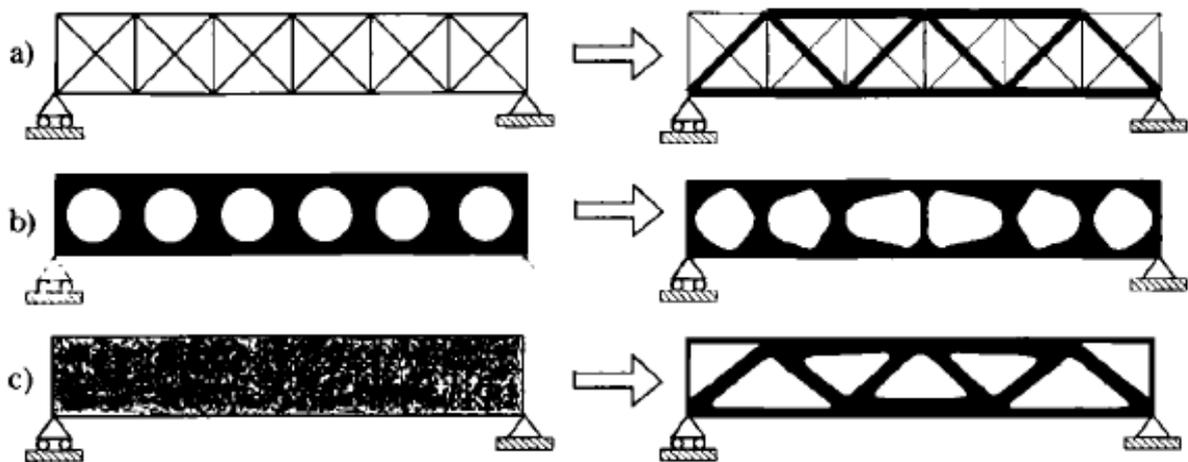


Figura 3.1 – *Categorias de otimização estrutural.* Fonte: [2]

Na *Figura 3.1* acima, são exibidas as diferentes categorias de otimização estrutural com os problemas originais exibidos na esquerda e suas respectivas otimizações exibidos na direita.

Na figura a) temos a otimização de tamanho (*sizing*) de uma treliça, na b) a de formato e na c) a otimização topológica. Ao contrário das duas primeiras otimizações, a OT permite a inserção e/ou a retirada de furos na estrutura, tornando-se, portanto, uma verdadeira ferramenta de projeto ótimo [28].

3.2. A 99 Line Topology Optimization Code

Publicado em 1999, o *paper* acadêmico intitulado “A 99 line topology optimization code written in Matlab” do professor de engenharia mecânica da DTU (Universidade Técnica da Dinamarca), Ole Sigmund, veio a se tornar referência no estudo educacional da otimização topológica para minimização de *compliance* em estruturas estaticamente carregadas [1,19]. O *paper* apresenta um código enxuto de OT no software Matlab com apenas 99 linhas, sendo 36 linhas para o programa principal, 12 linhas para o “otimizador” baseado em OC, 16 linhas para o filtro de independência de malha e 49 linhas de código para a implementação por EF [1]. O próprio autor defende o seu uso para desenvolver a intuição e as habilidades dos alunos em desenvolver designs ótimos, sendo aconselhado adaptações ao código para atender a outros tipos de problemas e restrições.

3.2.1. Considerações Iniciais

Caso fosse necessário a elaboração de um passo-a-passo para a facilitar a implantação bem-sucedida da OT, algo como as etapas exibidas a seguir seria apresentado. Em primeiro lugar, deve-se definir o domínio do problema, tendo este o maior tamanho possível para que não venha a ser um fator limitante na execução da OT. Restrições impactam negativamente no desempenho da solução e reduzem a qualidade e precisão da otimização [29]. A *Figura 3.2* ilustra um corpo bidimensional com domínio Ω e contorno Γ , com sistema de referência em coordenadas cartesianas.

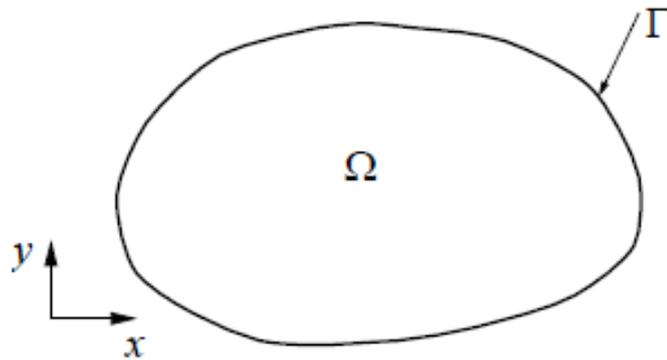


Figura 3.2 – Corpo bidimensional com domínio Ω e contorno Γ .

Como passo seguinte, o domínio deve ser “discretizado”, isto é, dividido em elementos finitos, aplicadas as devidas condições de contorno e então esses dados devem alimentar o software de OT escolhido. Se tratando de MEF, deve-se escolher uma malha que diminua a diferença entre o domínio real e o aproximado; as malhas não aproximam o domínio da mesma forma, certos tipos de malha mostram ser mais eficientes do que outras [30-35].

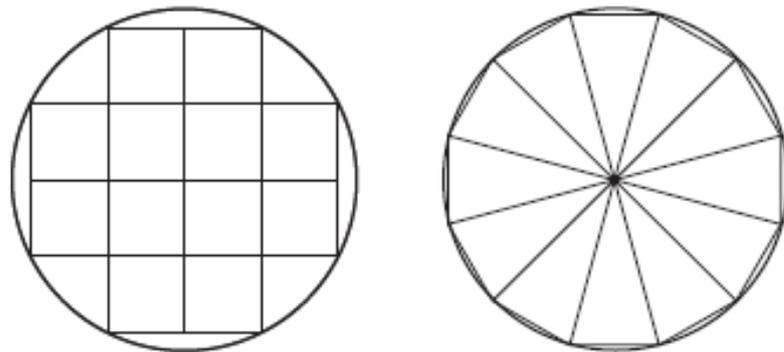


Figura 3.3 – Domínio aproximado por diferentes malhas.

A Figura 3.3 mostra duas aproximações distintas de um domínio de contorno suave, sendo a primeira malha composta por elementos finitos quadrados e a segunda, mais à direita, por elementos finitos triangulares. Através dessa visualização fica evidente, ao preencherem melhor o espaço interno, certas malhas aproximam melhor o domínio. Aumentando o número de

elementos e diminuindo seus respectivos tamanhos também é possível se obter ganhos de precisão e melhor preencher um dado domínio, como pode ser visto na *Figura 3.4* de refino de malha [7].

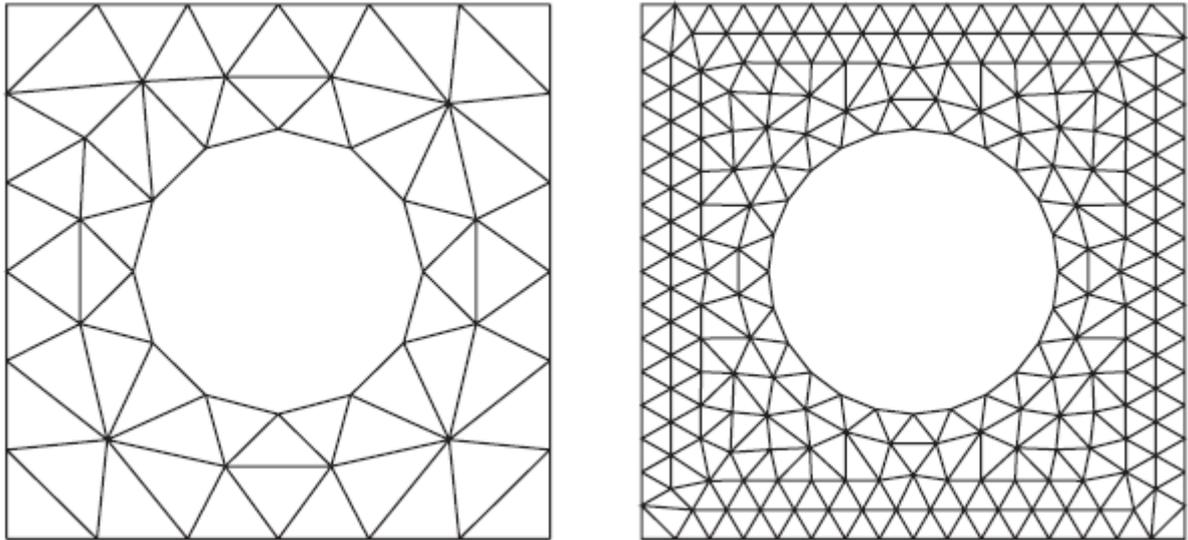


Figura 3.4 – Modelo de elementos finitos e modelo de elementos finitos refinado.

Fonte: [7]

O algoritmo do software, nesse caso o MATLAB, por sua vez, distribui o material no domínio em um processo iterativo de forma a minimizar a flexibilidade da viga, que nesse caso é a função objetivo (*objective function*) especificada [2,29]. A topologia obtida será apresentada de uma forma binária, com densidades individuais relativas associadas as propriedades físicas do material ao longo da malha, com 0 representando a ausência de material e sendo indicada com a cor branca e 1 o material sólido, exibido em preto, conforme pode ser visto na *Figura 3.1* [1,2,20,21].

3.2.2. Formulação do Problema

Para que o código em MATLAB pudesse ser escrito de forma concisa e enxuta, da forma com que Ole Sigmund o fez, algumas simplificações e convenções tiveram que ser

adotadas. Ao contrário dos domínios com formatos arbitrários exibidos anteriormente, o domínio de design foi adotado como sendo retangular e discretizado por elementos finitos quadrangulares. Dessa forma, dada à similaridade geométrica entre o domínio e os EF, a malha pode preencher corretamente toda a região de estudo. A numeração e a maneira com que elementos e nós são percorridos também se tornou mais fácil e intuitiva; sempre de coluna a coluna começando pelo canto superior esquerdo conforme exibido na *Figura 3.5* abaixo. Cada elemento foi identificado por sua posição na malha, correspondendo o primeiro número a linha e o segundo a coluna ocupada (ely, elx). Já os nós foram numerados percorrendo-se o domínio coluna-a-coluna a começar pelo canto superior esquerdo, como pode ser visto na *Figura 3.6*.

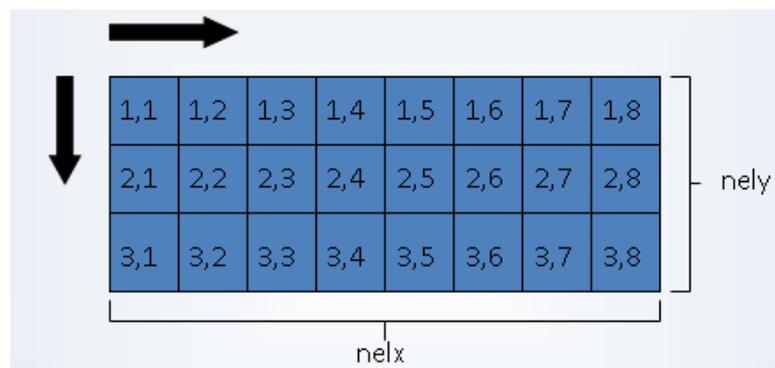


Figura 3.5 – Numeração dos elementos da malha. Fonte: [27]

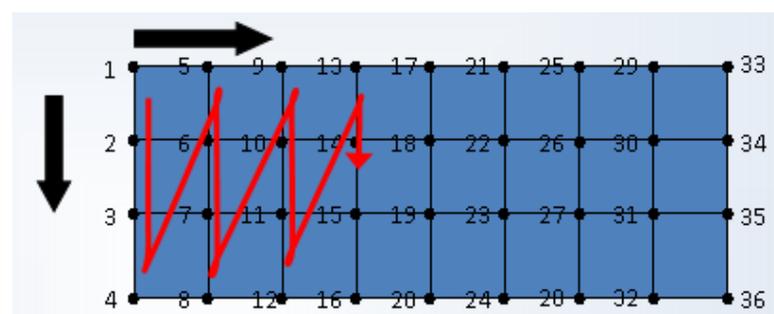


Figura 3.6 – Numeração dos nós da malha. Fonte:[27]

O problema de otimização foi formulado de forma genérica e apresentado da seguinte forma: ache a distribuição de material x para cada x_e resultando em um subdomínio $d\Omega$ dentro de um domínio fixo e preestabelecido Ω , que minimiza a função de custo (função objetiva) $c(x)$, sujeito a uma fração volumétrica prescrita f .

Matematicamente, isso pode ser escrito:

$$\begin{aligned} \min_x: \quad & c(x) = U^T K U = \sum_{e=1}^N (x_e)^p u_e^T k_0 u_e & (3.1) \\ \text{Sujeito a:} \quad & \frac{V(x)}{V_0} = f \\ & K U = F \\ & 0 < x_{min} \leq x \leq 1 \end{aligned}$$

onde U e F são os vetores globais de deslocamento e de força, respectivamente, K é a matriz global de rigidez, u_e e k_e correspondem aos vetores de deslocamento e matriz de rigidez do elemento, x é o vetor com variáveis de projeto, x_{min} é o vetor com densidades relativas mínimas, N é o número de elementos usado para discretizar o domínio, sendo ele o produto do número de elementos nas direções X e Y (ambos fornecidos como *input* do usuário), p é o expoente de penalização (*penalization power*) previsto no método SIMP [36-39], $V(x)$ é o volume do material e V_0 o volume do domínio de projeto, e f (*volfrac*) é a fração volumétrica prescrita para o problema [1].

Dessa forma, uma restrição é a quantidade de material que se pode remover da estrutura (*volfrac*, definida pelo usuário). A função objetiva a ser minimizada é a de *compliance*; pretende-se minimizar o trabalho externo gerado pelo carregamento enquanto se estipula uma restrição volumétrica na estrutura para se minimizar a quantidade de material utilizada [28].

3.2.3. Implementação do Código

Utilizando o código de MATLAB, e seguindo a forma programada por Sigmund para a chamada da função, pode-se facilmente obter a OT para viga MBB. A figura obtida corresponderá à estrutura bi-apoiada sujeita a um carregamento pontual em seu centro, com o menor peso e cuja resistência é máxima. A chamada da função TOP segue:

$top(nelx, nely, volfrac, penal, rmin)$

onde $nelx$ e $nely$ são os números de elementos nas direções horizontal e vertical do domínio, $volfrac$ é a fração volumétrica prevista na otimização, que limita a quantidade de material a ser retirada, $penal$ é o expoente de penalização e $rmin$ é o raio mínimo do filtro.

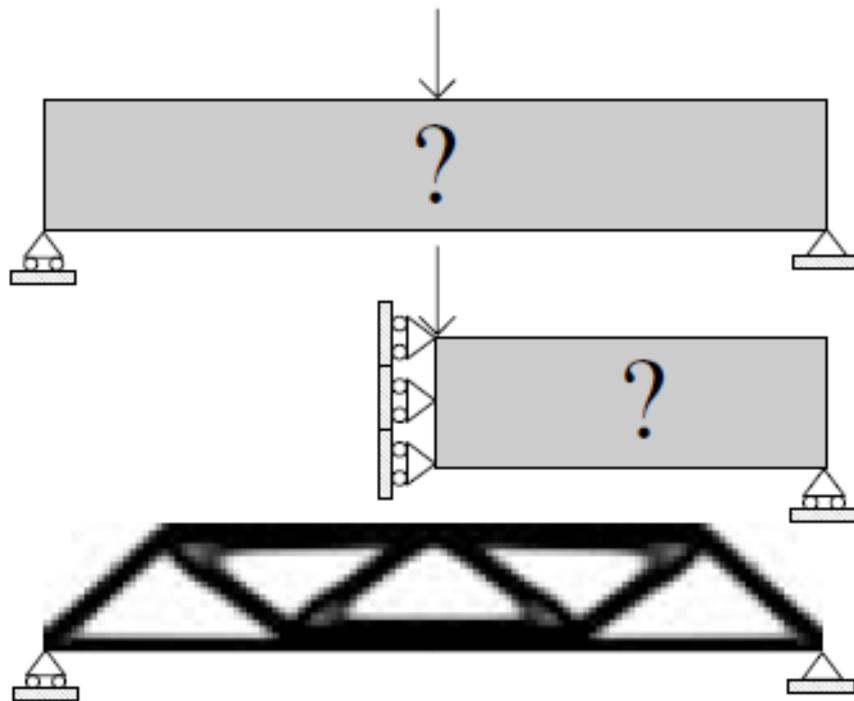


Figura 3.7 – Otimização topológica da viga MBB. No topo: domínio de projeto completo, ao centro: representação de sua metade com condições de simetria, e abaixo: a exibição da estrutura otimizada topologicamente. Fonte: [1]

Na *Figura 3.8* é possível ver algumas capturas de tela do processo de iteração do algoritmo, que começa com a distribuição de material de forma homogênea no domínio de projeto e à medida que é executado e a função de custo é minimizada, exibe uma imagem referente a atual distribuição de densidades no domínio de projeto. No começo é possível ver a coloração cinza correspondente aos materiais intermediários que não podem ser implementados

na prática, e ao final é possível ver a exibição das densidades do tipo 1-0, onde a cor preta indica a presença de material e o vazio a sua ausência.

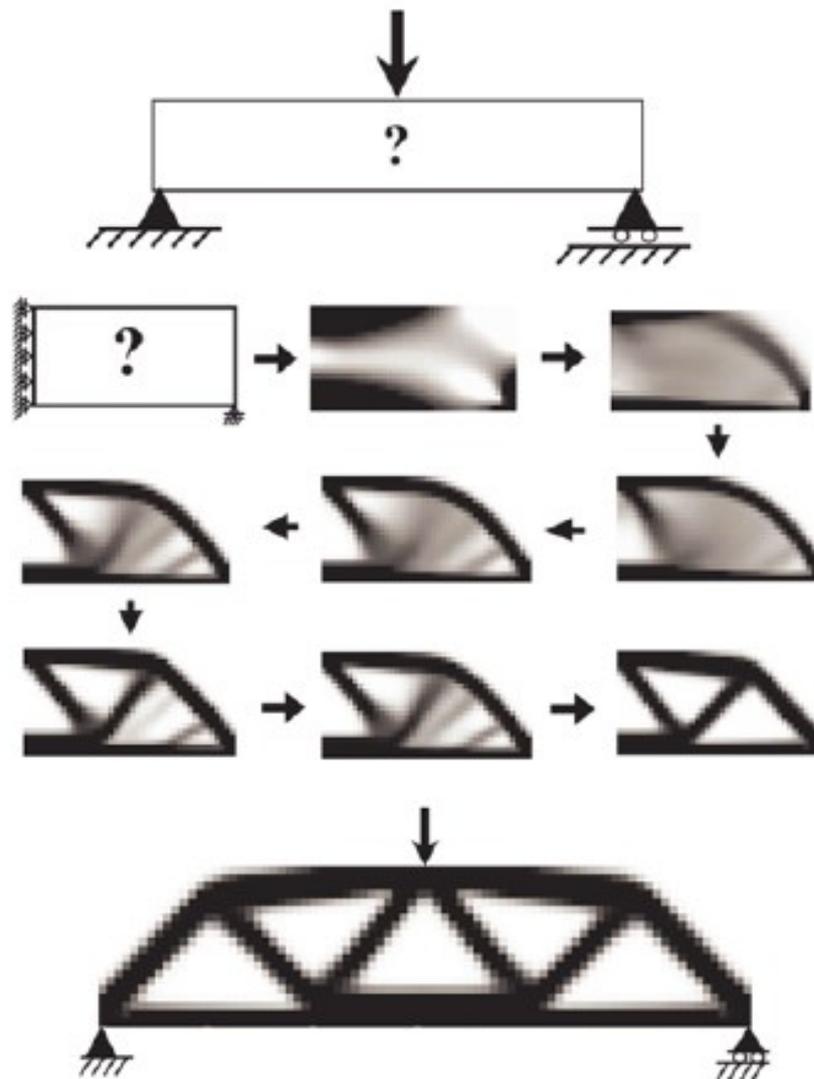


Figura 3.8 – Distribuição de material no interior de um domínio e ao longo de algumas iterações. Fonte: [40]

O resultado da OT obtida com o algoritmo de otimização do *99 lines* de Sigmund se aproxima da estrutura de treliças como sugerido nos primeiros *papers* de otimização estrutural do século XX [29].

3.2.4. Adaptação do Código

Sigmund advoga que apesar da aparente complexidade do método de OT, um problema simples de minimização de *compliance*, como o definido na equação (3.1), pode ser resolvido através de um programa com poucas linhas de código e com o auxílio de linguagens de programação de alto nível, tal como o Matlab [1]. Em uma obra subsequente, Bendsøe e Sigmund brevemente discutem extensões do código de 99 linhas inicial, como a adaptação para outras condições de contorno, o caso com carregamento múltiplo ou até mesmo para a condução de calor [2].

Seguindo em frente com a última sugestão, o código original [1], foi modificado para passar de um algoritmo para a minimização de flexibilidade de vigas MBB para um que resolvesse condução de calor pura em um domínio de projeto quadrado com distribuição de calor uniforme. A função objetivo passou a ser a minimização da temperatura média do domínio de projeto sujeito a uma dada restrição de volume, onde foi considerada a distribuição de dois materiais com condutividades isotrópicas de 1 e 0.001, respectivamente. Também foi incluído um dissipador de calor, posicionado no centro da lateral esquerda do domínio de projeto, cuja temperatura é fixada em zero.

A chamada do código permanece a mesma, sendo passados os números de elementos nas direções horizontal e vertical, a fração volumétrica de material o expoente de penalização e o tamanho do filtro como pode ser visto abaixo:

```
toph(nelx,nely,volfrac,penal,rmin)
```

A *Figura 3.9* exhibe à esquerda, o domínio retangular com aquecimento distribuído uniforme e o posicionamento do dissipador de calor, e à direita, a topologia resultante. O código completo com todas as alterações feitas pode ser encontrado [1,2].

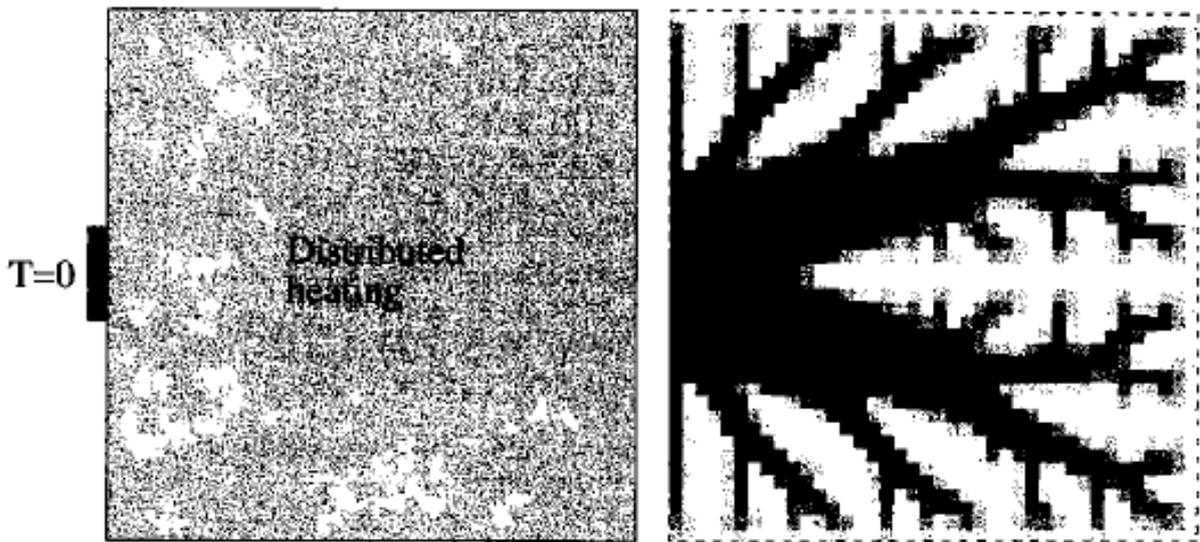


Figura 3.9 – Otimização topológica da condução de calor. Fonte: [1]

Abaixo foram listadas as modificações necessárias para transformar o código de 99 linhas para OT para a minimização de *compliance* no código de 91 linhas para a OT de condução de calor, conforme as deduções feitas na seção 2.4.1-2.4.5. Entre as modificações feitas, incluem adaptar o problema nas condições dadas (meio isotrópico e homogêneo) para apenas um grau de liberdade (temperatura nodal), definir as condições de contorno do dissipador de calor, cuja temperatura é nula, definir o vetor carregamento, o domínio, a nova matriz de rigidez K_e , que nesse caso será chamada de matriz de condutividade e as funções de forma.

Os números indicam a linha de código onde será feita a mudança, o operador matemático < indica a retirada da linha indicada enquanto > indica a nova linha a ser inserida [2].

```

20,22c20,22
< Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1)
< c = c + x(ely,elx)^penal*Ue'*KE*Ue;
< dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;

> Ue = U([n1; n2; n2+1; n1+1],1);
> c = c + (0.001+0.999*x(ely,elx)^penal)*Ue'*KE*Ue;
> dc(ely,elx) = -0.999*penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
68,69c68,69
< K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
< F = sparse(2*(nely+1)*(nelx+1),1); U = sparse(2*(nely+1)*(nelx+1),1);

> K = sparse((nelx+1)*(nely+1), (nelx+1)*(nely+1));
> F = sparse((nely+1)*(nelx+1),1); U = sparse((nely+1)*(nelx+1),1);
74,75c74,75
< edof [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
< K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;

> edof = [n1; n2; n2+1; n1+1];
> K(edof,edof) = K(edof,edof) + (0.001+0.999*x(ely,elx)^penal)*KE;

78,81c78,81
< % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
< F(2,1) = -1;
< fixeddofs = union([1:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
< alldofs [1:2*(nely+1)*(nelx+1)];

> % DEFINE LOADS AND SUPPORTS (SQUARE PLATE WITH HEAT SINK)
> F(:,1) = 0.01;
> fixeddofs = [nely/2+1-(nely/20):2:nely/2+1+(nely/20)];
> alldofs [1:(nely+1)*(nelx+1)];
88,99c88,91
< E = 1.;
< nu = 0.3;
< k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...

< -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
< KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
< k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
< k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
< k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
< k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
< k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
< k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
< k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];

> KE = [ 2/3 -1/6 -1/3 -1/6
> -1/6 2/3 -1/6 -1/3
> -1/3 -1/6 2/3 -1/6
> -1/6 -1/3 -1/6 2/3];

```

4. Equacionamento para o Problema de Condução de Calor

4.1. Introdução

Apesar do *99 lines*, a rotina em MATLAB criada por Sigmund, servir como um bom ponto de partida para as implementações de OT e uma referência para uso educacional, ela é dotada de falhas. A sua própria versatilidade e usabilidade é obtida ao custo da redução da velocidade computacional, sendo até cinco vezes mais lenta que um código similar em linguagem mais “baixo nível” tais como Fortran ou C++ [2]. Sendo assim o código para otimização estrutural e para condução de calor apresentados nas seções 3.2.3 e 3.2.4 deixam de ser práticos quando se passa a utilizar dezenas de milhares de elementos.

Em uma análise baseada em elementos, a parametrização do problema e sua solução aproximada está fortemente relacionadas com a discretização do domínio, e o fato de cada elemento do domínio estar associado a uma variável constante de projeto limita a resolução da topologia, principalmente caso elementos regulares sejam utilizados [28]. As características dos elementos usados na discretização do domínio acabam por influenciar os problemas de OT e o problema do “tabuleiro de xadrez” (*Figura 4.1*) não é algo difícil de se encontrar. Essa descontinuidade, exhibe simultaneamente elementos de cor preta, que representam o material sólido e elementos de cor branca, que representa a ausência de material, não trazendo nenhum significado real prático [28,29,40]. Mesmo que se utilizem filtros de vizinhança para minimizar essas instabilidades e eliminar a variação brusca de propriedade entre elementos adjacentes, não se elimina o problema por completo visto que a malha utilizada na formulação do MEF permanece sendo quadriculada. Sendo assim, adotar malhas estruturadas como é o caso para ambos os códigos discutidos até o momento, acabam por gerar instabilidades numéricas e na prática, restringir as áreas de aplicação da OT. Dependendo do domínio de projeto escolhido, a

malha gerada por elementos quadrangulares não será capaz de preencher totalmente o domínio sendo a melhor escolha para esses casos, uma malha não estruturada de EF poligonais.



Figura 4.1 – O efeito do “tabuleiro de xadrez” ou instabilidade de tabuleiro.

Fonte: [40]

Introduzido mais recentemente por Talischi et al.[3], os algoritmos de código aberto `PolyTop` e `PolyMesher`, também em MATLAB, adotam elementos finitos poligonais para a otimização topológica baseada em densidade; tornando-os adequados para domínios de projeto arbitrários não cartesianos em duas dimensões [19]. Talischi et al apresentam um código para OT estrutural que inclui uma rotina de EF baseados em elementos finitos poligonais isoparamétricos de estrutura modular, malhas de domínio de projeto arbitrárias e não estruturas e permanecem com a usabilidade e eficiência das rotinas de OT educacionais anteriores [3]. Os autores também distinguem para a forma desacoplada do código e sua inerente capacidade de ser adaptada para problemas mais complexos, o que apenas reforçam a escolha do `PolyTop` em contrapartida ao código *99 Lines* de Sigmund para a adaptação e uso em problemas de condução de calor.

4.2. Equações Governantes do Problema

Apresenta-se a seguir inicialmente, a “forma forte” das equações diferenciais que governam o problema. Considere novamente a *Figura 3.2*; um corpo bidimensional com domínio Ω e contorno Γ , e adotando o sistema de coordenadas cartesianas (x,y) como o sistema de referência. $Q(x,y)$ será a taxa de geração de calor por unidade de volume e tempo, interna ao domínio Ω , e as componentes do vetor fluxo de calor por unidade de área e tempo, $q_x(x,y)$ e $q_y(x,y)$.

$$q = q(x,y) = \begin{cases} q_x(x,y) \\ q_y(x,y) \end{cases} \quad (4.1)$$

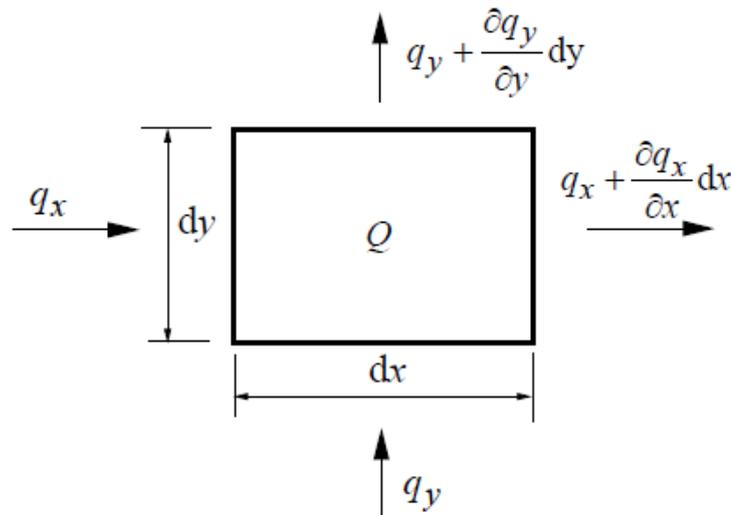


Figura 4.2 – Elemento diferencial com fluxo de calor atravessando seu contorno.

Fonte: [25]

Partindo de um elemento diferencial da *Figura 4.2* de lados dx e dy com fluxo de calor atravessando seu contorno pode-se chegar a equação que governa o problema estacionário de condução de calor [25]:

$$-\frac{\partial q_x}{\partial x} - \frac{\partial q_y}{\partial y} + Q = 0 \text{ em } \Omega \quad (4.2)$$

Ou, de uma forma mais compacta:

$$-\operatorname{div}q + Q = 0 \text{ em } \Omega \quad (4.3)$$

$$-\nabla^T q + Q = 0 \text{ em } \Omega \quad (4.4)$$

Visto que:

$$\nabla^T q = \left(\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \right) \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = \frac{\partial q_x}{\partial x} - \frac{\partial q_y}{\partial y} = \operatorname{div}q \quad (4.5)$$

Para o caso bidimensional geral, o vetor fluxo de calor é função do gradiente de temperatura T

$$q = -k\nabla T \quad (4.6)$$

Sendo k a matriz de condutividade térmica dada por

$$k = k(x, y) = \begin{bmatrix} k_{xx}(x, y) & k_{xy}(x, y) \\ k_{xy}(x, y) & k_{yy}(x, y) \end{bmatrix} \quad (4.7)$$

e podendo o gradiente de temperatura, ∇T , ser reescrito por

$$\nabla T = T \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix} \quad (4.8)$$

A equação (4.6) pode ser reescrita com a (4.7) e (4.8) para a forma

$$\begin{Bmatrix} q_x \\ q_y \end{Bmatrix} = - \begin{bmatrix} k_{xx} & k_{xy} \\ k_{xy} & k_{yy} \end{bmatrix} \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{Bmatrix} = \begin{Bmatrix} - \left(k_{xx} \frac{\partial T}{\partial x} + k_{xy} \frac{\partial T}{\partial y} \right) \\ - \left(k_{xy} \frac{\partial T}{\partial x} + k_{yy} \frac{\partial T}{\partial y} \right) \end{Bmatrix} \quad (4.9)$$

A substituição as equações individuais de q_x e q_y da equação (4.9) na equação (4.2) que governa o problema estacionário de condução de calor, tem-se:

$$\frac{\partial}{\partial x} \left(k_{xx} \frac{\partial T}{\partial x} + k_{xy} \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial y} \left(k_{xy} \frac{\partial T}{\partial x} + k_{yy} \frac{\partial T}{\partial y} \right) + Q = 0 \quad (4.10)$$

Seguindo novamente as hipóteses [25], considera-se $k_{xy} = 0$ e $k_{xx} = k_{yy} = k$ visto que as direções cartesianas coincidem com as direções principais do material e estamos considerando o caso particular de meio isotrópico, onde a condutividade térmica é a mesma em todas as direções [2].

Sendo assim, matriz de condutividade térmica passa a ser dada por

$$k = k(x, y) = \begin{bmatrix} k(x, y) & 0 \\ 0 & k(x, y) \end{bmatrix} = k(x, y) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.11)$$

Onde $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ é a matriz identidade de ordem 2.

Para reproduzir os resultados da seção (3.2.4) e da *Figura 3.9*, o meio foi considerado, além de isotrópico, como sendo homogêneo, pois, a condutividade térmica não depende das coordenadas (x, y) , conhecimento esse que torna k_{xy} e k_{yx} constantes.

Neste caso particular de meio homogêneo e isotrópico, a equação (4.10) é escrita como

$$k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q = 0 \quad (4.12)$$

Ou em uma forma mais compacta, como ficaria caso, ao invés de todas essas etapas, fosse substituída a equação (4.6) na equação (4.4):

$$k \nabla^2 T + Q = 0 \quad (4.13)$$

A “forma fraca” da equação governante que rege o problema de condução de calor e que possibilita a aplicação direta do MEF também será apresentada a seguir [7]:

Escrevendo na forma matricial, ache $T(x, y) \in U$ tal que,

$$\int_{\Omega} (\nabla w)^T D \nabla T d\Omega = - \int_{\Gamma_q} w^T \bar{q} d\Gamma + \int_{\Omega} w^T s d\Omega \quad \forall w \in U_o \quad (4.14)$$

Onde

$$\nabla T = \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} \quad D = \begin{bmatrix} k_{xx} & k_{xy} \\ k_{xy} & k_{yy} \end{bmatrix} \quad (4.15)$$

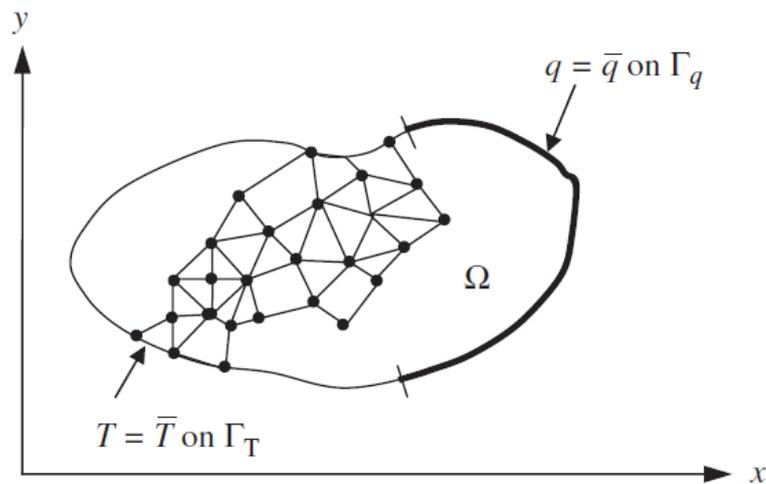


Figura 4.3 – Modelo de elementos finitos em duas dimensões na forma fraca para o problema de condução de calor. Fonte: [7]

Fazendo com que o domínio de projeto seja dividido em uma combinação de elementos quadrangulares e triangulares, assim como o é na *Figura 4.3*, o número total de elementos é n_{el} e o domínio de cada elemento é Ω^e . Segue portanto, ao adotar um somatório das integrais sobre os elementos n_{el} e a equação (4.14) que:

$$\sum_{e=1}^{n_{el}} \left(\int_{\Omega^e} (\nabla w^e)^T D^e (\nabla T^e) d\Omega = - \int_{\Gamma_q^e} w^{eT} \bar{q} d\Gamma + \int_{\Omega^e} w^{eT} s d\Omega \right) = 0 \quad \forall w \in U_o \quad (4.16)$$

A aproximação da solução e das funções peso associadas a cada elemento são escritas em termos do número do número de nós do elemento n_{en} , da matriz de função de forma do

elemento $N^e(x, y)$, $d^e = [T_1^e \ T_2^e \ \dots \ T_{n_{en}}^e]^T$ é a matriz de temperatura do elemento e $w^e = [w_1^e \ w_2^e \ \dots \ w_{n_{en}}^e]^T$ é a matriz dos valores nodais das funções peso dos elementos.

$$T(x, y) \approx T^e(x, y) = N^e(x, y)d^e = \sum_{l=1}^{n_{en}} N_l^e(x, y)T_l^e \quad (x, y) \in \Omega^e \quad (4.17)$$

$$w^T(x, y) \approx w^{eT}(x, y) = N^e(x, y)w^e = \sum_{l=1}^{n_{en}} N_l^e(x, y)w_l^e \quad (x, y) \in \Omega^e \quad (4.18)$$

O gradiente de temperaturas é dado por:

$$\nabla T^e = \begin{bmatrix} \frac{\partial T^e}{\partial x} \\ \frac{\partial T^e}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} T_1^e + \frac{\partial N_2^e}{\partial x} T_2^e + \dots + \frac{\partial N_{n_{en}}^e}{\partial x} T_{n_{en}}^e \\ \frac{\partial N_1^e}{\partial y} T_1^e + \frac{\partial N_2^e}{\partial y} T_2^e + \dots + \frac{\partial N_{n_{en}}^e}{\partial y} T_{n_{en}}^e \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial x} & \dots & \frac{\partial N_{n_{en}}^e}{\partial x} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_2^e}{\partial y} & \dots & \frac{\partial N_{n_{en}}^e}{\partial y} \end{bmatrix} d^e \quad (4.19)$$

Ou em forma compacta:

$$\nabla T^e(x, y) = (\nabla N^e(x, y))d^e = B^e(x, y)d^e \quad (4.20)$$

A matriz de rigidez de condutividade do elemento é definida por sua vez de forma similar à maneira com que foi feita na seção 2.4.4, e vale:

$$K^e = \int_{\Omega} B^{eT} D^e B^e d\Omega \quad (4.21)$$

4.3. Formulação do Problema

Considere um domínio discreto de tamanho finito, Ω , no qual calor está sendo uniformemente gerado a cada discretização, $d\Omega$. Ele está sendo resfriado através de um dissipador de calor posicionado no seu contorno e cuja temperatura é conhecida e definida com sendo igual a zero. O domínio está sujeito a uma fração volumétrica, isto é, um volume finito de material de alta condutividade térmica, k_{max} , está disponível, e onde ele não estiver presente, será usado um material com baixa condutividade térmica, k_{min} . A presença ou ausência do

material de alta condutividade não afetará o calor sendo gerado ao longo do domínio. A distribuição do material de alta condutividade k_{max} sob as restrições de quantidade de material (fração volumétrica) é obtida com a minimização da função objetivo c , para minimizar a temperatura média do domínio de projeto sujeito à geração de calor constante e uniforme [41].

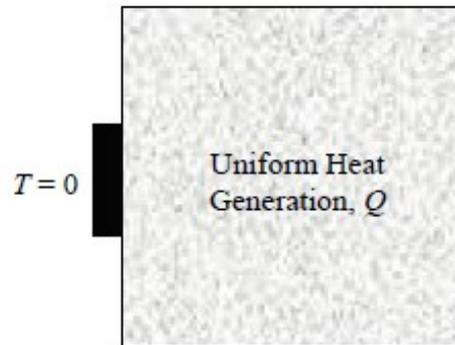


Figura 4.4 – Domínio de projeto 2D para a otimização topológica da condução de calor pura. Fonte: [21]

Assim como em muitos exemplos da literatura, não será considerada os efeitos da convecção nos contornos convectivos do dissipador de calor devido à complexidade adicional que ele traz. O problema é posto para permitir apenas valores discretos (k_{max} e k_{min}) em cada discretização do domínio e se tratando de uma OT baseada em densidade, que prevê resultados da forma de densidades 0-1, a interpolação material é utilizada para converter o problema dos valores discretos de k_{max} e k_{max} para o problema com variáveis contínuas ρ_{max} e ρ_{min} [41].

Considerando o domínio de projeto da *Figura 4.4* homogeneamente aquecido e com temperatura nula no contorno do dissipador de calar, a condução de calor através do domínio é representado por:

$$\nabla \cdot (k\nabla T) + Q = 0 \text{ em } \Omega \quad (4.22)$$

$$T = 0 \text{ em } \Gamma$$

$$R(x) \geq R_{min}$$

Onde T é a variável de estado de temperatura, Q é o calor volumétrico gerado e k é a condutividade térmica do material no domínio Ω .

O problema de otimização topológica é a minimização da função objetivo c , matematicamente escrita como:

$$\min_x: \quad c(x) = Q^T T = \int \nabla T (k \nabla T) d\Omega = \int k (\nabla T)^2 d\Omega \quad (4.23)$$

$$\text{Sujeito a:} \quad KT = Q$$

$$\frac{V(x)}{V_0} = f$$

$$0 < x_{min} \leq x \leq 1$$

$$R(x) \geq R_{min}$$

onde $V(x)$ é a quantidade de material de condutividade térmica elevada e é restringida a uma fração volumétrica f , muito similar à forma como foi feito na equação (3.1). $R(x)$ é o raio do caminho condutivo que deve sempre estar acima do espaçamento mínimo R_{min} , condição que permite ao projeto ser de fato fabricável [42].

Abaixo são apresentadas as modificações feitas aos algoritmos PolyTop e PolyScript de Talischi et al.[3], indicando à esquerda o número da linha de código e indicando com a cor vermelha as linhas de código retiradas e em verde as linhas que vieram a substituí-las:

PolyTop:

código base vs. código modificado.

	<code>function [z,V,fem] = PolyTop(fem,opt)</code>
7	<code>function [z,V,fem] = PolyTopHeat(fem,opt)</code>
	<code>fem.ElemNDof = 2*cellfun(@length,fem.Element); % # of DOFs per element</code>
63	<code>fem.ElemNDof = 1*cellfun(@length,fem.Element); % # of DOFs per element</code>
	<code>eDof = reshape([2*fem.Element{e1}-1;2*fem.Element{e1}],NDof,1);</code>
71	<code>eDof = fem.Element{e1}';</code>

```

NLoad = size(fem.Load,1);
fem.F = zeros(2*fem.NNode,1); %external load vector
fem.F(2*fem.Load(1:NLoad,1)-1) = fem.Load(1:NLoad,2); %x-crdnt
fem.F(2*fem.Load(1:NLoad,1)) = fem.Load(1:NLoad,3); %y-crdnt
NSupp = size(fem.Supp,1);
FixedDofs = [fem.Supp(1:NSupp,2).*(2*fem.Supp(1:NSupp,1)-1);
fem.Supp(1:NSupp,3).*(2*fem.Supp(1:NSupp,1))];
FixedDofs = FixedDofs(FixedDofs>0);
AllDofs = [1:2*fem.NNode];
79 fem.F = zeros(fem.NNode,1); %external load vector
80 fem.F = fem.Load(:,2);
81 FixedDofs = fem.Supp;
82 AllDofs = 1:fem.NNode;

U = zeros(2*fem.NNode,1);
87 U = zeros(fem.NNode,1);

%----- ELEMENT STIFFNESS MATRIX
89 %----- HEAT ELEMENT STIFFNESS MATRIX

D=fem.EO/(1-fem.NuO^2)*[1 fem.NuO 0;fem.NuO 1 0;0 0 (1-fem.NuO)/2];
nn=length(eNode); Ke=zeros(2*nn,2*nn);
91 D=fem.kO*[1,0;0,1]; % isotropic
92 nn=length(eNode); Ke=zeros(nn,nn);

B = zeros(3,2*nn);
B(1,1:2:2*nn) = dNdx(:,1)';
B(2,2:2:2*nn) = dNdx(:,2)';
B(3,1:2:2*nn) = dNdx(:,2)';
B(3,2:2:2*nn) = dNdx(:,1)';
98 B = zeros(2,nn);
99 B(1,1:nn) = dNdx(:,1)';
100 B(2,1:nn) = dNdx(:,2)';

```

PolyScript:

código base vs. código modificado.

```

[Node,Element,Supp,Load] = PolyMesher(@MbbDomain,5000,30);
9 [Node,Element,Supp,Load] = PolyMesher(@SquareDomain,6400,100);

'NuO',0.3,... % Poisson's ratio of solid material
'EO',1.0,... % Young's modulus of solid material
'Reg',0 ... % Tag for regular meshes
17 'Reg',0, ... % Tag for regular meshes
18 'kO',5 ... % Conductivity of solid material

[opt.zIni,V,fem] = PolyTop(fem,opt);
43 [opt.zIni,V,fem] = PolyTopHeat(fem,opt);

```

5. Resultados

O código `Toph` foi simulado para diferentes números de elementos e frações volumétricas, mantendo-se constantes, e iguais a literatura, os valores do raio mínimo do filtro (r_{min}) e expoente de penalização ($penal$) do SIMP e comparado com a *Figura 3.9* fornecida por Sigmund [2].

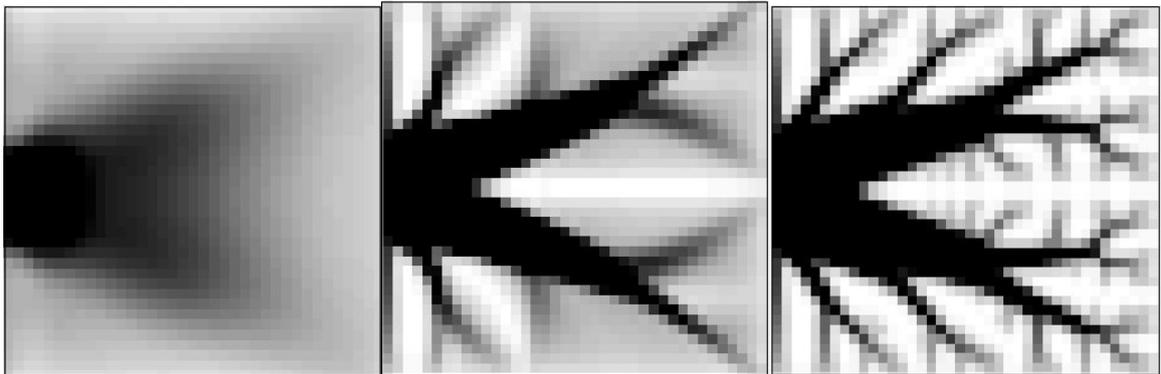


Figura 5.1 – Simulação do código de 91 linhas para OT de problemas de condução de calor.

Chamada da função: `toph(40,40,0.4,3.0,1.2)`



Figura 5.2 – Simulação do código de 91 linhas para OT de problemas de condução de calor.

Chamada da função: `toph(40,40,0.5,3.0,1.2)`

Foi dado como ponto de partida a chamada da função sugerida [2] com 40 elementos na direção horizontal e 40 na direção vertical, fração volumétrica permitida fixada em 0.4, *penalization power* igual a 3 e raio mínimo igual a 1.2. O resultado obtido em três estágios

distintos da discretização pode ser visto na *Figura 5.1*. O domínio otimizado obtido é exatamente igual à *Figura 3.9* o que indica que o código estava funcionando da maneira esperada. Na *Figura 5.2* a simulação foi novamente rodada, porém aumentando de 40% para 50% a fração volumétrica do material de alta condutividade, afim de se observar se existiria diferenças nos caminhos condutores ou na figura otimizada final. Como esperado, ao restringir menos a quantidade de material, os caminhos condutivos tornam-se maiores e ocupam mais o domínio.

As *Figuras 5.3 e 5.4* foram obtidas aumentando o número de elementos das direções horizontal e vertical para 80 e as frações volumétricas fixadas em 0.4 e 0.5, respectivamente. Nota-se que o aumento do número de elementos da malha acaba por melhorar o resultado, tornando os contornos da superfície otimizada mais suaves e fluidos.

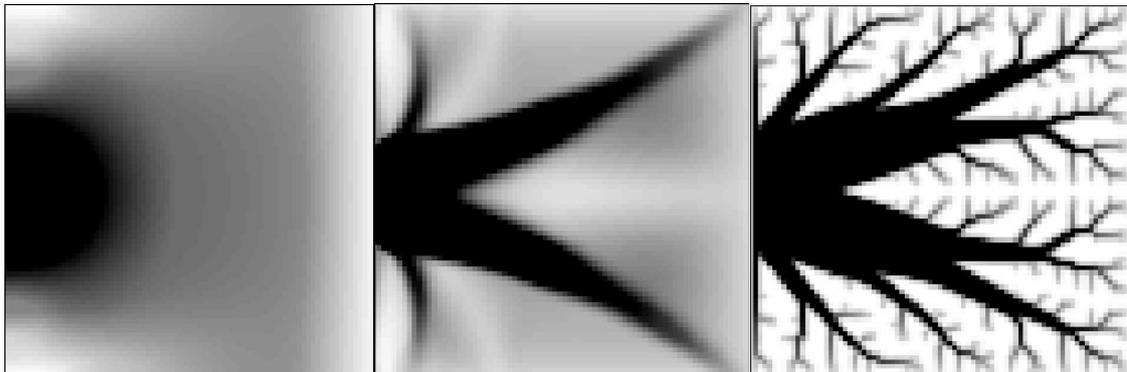


Figura 5.3 – Simulação do código de 91 linhas para OT de problemas de condução de calor.

Chamada da função: `toph(80,80,0.4,3.0,1.2)`

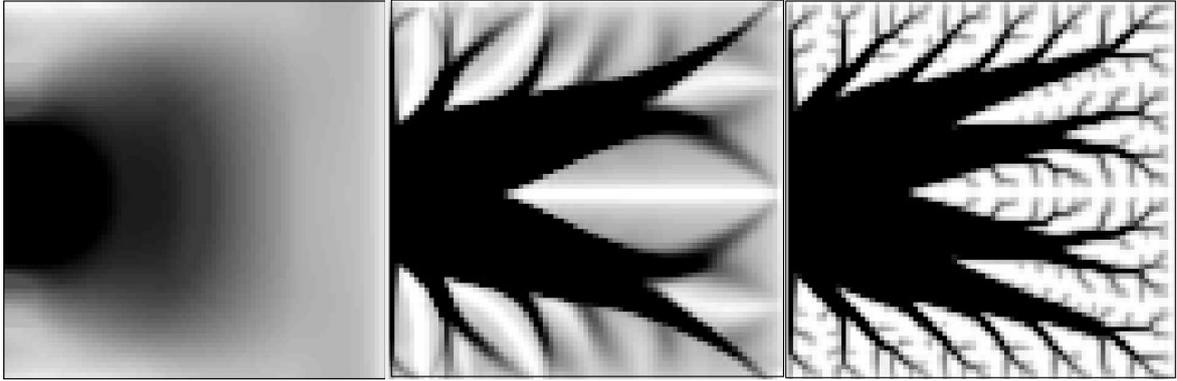


Figura 5.4 – Simulação do código de 91 linhas para OT de problemas de condução de calor.

Chamada da função: $toph(80,80,0.5,3.0,1.2)$

Para gerar as *Figuras 5.5 e 5.6*, o número de elementos nas direções horizontal e vertical foi aumentado novamente, para 100 elementos em cada direção. Assim como nos casos anteriores, as frações volumétricas fixadas em 0.4 e 0.5, respectivamente.

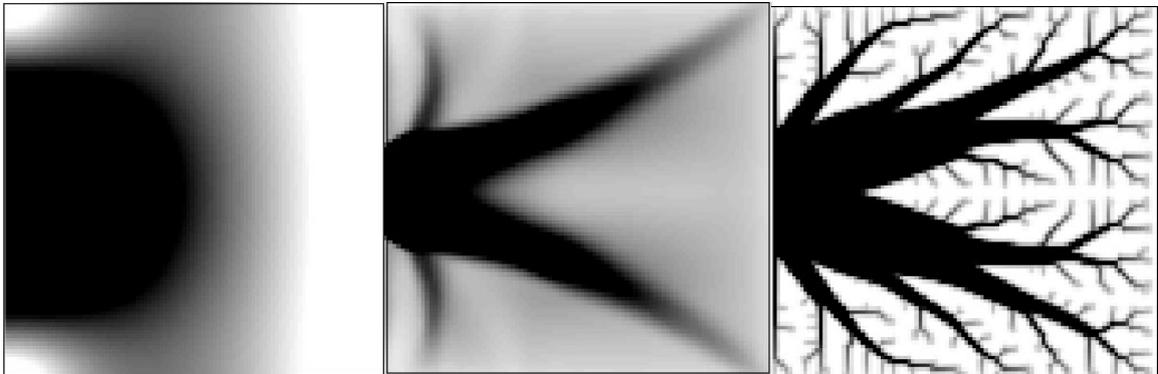


Figura 5.5 – Simulação do código de 91 linhas para OT de problemas de condução de calor.

Chamada da função: $toph(100,100,0.4,3.0,1.2)$

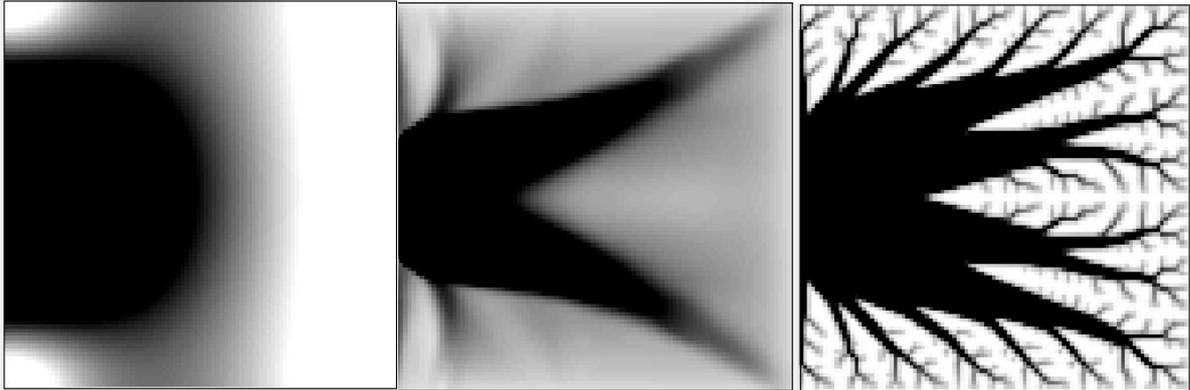


Figura 5.6 – Simulação do código de 91 linhas para OT de problemas de condução de calor.

Chamada da função: `toph(100,100,0.5,3.0,1.2)`

De posse das rotinas originais para a OT de elementos finitos de malha não estruturada de Talischi et al.[3], foram feitas todas as modificações pertinentes para a adequação do código para a condução de calor, tomando como base o problema introdutório de condução de calor apresentado por Bendsøe e Sigmund [2]. O resultado foram três novos códigos; dois códigos principais para a OT, `PolyTopHeat` e `PolyScriptHeat`, e um código para a representação do domínio de projeto, `SquareDomain`. Não houve a necessidade de alterar as sub-rotinas de geração de malhas, `PolyMesher`, de interpolação de material, `MatIntFnc`, ou de função de filtro, `PolyFilter`; validando a usabilidade e capacidade de adequação do código original para variados problemas argumentada pelos seus criadores [3].

Buscando validar o código `PolyTopHeat` e `PolyScriptHeat`, adaptados para a condução de calor isotrópica e homogênea no domínio `SquareDomain`, foram realizadas simulações para confrontar e investigar os resultados. O algoritmo `SquareDomain` é apresentado abaixo:

```

%----- PolyMesher -----%
% Ref: C Talischi, GH Paulino, A Pereira, IFM Menezes, "PolyMesher: A %
%   general-purpose mesh generator for polygonal elements written in %
%   Matlab," Struct Multidisc Optim, DOI 10.1007/s00158-011-0706-z %
%-----%
function [x] = SquareDomain(Demand,Arg)
    BdBox = [0 1 0 1]; % xmin, xmax, ymin, ymax
    switch (Demand)

```

```

    case('Dist'); x = DistFnc(Arg,BdBox);
    case('BC');   x = BndryCnds(Arg{:},BdBox);
    case('BdBox'); x = BdBox;
    case('PFix'); x = FixedPoints(BdBox);
end
%----- COMPUTE DISTANCE FUNCTIONS
function Dist = DistFnc(P,BdBox)
    Dist = dRectangle(P,BdBox(1),BdBox(2),BdBox(3),BdBox(4));
%----- SPECIFY BOUNDARY CONDITIONS
function [x] = BndryCnds(Node,Element,BdBox)
    eps = 0.1*sqrt((BdBox(2)-BdBox(1))*(BdBox(4)-BdBox(3))/size(Node,1));
    FixedNodes = find(abs(Node(:,1)-BdBox(1))<eps & ...
        (Node(:,2)<2/3 & Node(:,2)>1/3));
    Supp = zeros(length(FixedNodes),1);
    Supp(:,1)=FixedNodes;
    Load = zeros(length(Node),2);
    Load(:,1) = 1:length(Node);
    Load(:,2) = 0.01;
    x = {Supp,Load};
%----- SPECIFY FIXED POINTS
function [PFix] = FixedPoints(BdBox)
    PFix = [];
%-----

```

Ao contrário do código enxuto de 91 linhas para a OT de condução de calor onde se passa como parâmetro na chamada da função o número de elementos nas direções horizontal e vertical, o algoritmo de Talischi et al.[3] prevê que sejam escolhidos o domínio e o número de elementos total da malha, porém, vale ressaltar que agora a malha é não estruturada e será preenchida por elementos poligonais. Para exemplificar são apresentados abaixo duas malhas não estruturadas, gerados pelo função-domínio `SquareDomain` com calor distribuído por todo o meio. Os nós são selecionados em roxo enquanto os pontos fixos correspondente ao dissipador de calor ($T=0$) são selecionados em azul.

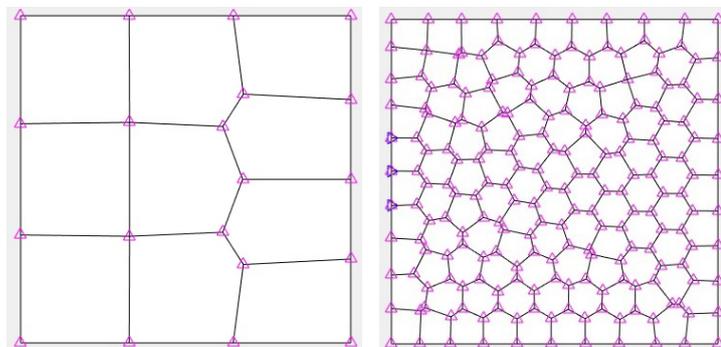


Figura 5.7 – *SquareDomain* com malha não estruturada de 10 (à esquerda) e 100 elementos poligonais (à direita).

Para que haja alguma forma de comparar a OT do problema de condução de calor em duas rotinas distintas como é pretendido aqui, as simulações foram feitas com 1600, 6400 e 10000 elementos, o que seria próximo ao número de elementos totais em uma malha estruturada gerada pelo produto de 40x40, 80x80 e 100x100 elementos nas direções horizontal e vertical, respectivamente.

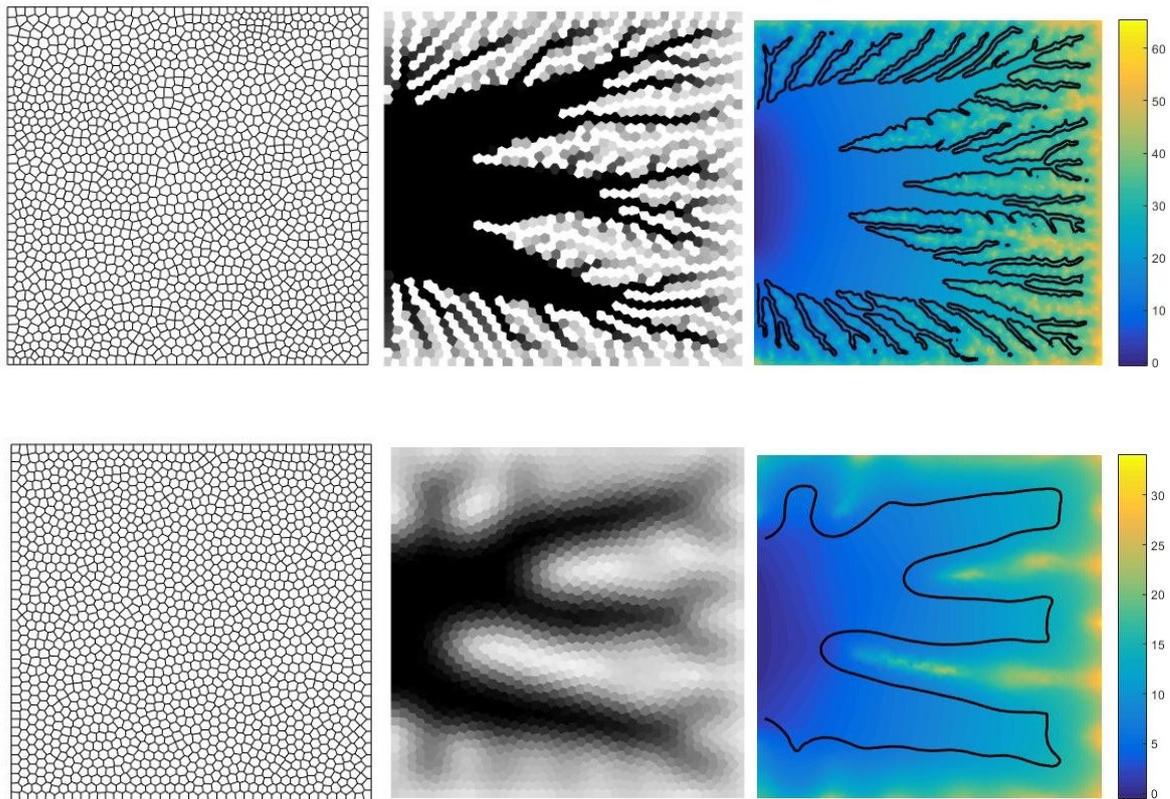


Figura 5.8 – *PolyScriptHeat(@SquareDomain,1600,100)*. Da esquerda para a direita: malha com 1600 elementos, a distribuição de densidades (caminho ótimo para a condução de calor) e a distribuição de temperaturas. Simulação feita com 0.5 de fração volumétrica. Primeira fileira com raio do filtro de $R=0.01$ e a segunda com $R=0.1$

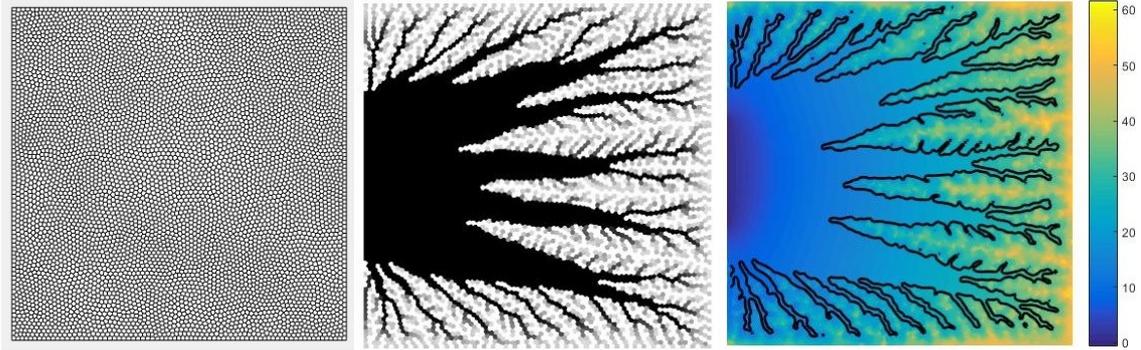


Figura 5.9 – *PolyScriptHeat(@SquareDomain,6400,100)*. Da esquerda para a direita: malha com 6400 elementos, a distribuição de densidades (caminho ótimo para a condução de calor) e a distribuição de temperaturas. Simulação feita com 0.5 de fração volumétrica e raio do filtro de $R=0.01$.

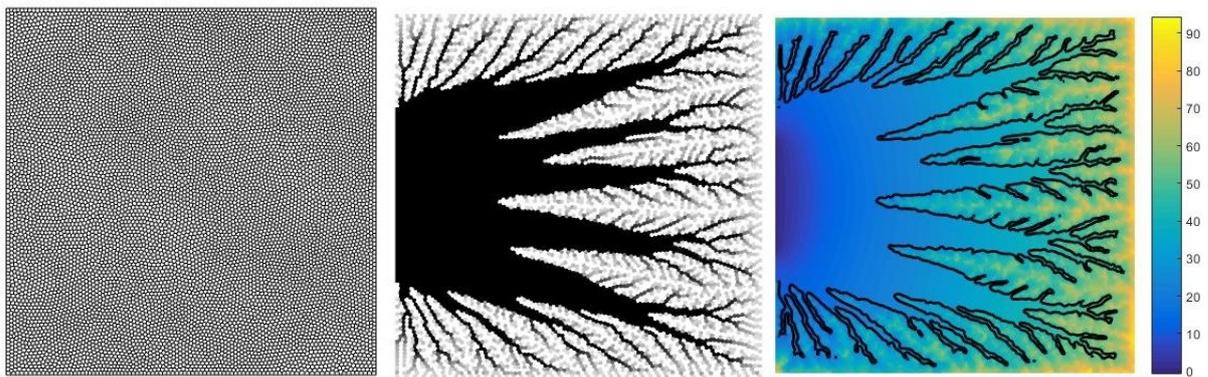


Figura 5.10 – *PolyScriptHeat(@SquareDomain,10000,100)*. Da esquerda para a direita: malha com 10000 elementos, a distribuição de densidades (caminho ótimo para a condução de calor) e a distribuição de temperaturas. Simulação feita com 0.5 de fração volumétrica e raio do filtro de $R=0.01$.

As Figuras 5.9, 5.10 e 5.11 foram geradas ao aumentar o número de elementos de 1600 para 6400 e por último com 10000. Fica evidente através da primeira imagem (mais à esquerda) de cada figura que a malha está assumindo cada vez mais discretizações, o que contribui para soluções melhores. A fração volumétrica de material condutor para o domínio de projeto foi de 0.5 ou 50%, e foi feita a interpolação material com o método SIMP. Os raios do filtro são

utilizados na função auxiliar `PolyFilter` do algoritmo que computa a distância entre centroides dos elementos da malha e define o peso dado ao filtro, sendo valores muito baixos retornam à matriz identidade. Um de suas funções principais é evitar os efeitos da instabilidade de tabuleiro discutidos na seção 4.1.

Em seguida se simulou repetidas vezes o mesmo problema considerando mesmo número de iterações, elementos e fração volumétrica, foi alterado apenas o raio do filtro para procurar entender seu comportamento.

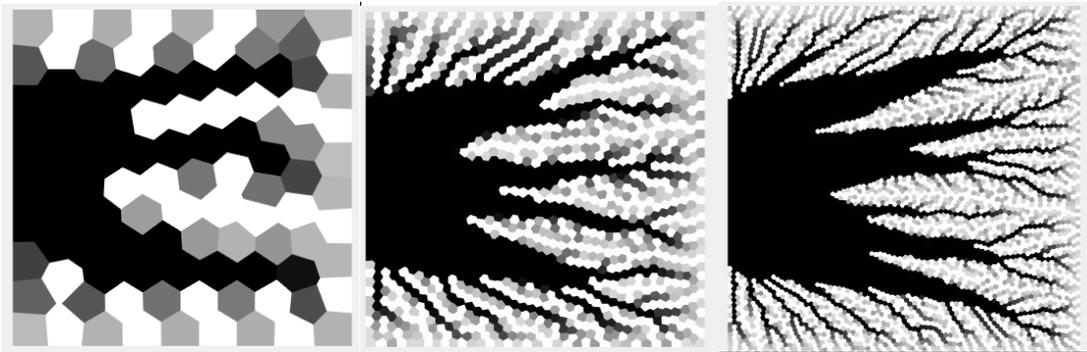


Figura 5.11 – OT de malha não estruturada para a condução de calor; da esquerda para a direita: malha com 100 elementos, 1600 elementos e 6400 elementos. Raio do filtro $R=0.01$.

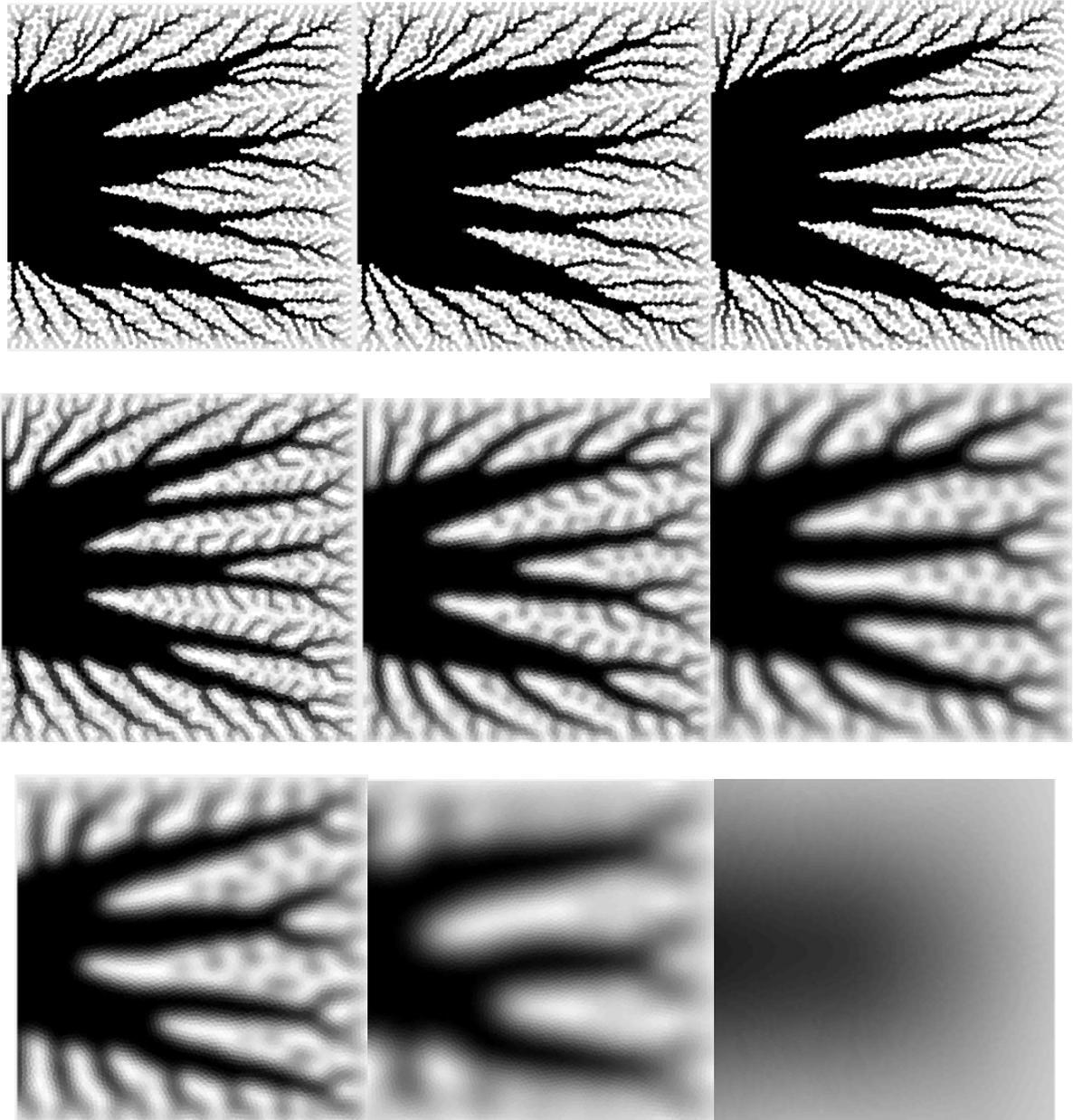


Figura 5.12 – Influência do raio do filtro na OT de malha não estruturada para a condução de calor, com fração volumétrica de 0.5, 6400 elementos e raio do filtro variando entre 0.0001-0.5. Primeira fileira (da esquerda para a direita): $R=0.0001$, $R=0.001$ e $R=0.01$. Segunda fileira (da esquerda para a direita): $R=0.02$, $R=0.03$, $R=0.04$. Última fileira (da esquerda para a direita): $R=0.05$, $R=0.1$ e $R=0.5$.

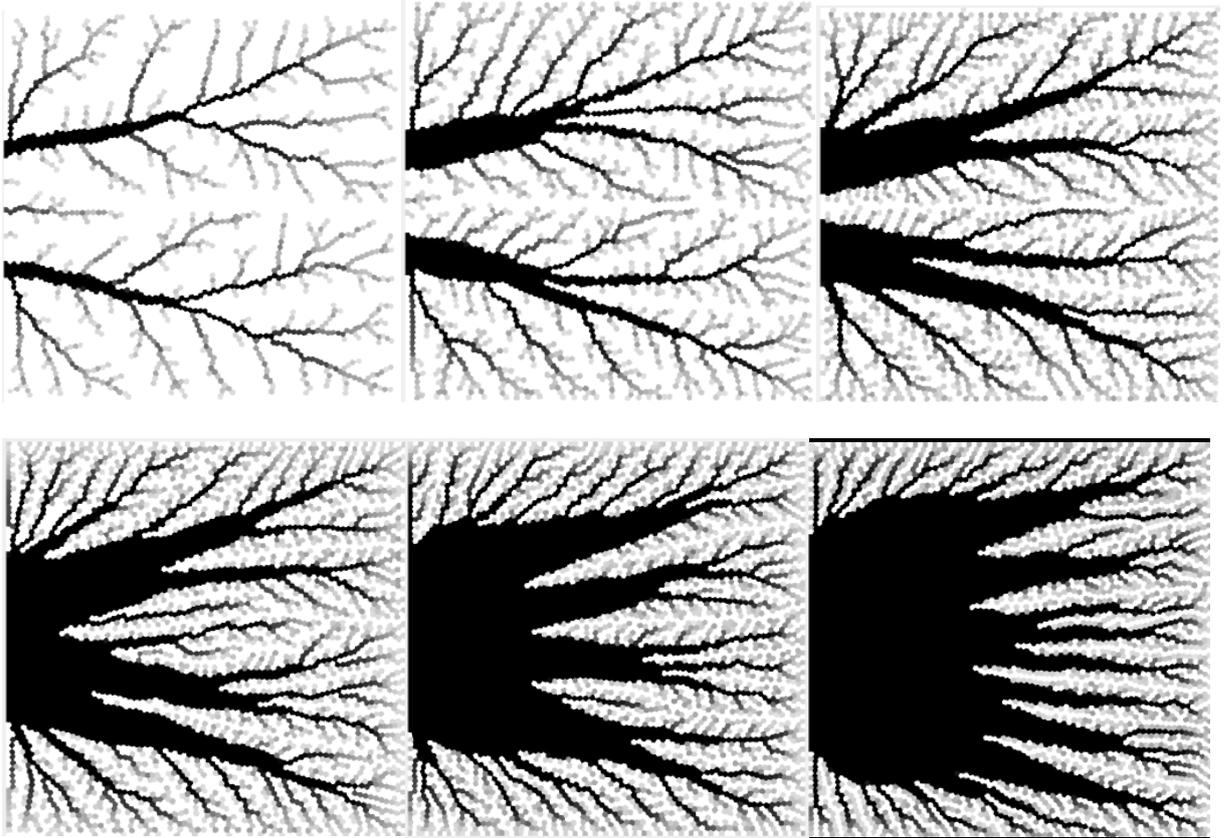


Figura 5.13 – Influência da fração volumétrica na OT de malha não estruturada para a condução de calor, com 6400 elementos, raio do filtro $R=0.01$ e fração volumétrica variando entre 0.1-0.6. Primeira fileira (da esquerda para a direita): *VolFrac:0.1*, *VolFrac:0.2* e *VolFrac:0.3*. Segunda fileira (da esquerda para a direita): *VolFrac:0.4*, *VolFrac:0.5* e *VolFrac:0.6*.

6. Conclusão

Sigmund expressa explicitamente que a sua intenção era apresentar um algoritmo de programação matemática de otimização topológica simples e intencionalmente mantido abaixo de 100 linhas de código. Desde então, o código foi baixado inúmeras vezes e utilizado por muitos professores ao redor do mundo para cursos de otimização estrutural, obtendo sempre, um *feedback* muito positivo [1]. Levando isso em consideração, sua importância para o estudo inicial sobre o tema de OT, não pode ser descartada. O autor não se omite para as ineficiências de processamento e custo operacional superior em comparação a uma implementação em Fontran, porém, frisa que ganhos computacionais são obtidos ao se abrir mão da compatibilidade do código. A adaptação do código de Sigmund para a OT em problemas de condução de calor se mostrou bastante simples e manteve o algoritmo compacto, porém exibiu claras limitações [2]. O algoritmo `PolyTop` de Talischi et al.[3] por sua vez, apresentou claras vantagens em comparação a códigos compactos de malha estruturada, com performance igualável a de códigos mais enxutos como o de 88 linhas publicado por Andreassen em 2010, e sendo mais rápido em malhas maiores [3]. Como era de se esperar, a estruturação e formulação do algoritmo `PolyTop` foi crucial para que este pudesse ser tão facilmente adaptado à condução de calor quanto o código de 99 linhas, estendendo sua importância para um software educacional assim como os anteriores.

O primeiro contato com o uso de malhas não estruturadas arbitrárias que esse trabalho permitiu, também deve ser colocado em evidência visto que muitas das aplicações de engenharia requerem o uso de malhas não estruturadas para a completa e precisa representação da geometria do domínio de projeto, especificações de carregamento e suporte e uma resultante análise confiável. O suposto custo computacional mais elevado de malhas não estruturadas que seriam utilizados como motivo para adoção de malhas estruturadas são apontadas por Talischi et al. como prováveis causas de repetidos cálculos da matriz de rigidez dos elementos,

eliminadas do código `PolyTop` ao se adotar meios de guardar tais informações em memória para as próximas iterações de otimização [3].

Os resultados obtidos através das simulações, se comportam como previsto em literatura, exibindo ramificações que lembram galhos de árvores na otimização topológica em duas dimensões [42]. Para aplicações de troca de calor foi reconhecida a tendência de a topologia ótima adotar padrões que se assemelham padrões de dendritos e minimizar a resistência térmica entre a fonte e o dissipador de calor.

Pode ser defendido que a verdadeira otimização ótima para problemas de resfriamento ainda não foi encontrada, visto que o estudo e aprendizado no campo dos materiais, tecnologias, projeto e manufatura de componentes é contínuo e segue numa busca contínua por melhorias [41]. Os estudos dos caminhos condutores de calor continuará sendo uma importante área de estudo dada a sua relevância para os projetos de canais térmicos condutivos para microcomponentes eletrônicos, cuja performance e confiabilidade estão relacionadas as variações de temperaturas no substrato no qual estão montados e as tensões térmicas geradas.

Existe muito espaço para futuras melhorias, seja no código (geração de novos domínios) ou na abordagem do problema. Todas as simulações foram feitas no `SquareDomain` para se manter fiel ao problema enunciado por Sigmund [1,2] e constantemente representado na literatura como problema introdutório de OT em problemas de condução de calor. Todavia, o código elaborado nesse trabalho poderia ter sido validado em outros domínios, com posicionamento distinto do dissipador de calor, com diferentes condições de contorno e em diferentes meios, além do isotrópico e homogêneo.

Referências

- [1] Sigmund, O. 2001: A 99 line topology optimization code written in Matlab. *Struct. Optim.* 21, 120-127.
- [2] Bendsøe & Sigmund (2003) – Topology Optimization: Theory, Methods and Applications
- [3] TALISCHI, C.; PAULINO, G. H.; PEREIRA, A.; MENEZES, I. F. M. PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Struct. Multidiscip. Optim.*, V.45, n.3, p. 329-357, 2012.
- [4] Criteria for Accrediting Programs in Engineering in the United States, Section IV.C.2. Effective for Evaluation During the 1995-96 Accreditation Cycle, Engineering Accreditation Commission, Accreditation Board for Engineering and Technology, Inc. 111 Market Place, Suite 1050, Baltimore, Maryland 21202.
- [5] FOLEY, A. M.; LEAHY, P. G. The Role of the Professional Engineer in the 21st Century. Environmental Research Institute, University College Cork, Lee Rd., Cork, Ireland. Dept. of Civil & Environmental Engineering, School of Engineering, University College Cork, College Rd., Cork, Ireland. 3rd International Symposium for Engineering Education, 2010, University College Cork, Ireland
- [6] CLOUGH, R. W. The finite element in plane stress analysis. Proc. 2nd ASCE Conf. on Electronic Computation. Pittsburgh, Pa., Sept. 1960.
- [7] Jacob Fish, Ted Belytschko-A First Course in Finite Elements [With CDROM]-John Wiley & Sons Ltd (2007)
- [8] B.Y. Lee, Shape sensitivity formulation for an axisymmetric thermal conducting solids, Proceedings of Institution of Mechanical Engineers Part C: Journal of Mechanical Engineering Science 207 (C3) (1993) 209-216.
- [9] M. Sasikumar, C. Balaji, Optimization of convective fin systems: a holistic approach, *Int. J. Heat Mass Transfer.* 39 (2002) 57-68.
- [10] D.K. Yang, K.S. Lee, S. Song, Fin spacing optimization of a fin-tube heat exchanger under frosting conditions, *Int. J. Heat Mass Transfer,* 49 (2006) 2619-2625.
- [11] K.T. Zuo, L.P. Chen, Y.Q. Zhang, S.T. Wang, Structural optimal design of heat conductive body with topology optimization method, *China J. Mech. Eng.* 41 (2005).
- [12] C.A. Felippa, Introduction to Finite Element Methods: Notas de Aula da disciplina Introduction to Finite Elements Methods (ASEN 5007) - Fall 2016, Department of Aerospace Engineering Sciences, University of Colorado at Boulder, 2016.
- [13] Turner, M.J., Clough, R.W., Martin, H.C. and Topp, L.J. (1956) Stiffness and deflection analysis of complex structures. *J. Aeronaut. Sci.*, 23, 805–23.
- [14] Merrill, C., Custer, R., Daugherty, J., Westrick, M., & Zeng, Y. (2007). Delivering core engineering concepts to secondary level students. American Society for Engineering Education Conference Paper.
- [15] Kelley, T. R. (2010). Optimization, an important stage of engineering design. *The Technology Teacher*, 69(5), 18-23.
- [16] Maxwell, C., “Scientific Paper II”, Cambridge University Press, 1890

- [17] Michell, A. G. M., “The Limits of Economy of Material in Frame-Structures”, *Philosophical Magazine, Series 6, Vol. 8, No. 47, 1904*
- [18] Bendsoe, M.P., Kikuchi, N. (1988). Generating optimal topologies for structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71(2): 197-224.
- [19] Liu, K., Tovar, A. An efficient 3D topology optimization code written in Matlab. *Struct Multidisc Optim* (2014) 50:1175-1196.
- [20] Li, Q., Steven, G. P., Querin, O. M., Xie, Y. M. Shape and topology design for heat conduction by Evolutionary Structural Optimization. *International Journal of Heat and Mass Transfer* 42 (1999) 3361-3371.
- [21] Dede, E. M. Multiphysics Topology Optimization of Heat Transfer and Fluid Flow Systems. *Proceedings of the COMSOL Conference 2009 Boston*.
- [22] Incropera, F. P., Dewitt, D.P. (2011). *Fundamentals of Heat and Mass Transfer*. New York: J.Wiley.
- [23] CIV 2118 – Metodo dos Elementos Finitos – Modelo dos Elementos Finitos para analise estrutural – Marcelo Gattass – Puc-Rio.
- [24] CEFET – Introdução ao Método dos Elementos Finitos.pdf
- [25] Souza, R. M. (2003). *O Método dos Elementos Finitos Aplicado ao Problema de Condução de Calor*. Universidade Federal do Pará.
- [26] Miranda, A. Apostilas de MEF. Cap.5 – Elementos Finitos Isoparamétricos. Pontificia Universidade Católica do Rio de Janeiro. Dept. de Engenharia Civil.
- [27] SOM/UIUC Collaboration – Short Course on Topology Optimization – February 3, 2009
- [28] Thedin, R. S. (2014). *Topology Optimization using Polyhedral Meshes*. Master thesis, Pontifical Catholic University of Rio de Janeiro.
- [29] Junior, P. C. V. (2014). *Análise Level Set da Otimização Topológica de Estruturas Planas Utilizando o Método dos Elementos de Contorno*. Dissertação de mestrado, Escola de Engenharia de São Carlos, da Universidade de São Paulo.
- [30] Bathe, K. J. *Finite Element Procedures*. Prentice Hall, 1996
- [31] Hughes, T. J. R. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.
- [32] Meirovitch, L. *Principles and Techniques of Vibrations*. Prentice Hall, 1997
- [33] Strang, G., and Fix, G. J. *An Analyses of the Finite Element Method*. Prentice Hall, 1973.
- [34] Dhatt, G., and Touzot, G. *Une présentation de la méthode des éléments finis*. Prentice Hall, 1984.
- [35] Zienkiewicz, C. O., and Taylor, R. L. *The Finite Element Method for Solid and Structural Mechanics*. Elsevier, 2005.
- [36] Bendsoe, M.P. 1989: Optimal shape design as a material distribution problem. *Struct. Optim.* 1, 193-202.
- [37] Mlejnek, H.P. 1992: Some aspects of the genesis of structures. *Struct. Optim.* 5, 64-69
- [38] Sigmund, O.; Petersson, J. 1998: Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Optim.* 16, 68-75.

- [39] Zhou, M.; Rozvany, G.I.N. 1991: The COC algorithm, part II: Topological, geometry and generalized shape optimization. *Comp. Meth. Appl. Mech. Engrng.* 89, 197-224.
- [40] Silva, E. C. N., Notas de Aula: Otimização aplicada ao projeto de sistemas mecânicos. Universidade de São Paulo – Escola Politécnica. 2003.
- [41] Manuel, M. C. E., Lin, P. T., Design explorations of heat conductive pathways. *International Journal of Heat and Mass Transfer* 104 (2017) 835-851
- [42] Lohan, D. J., Dede, E. M., Allison, J. T. 2016. Topology optimization for heat conduction using generative design algorithms. *Struct. Optim*
- [43] Sukumar, N., Tabarraei, A. Conforming polygonal finite elements. *Int. J. Numer. Meth. Engng* 2004; 61:2045-2066. DOI:10.1002/nme.1141