

Universidade Federal de Santa Catarina
Curso de Pós - Graduação em Engenharia Mecânica

**Procedimento de Otimização Estrutural de
Componentes Usando um Programa
Comercial de Elementos Finitos**

Dissertação submetida à Universidade Federal de Santa Catarina
para obtenção do grau de Mestre em Engenharia Mecânica

Kleber José Zanini Legnani

Florianópolis, 18 de Agosto de 2000.

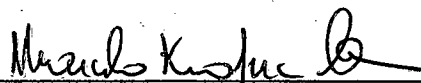
Procedimento de Otimização Estrutural de Componentes Usando um Programa Comercial de Elementos Finitos

Kleber José Zanini Legnani

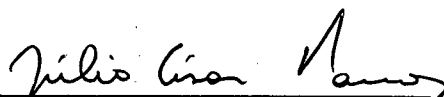
Esta Dissertação foi julgada adequada para obtenção do título de

Mestre em Engenharia Mecânica

Especialidade Engenharia Mecânica, área de concentração Projeto e Análise de Componentes Mecânicos, e aprovada em sua forma final pelo curso de Pós Graduação em Engenharia Mecânica



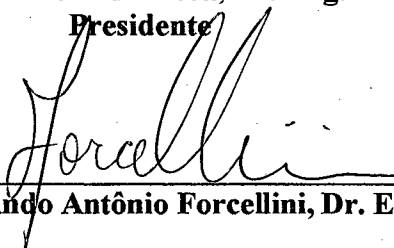
Prof. Marcelo Krajnc Alves, Ph.D.
Orientador



Prof. Júlio César Passos, Dr. Sc.
Coordenador do Curso



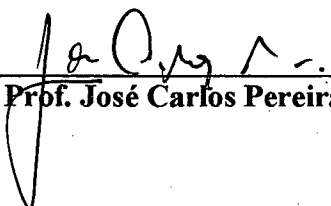
Prof. Edison da Rosa, Dr. Eng.
Presidente



Prof. Fernando Antônio Forcellini, Dr. Eng.



Prof. Carlos Henrique Ahrens, Dr. Eng.



Prof. José Carlos Pereira, Dr.

Dedicatória

Em primeiro lugar a Deus pela vida.

*À minha esposa Liliane e meu filho Arthur
pelo amor e paciência ao longo desta jornada.*

*Aos meus pais José e Sirlei pelo incentivo e
carinho em todos os momentos da minha vida.*

Agradecimentos

É muito importante chegar ao fim de uma jornada e saber reconhecer a ajuda e contribuição das inúmeras pessoas que auxiliaram na conquista deste objetivo. Portanto, é com muita satisfação que agradeço, primeiramente, ao amigo, e incentivador Marcelo Krajnc Alves, o qual deu-me apoio nas horas de desânimo, pelos conhecimentos transmitidos, e sem o qual acredito que não teria a força de vontade de continuar no mestrado e concluir este trabalho.

Quero também deixar o meu muito obrigado aos professores do grante pelos ensinamentos e trocas de experiências, que em muito me auxiliaram no meu desenvolvimento técnico, durante o período que estive nas cadeiras de sala de aula.

A Multibras S.A. Eletrodomésticos, que permitiu-me fazer o curso de pós graduação. A UFSC e o Departamento de Pós Graduação em Engenharia Mecânica pela oportunidade de participar do curso.

Ao amigo Leonardo Romano, que durante sua passagem pela Multibras deu-me incentivo para ingressar no mestrado, e mesmo depois de sua saída da empresa continuou incentivando-me. A ele desejo prosperidade e sucesso na vida.

Durante o curso tive a oportunidade de conhecer e trocar idéias com muitas pessoas e o mais importante foi a amizade feita durante este período, agradeço a todos, principalmente, aos amigos Nederson da Silva Koehler e Valdir Mendes Cardoso, desejo sucesso aos dois. Em especial quero agradecer ao amigo João Carlos Arantes Costa Jr. pelo incentivo e dicas durante o curso, também pela sua hospitalidade quando de minha passagem por Florianópolis.

Enfim se esqueci alguém quero deixar meu agradecimento a todos que de alguma forma me ajudaram a conquistar mais este objetivo.

Sumário

1	<i>Introdução</i>	1
1.1	Apresentação do Trabalho	3
1.2	Revisão Bibliográfica	5
2	<i>Otimização</i>	10
2.1	Introdução	10
2.2	Formulação do Problema de Otimização	11
2.3	Processo Iterativo Genérico para Otimização	14
2.4	Métodos Numéricos	18
2.4.1	Problemas Sem Restrições	19
2.4.1.1	Problemas de Minimização 1-D	20
2.4.1.2	Problemas Sem Restrições N-dimensional	23
2.4.2	Problemas Com Restrição	31
2.4.2.1	Método da Função Penalidade Exterior	31
2.4.2.2	Método do Lagrangeano Aumentado – M.L.A.	35
3	<i>Análise de Sensibilidade</i>	43
3.1	Introdução	43
3.2	O Método de Elementos Finitos	44
3.2.1	Princípios Matemáticos para o MEF	45
3.3	Gradiente da Função Objetivo e Restrições	52
3.3.1	Método Direto	53
3.3.2	Método Adjunto	55
3.3.3	Método Semi Analítico	56
3.3.4	Método de Diferenças Finitas	58
4	<i>Procedimento de Otimização</i>	62
4.1	Introdução	62

4.2	Desenvolvimento Teórico	65
4.2.1	Modelo Geométrico Parametrizado	66
4.2.2	Modelagem do Problema Físico	67
4.2.3	Rotinas de Otimização	69
4.2.4	Integração com Programa Comercial	70
4.3	Abordagem Genérica do Procedimento	70
4.4	Descrição do Programa	72
4.4.1	Aspectos Gerais	72
4.4.2	Preparação do Arquivo de Entrada	75
4.4.3	Descrição das Sub-rotinas	80
4.4.4	Fluxograma das Principais Sub-rotinas	87
4.4.5	Procedimento para Programação	96
5	Resultados	98
5.1	Introdução	98
5.2	Exemplo 1 – Viga Engastada	99
5.2.1	Caso 1 – Otimização de volume	99
5.2.2	Caso 2 – Determinação do módulo de elasticidade	103
5.2.3	Caso 3 - Minimização do trabalho para deflexão de uma viga engastada	106
5.3	Exemplo 2 – Pórtico Bi-dimensional	110
5.4	Exemplo 3 – Otimização de Forma	115
5.5	Exemplo 4 – Caso Prático - Otimização de espessura de prateleira para refrigerador	120
6	Conclusões e Sugestões	128
	Referência Bibliográfica	131

Lista de Figuras

Figura 2.1 - Função unimodal.	11
Figura 2.2 - Fluxograma para processo genérico de otimização.	16
Figura 2.3 - Função unimodal - redução do intervalo de incerteza.	21
Figura 2.4 - Interpretação geométrica do Método " <i>Steepest Descent</i> " - Vanderplaats.	25
Figura 2.5 - Influência do parâmetro de penalização.	34
Figura 2.6 - Fluxograma para o Método do Lagrangeano Aumentado.	42
Figura 3.1 - Sólido genérico e condições de contorno - Belegundu & Chandrupatla.	45
Figura 4.1 - Processo de otimização tradicional.	62
Figura 4.2 - Processo de otimização integrado.	64
Figura 4.3 - Módulos para um procedimento de otimização.	65
Figura 4.4 - Fases da modelagem de um problema de engenharia.	68
Figura 4.5 - Fluxograma genérico do processo de otimização proposto.	74
Figura 4.6 - Fluxograma da rotina Main.	87
Figura 4.7 - Fluxograma da sub-rotina AugLagMethod.	88
Figura 4.8 - Fluxograma da sub-rotina dfpmin.	89
Figura 4.9 - Fluxograma da função Funct(x).	90
Figura 4.10 - Fluxograma da função Fx(x).	91
Figura 4.11 - Fluxograma da sub-rotina Ansys_Run.	92
Figura 4.12 - Fluxograma da sub-rotina Gx(x,g).	93
Figura 4.13 - Fluxograma da sub-rotina GradFunct_Num.	94
Figura 4.14 - Fluxograma da função LineSearch.	95

Figura 5.1 - Modelo da viga engastada.	99
Figura 5.2(a) - Modelo da viga engastada - configuração inicial.	101
Figura 5.2(b) - Deslocamento equivalente [mm] - configuração inicial.	101
Figura 5.2(c) - Distribuição da tensão principal - S1 [MPa] - configuração inicial.	101
Figura 5.3(a) - Modelo da viga engastada - configuração final.	101
Figura 5.3(b) - Deslocamento equivalente [mm] - configuração final.	102
Figura 5.3(c) - Distribuição da tensão principal - S1 [MPa] - configuração final.	102
Figura 5.4 - Convergência da função objetivo (ANSYS x Prog. Acadêmico).	103
Figura 5.5(a) - Convergência da função objetivo.	105
Figura 5.5(b) - Convergência da variável de projeto.	105
Figura 5.6 - Gráfico de convergência para função objetivo e variáveis de projeto.	108
Figura 5.7 - Gráfico do comportamento do volume e restrição de volume.	109
Figura 5.8 - Estrutura pórtico bi-dimensional.	110
Figura 5.9 - Modelo de elementos finitos do pórtico.	111
Figura 5.10(a) - Configuração inicial: modelo.	112
Figura 5.10(b) - Configuração inicial: deformada.	112
Figura 5.10(c) - Configuração inicial: deslocamento equivalente.	112
Figura 5.11(a) - Configuração final: modelo.	112
Figura 5.11(b) - Configuração final: deformada.	112
Figura 5.11(c) - Configuração final: deslocamento equivalente.	112
Figura 5.12 - Gráfico comparativo da convergência entre os métodos de otimização (ANSYS x Procedimento Proposto).	114
Figura 5.13 - Modelo geométrico da viga engastada (EPT).	115
Figura 5.14(a) - Modelo de elementos finitos viga engastada (EPT) - configuração inicial.	117
Figura 5.14(b) - Tensão máxima [psi] - configuração inicial.	117

Figura 5.14(c) - Deflexão máxima [<i>in</i>] - configuração inicial.	117
Figura 5.15 - Modelo de elementos finitos viga engastada (EPT) - configuração final.	117
Figura 5.16(a) - Tensão máxima [<i>psi</i>] - configuração final.	118
Figura 5.16(b) - Deflexão máxima [<i>in</i>] - configuração final.	118
Figura 5.17 - Gráfico comparativo da convergência entre os métodos de otimização (ANSYS x Procedimento Proposto).	119
Figura 5.18 - Modelo geométrico da prateleira de um refrigerador [mm].	120
Figura 5.19 - Modelo de elementos finitos com as condições de contorno.	122
Figura 5.20(a) - Resultado de deflexão [mm] - configuração inicial.	123
Figura 5.20(b) - Resultado de tensões de von Mises [MPa], Top - configuração inicial.	124
Figura 5.21(a) - Resultado de deflexão [mm] - peça otimizada.	125
Figura 5.21(b) - Resultado de tensões de von Mises [MPa], Top - peça otimizada.	125
Figura 5.22 - Curvas de convergência da função objetivo dos métodos do Programa ANSYS (M.S.A e M.P.O.) e Procedimento proposto	126

Lista de Tabelas

Tabela 5.1 - Propriedades do modelo da viga engastada.	99
Tabela 5.2 - Limites para as restrições - exemplo 1.	100
Tabela 5.3 - Resultados: configuração inicial x final.	102
Tabela 5.4 - Resultados: determinação de E	104
Tabela 5.5 - Limites para as restrições - exemplo 1 - caso3.	107
Tabela 5.6 Resultados: otimização do trabalho.	108
Tabela 5.7 - Propriedades: modelo pórtico 2D.	111
Tabela 5.8 - Resultado comparativo entre os métodos de otimização - pórtico 2D.	113
Tabela 5.9 - Propriedades: viga engastada (E.P.T.).	116
Tabela 5.10 - Resultado comparativo entre os métodos de otimização - viga (E.P.T.).	118
Tabela 5.11 - Características do componente - caso prático.	121
Tabela 5.12 - Limites para as restrições - caso prático.	123
Tabela 5.13 - Resultado comparativo entre os métodos de otimização - caso prático.	124

Resumo

O presente trabalho tem como objetivo apresentar o desenvolvimento de um procedimento flexível e de baixo custo, que auxilie o processo de otimização estrutural em um ambiente de projeto.

Para isto foi tomado como premissa o fato de que em muitos ambientes de projeto é comum encontrar uma infinidade de ferramentas computacionais tipo CAD, CAM e CAE. Com relação as ferramentas de CAE, normalmente, estas são pacotes básicos, limitados à análises simples, como por exemplo, estática linear, térmica sem regime transiente e modal, ou ainda programas sofisticados, contudo fechados. Neste sentido, desenvolveu-se um procedimento para otimização de projeto estrutural, integrando a tecnologia existente com algoritmos de otimização.

Em linhas gerais, o procedimento foi desenvolvido a partir da base de elementos finitos já instalada, ou seja, utilizando-se de um programa comercial, como o programa comercial Ansys. Além deste, foram implementados um conjunto de algoritmos para manipulação de arquivos e de otimização.

O processo de otimização visa, entre as múltiplas possibilidades, a minimização da massa de uma estrutura ou componente, a minimização da “*compliance*”, a identificação de propriedades dos materiais (“*Young’s modulus*”). No caso da otimização estrutural de dimensão e de forma, são consideradas restrições, que limitam o domínio de projeto de uma estrutura ou de um componente. Estas limitações são as restrições laterais, impostas sobre as variáveis de projeto, restrições de deslocamentos, e de tensões.

Os exemplos apresentados têm como objetivo comprovar a viabilidade do procedimento proposto e sua aplicação para o projeto ótimo de estruturas e componentes em um ambiente industrial. Os resultados são analisados e comparados, quando possível, com as soluções apresentadas na literatura.

Abstract

The present work has the objective to present the development of a flexible and low cost procedure that assists the process of structural optimization in a design environment.

For this was taken as premise that in many environments of design it is common to find several computational technologies like CAD, CAM and CAE. With respect to CAE tools, normally, these are basic packages, limited to simple analyses, as for example, linear static, non-transient thermal and modal analysis or sophisticated software, however closed. In this sense, a procedure for structural design optimization was developed, integrating the existing technology with optimization algorithms.

In general, the developed procedure makes use of existing finite element program available or commercial codes, such as Ansys. In order to achieve the goals, were implemented a large set of algorithms responsible for the manipulation of files, management of the integrated tool and several optimization methods.

The optimization process aims at, among many possibilities, the minimization of the mass of a structure or component, the minimization of the compliance, the identification of material properties (Young's modulus). In the case of sizing and shape optimization, the constraints considered which limit the design domain are: boundary constrains, displacement constrains, and stress constrains.

The examples presented have the objective to validate the procedure and its application for the optimal design of structures and components in an industrial environment. The results are analyzed and compared, as possible, with solutions presented in the literature.

Capítulo 1

Introdução

A atividade de engenharia sempre envolveu a tentativa de obter a melhor condição de projeto, mais econômica e segura, isto é, o resultado otimizado, sendo isto verdade para o campo da engenharia de projeto. Um projeto de engenharia é concebido como um tipo de arte que demanda grande esforço e criatividade, e a experiência dos engenheiros e projetistas, tem sido de grande importância no processo. A algumas décadas atrás, o processo de projeto era sequencial e realizado na base de “testes e erros” ou “tentativa e erro”, onde as habilidades e experiência de engenheiros e projetistas eram os mais importantes pré-requisitos para a decisão e sucesso do projeto e dos testes. Entretanto, na atualidade, apesar de ainda encontrar-se alguns métodos tradicionais, está se tornando muito freqüente, com o desenvolvimento da engenharia e das ferramentas computacionais, o uso de métodos matemáticos e científicos aplicados para soluções de problemas de engenharia, para os quais o uso de métodos experimentais e a experiência do engenheiro, apesar de úteis, torna o processo tedioso e caro.

Atualmente, a forte competição tecnológica requer redução do tempo de projeto e custo dos produtos, além da alta qualidade e funcionalidade. Muita ênfase se tem dado sobre a economia de energia, reciclagem de materiais, consideração de problemas ambientais, etc; e estes aspectos muitas vezes envolve a criação de novos produtos, para os quais a intuição na engenharia deve estar quase que totalmente ausente. Para o desenvolvimento de novos produtos deve, naturalmente, utilizar-se da aplicação de métodos científicos matemáticos, tais como, elementos finitos para análise estrutural, e de otimização, e também a aplicação de novas tecnologias, e metodologias eficientes de projeto.

As pesquisas científicas no campo da engenharia, mais especificamente na área da otimização estrutural, têm aumentado rapidamente nas últimas três décadas. Este interesse deve-

se, principalmente, pelo advento dos métodos matemáticos de análise, como elementos finitos e programação matemática, além do vertiginoso crescimento da capacidade computacional.

Embora, tenha-se notado um avanço teórico e tecnológico, pode-se observar que os métodos para análise de engenharia, bem como os de otimização, em geral, têm sido pouco adotados por projetistas e engenheiros no ambiente industrial. Na indústria, de um modo geral, ainda pode-se encontrar a prática de métodos tradicionais, realizando os projetos através da “tentativa e erro”. Somente poucas indústrias, tais como, aeronáutica, aeroespacial, naval, nuclear, de petróleo, pode-se encontrar ferramentas computacionais com métodos matemáticos sendo usados de maneira mais efetiva no ambiente de projeto, preocupando-se em fazer certo na primeira vez, aplicando os métodos que permitam conceber um produto ou um componente otimizado e seguro. No contexto industrial observa-se uma série de fluxos que traçam o perfil de uma área de projeto, como também pode-se verificar a utilização das várias ferramentas de apoio, que auxiliam os projetistas e engenheiros no cumprimento dos objetivos. Dentre os vários métodos e ferramentas destacam-se os métodos de análise de engenharia, como o método de elementos finitos (CAE) e os programas geométricos que permitem o modelamento eletrônico (CAD). Porém, as tecnologias funcionam de modo disperso com muito pouca integração, favorecendo para um maior tempo de projeto.

No processo de análise de engenharia é muito importante ter um procedimento adequado e estruturado integrando as ferramentas disponíveis, além, é claro, da flexibilidade. Quanto aos sistemas CAE/CAD/CAM, disponíveis no mercado, a integração e o uso já estão bem sedimentados e dominados em algumas indústrias. No entanto, quando se fala de CAE, o processo de otimização estrutural, é ainda um tanto rudimentar na sua forma de fazer por parte da indústria, e inflexível por parte dos desenvolvedores e vendedores de programas. O processo rudimentar, se deve ao uso do conhecido método da “tentativa e erro”. Este pode ser explicado pela dificuldade da formulação do problema de otimização, isto é, o projetista ou o engenheiro deve ser capaz de formular o problema de otimização. O projetista necessita conhecer fisicamente o problema, e ser capaz de traduzir para uma linguagem matemática, de modo a definir uma função objetivo, identificar as restrições, e o mais importante, definir as variáveis com as quais irá trabalhar, bem como os seus limites. O procedimento para um sistema de otimização estrutural deve conter facilidades genéricas e flexíveis para a definição do problema a ser estudado, de modo a cobrir as definições possíveis, como também, estar disponível para

integração com os diversos programas comerciais de elementos finitos. Se tais facilidades não se apresentam, os projetistas da indústria não usarão um procedimento para otimização nos seus critérios de projeto, já que, muitos destes foram estabelecidos por muitos anos de prática. Portanto, se projetistas na indústria não podem definir seus critérios usuais, eles se absterão do uso do procedimento de otimização, em função da mudança dos seus já tão bem testados critérios práticos. A inflexibilidade se deve ao fato de que os métodos de otimização que estão incorporados à programas comerciais, são fechados, verdadeiras caixas pretas, e que não permitem aos usuários maior familiaridade com os critérios e recursos para otimização. Dentre as várias opções envolvidas na análise de engenharia, é muito importante ter um procedimento estruturado e adequado às diversas condições de projeto encontradas nas diferentes indústrias. Para que um procedimento de otimização tenha uma integração flexível e de baixo custo, é muito importante o aproveitamento dos recursos CAE e CAD já disponíveis no ambiente de projeto, pois todo o conhecimento adquirido em determinados programas comerciais, o alto custo despendido para implementação destes programas são fatores importantes e não podem ser descartados, uma vez que o custo seria muito alto, e com certeza, estaria provocando um problema bastante sério numa organização.

1.1 Apresentação do Trabalho

A presente dissertação tem como objetivo desenvolver, implementar e integrar um método matemático de otimização com um programa comercial de elementos finitos disponível, de modo que o procedimento aplicado ao processo de otimização no ambiente de projeto, seja flexível e de baixo custo. Obviamente, este procedimento engloba tanto um programa comercial, como também rotinas específicas desenvolvidas para manipulação de arquivos e dados, além de métodos matemáticos para otimização, viabilizando o procedimento no ambiente de projeto.

Atualmente, pode-se encontrar no mercado uma série de programas de elementos finitos genéricos, que muitas vezes possuem em seu corpo uma parte direcionada à solução de problemas de otimização. No entanto, o otimizador somente funciona dentro destes programas. Quanto a flexibilidade, se quer dizer que o procedimento pode integrar-se à vários programas

comerciais, se estes permitirem, como também pode permitir a formulação de uma classe bem maior de problemas do que a disponível em um programa comercial, que normalmente são como uma caixa preta. O baixo custo está associado à disponibilidade, por exemplo, na “internet”, de se obter algoritmos de programação não linear gratuitos, inclusive com códigos fontes. Outro aspecto quanto ao custo é que este procedimento permite aproveitar o programa comercial existente ou ainda o investimento pode ser reduzido a um módulo mais simples para análise por elementos finitos. O programa ANSYS, por exemplo, pode ser adquirido em módulos, em vez do programa completo.

Para proceder o cálculo dos gradientes das funções objetivo e das restrições, fez-se uso do método das diferenças finitas, por ser um método simples e de fácil implementação. Apesar de que em função do número de variáveis de projeto, o tempo de solução fica prejudicado, o método é uma boa alternativa para o caso onde não se tem o conhecimento dos detalhes do programa comercial. Com relação ao método utilizado para otimização foi selecionado o Método do Lagrangeano Aumentado, por sua eficiência e robustez.

A seguir faz-se uma descrição genérica dos vários capítulos que formam o conteúdo desta dissertação.

O capítulo 1 apresenta uma introdução genérica e conceitual do procedimento para otimização, sua importância no projeto estrutural, bem como, um panorama do ambiente de projeto dentro das indústrias, e os meios utilizados para efetuar uma otimização e seus recursos. Também são apresentadas algumas das literaturas utilizadas como referência para o estudo e compreensão dos conceitos, desenvolvimento e implementação do procedimento para otimização aqui apresentado.

No capítulo 2 é feita uma revisão dos conceitos básicos de otimização, com a citação de algumas definições de autores sobre o que significa otimização, variáveis de projeto, modelo parametrizado, restrições, e outros itens necessários ao entendimento do trabalho. Também neste capítulo, são abordados alguns conceitos matemáticos de otimização importantes para o desenvolvimento deste trabalho, tais como, os métodos mais utilizados para otimização sem e com restrição.

Já no capítulo 3 são abordados, de forma genérica, alguns dos métodos de análise de sensibilidade mais comuns, e de forma mais específica, o método de diferenças finitas para o cálculo dos gradientes das funções objetivo e restrição. Além disso, também é apresentado um resumo dos aspectos conceituais de elementos finitos para elasticidade tridimensional, para melhor entendimento dos conceitos abordados neste capítulo.

O capítulo 4 destina-se a definição e descrição do procedimento de otimização proposto, abordando os aspectos positivos e negativos do procedimento. São apresentados também, a forma de organizar o procedimento, os aspectos funcionais e também a forma de implementação do algoritmo.

Em seguida, no capítulo 5 são apresentados exemplos que permitem a validação do procedimento proposto e também um exemplo prático da indústria.

Por fim, no capítulo 6 o presente trabalho é concluído, sendo discutido a viabilidade do procedimento e também sugestões para melhoria e extensão para trabalhos futuros.

1.2 Revisão Bibliográfica

A atividade de pesquisa no campo da otimização estrutural e análise de sensibilidade tem gerado uma infinidade de livros e artigos, dos quais alguns são apresentados como referências utilizadas nesse trabalho.

Vanderplaats [1984] com o livro “*Numerical Optimization Techniques for Engineering Design with Applications*”, aborda de maneira prática os aspectos teóricos e básicos de algoritmos numéricos para otimização. Inicialmente é feita uma abordagem básica sobre os conceitos de otimização e a identificação das características de problemas de otimização com restrição e sem restrição com aspectos matemáticos gerais. Estas características e aspectos são aglutinados para iniciar o desenvolvimento de técnicas numéricas necessárias para uma eficiente capacidade de projeto estrutural. São discutidos técnicas para problemas de minimização de funções de uma variável e multivariável sem restrição, pois são os mais simples e formam a base

do conhecimento para o estudo de métodos numéricos para otimização. A partir de então, são discutidas técnicas numéricas mais complexas para problemas de otimização com restrição e multivariável, como o método simplex, técnicas de programação linear para solução de problemas de minimização com restrição não lineares. Outra técnica que está bem detalhada nesta bibliografia é a técnica para solução de problemas com restrição pela conversão à um problema sem restrição, isto é, para uma seqüência de problemas sem restrição, através do uso da técnica da penalidade, onde é destacado o Método do Lagrangeano Aumentado. Este livro incluiu programação linear seqüencial, o Método das Direções Possíveis e método do Gradiente conjugado como também aborda de modo introdutório, o conceito de dualidade que oferece uma outra dimensão à otimização estrutural.

Haftka e Kamat [1985] publicaram o livro “*Elements of Structural Optimization*”. Do mesmo modo que a referência anterior, esta também procura familiarizar o público alvo, com as técnicas de programação matemática gerais. É apresentado um minucioso levantamento de algoritmos para técnicas de otimização com restrição e sem restrição, incluindo gradiente reduzido, e técnicas dos multiplicadores de Lagrange (“*Multiplier Techniques*”). Haftka, também apresenta uma descrição da interface que o projetista estrutural tem que criar entre a análise estrutural e os programas de otimização.

Arora [1989] lança o livro clássico de otimização “*Introduction to Optimum Design*”. A referência faz uma introdução do processo de otimização, abordando conceitos matemáticos genéricos sem se deter em provar matematicamente teoremas acerca dos critérios de otimização. Arora faz uma revisão dos conceitos básicos do cálculo vetorial, como gradiente de funções, matriz hessiana, expansão em séries de Taylor, mínimo global e mínimo local de uma função. Também são definidos e ilustrados as condições necessária e suficiente para mínimo local de uma função de várias variáveis. As condições necessárias de Kuhn-Tucker para um problema genérico com restrição são também discutidas e ilustradas com vários exemplos. Esta referência aborda também os vários métodos para solução de problemas, como simplex, e algoritmos como “*Steepest Descent*”, Gradiente Conjugado, método de Newton e Newton Modificado são também discutidos para determinação do vetor direção descida.

Rao [1996] com o livro “*Engineering Optimization - Theory and Practice*”, tem incorporado novos conceitos para a otimização, principalmente no que se refere ao uso de

técnicas de programação não linear. Existe uma variedade de aplicações de engenharia para as quais outros métodos de otimização, tais como técnicas de programação linear, geométrica, dinâmica, inteira, e estocástica, são mais adequadas. Este livro apresenta a teoria e aplicações de todas as técnicas de otimização de uma maneira compreensível e didática. Algumas técnicas apresentadas foram desenvolvidas recentemente, tais como algoritmos genéticos, “*Simulated Annealing*”, métodos baseados em redes neurais, lógica “*fuzzy*”, e são também discutidos neste livro.

Também foram pesquisados os vários métodos matemáticos para otimização nos livros escritos por Bazaraa e Shetty [1993], Bertsekas [1982], Luemberger [1982]. Todos os livros citados permitiram a absorção do conhecimento necessário para o desenvolvimento deste trabalho, além da contribuição no processo de obtenção de um método de otimização flexível, e eficiente. Além destes livros, também foram alvo de estudo alguns artigos que abrangem com foco mais específico o escopo desse trabalho. Entre estes são descritos a seguir os mais importantes.

Chen e Ho [1993] descreveram uma forma de integrar programas comerciais de elementos finitos com programas de otimização. Esta abordagem usa um método analítico para análise de sensibilidade. A análise de sensibilidade do projeto requer somente a solução do problema original com nova imposição de carregamento, consistindo de tensões iniciais ou deformação inicial. A vantagem desta abordagem é que ela pode ser implementada num programa comercial de elementos finitos sem conhecimento prévio dos detalhes do programa ou acesso aos seus códigos de origem. Para este artigo foi utilizado o programa MSC/Nastran, como programa comercial.

Mahmoud, Engl e Holzleitner [1994] descreveram uma metodologia para otimização estrutural usando um programa comercial de elementos finitos (MSC/Nastran) para análise estrutural, um método Quase Analítico para análise de sensibilidade, e programação sequencial quadrática com um ativo grupo de estratégias para otimização.

Kikuchi e Horimatsu [1994] apresentaram um procedimento escrito em UNIX C-shell que envolve a integração de programas externos, podendo resolver problemas de otimização que envolvem otimização dimensional e de forma de uma estrutura. Destaca-se aqui a grande

flexibilidade na escolha de análise e código de otimização, enquanto as tecnologias comerciais de otimização disponíveis são sistemas muito sofisticados, e restritos a determinadas classes de problemas. O ponto chave desta filosofia é a capacidade de geração de malha automática na análise do modelo, pelo uso de pré-processadores, como PATRAN ou outros.

Han e Wang [1993] apresentaram um estudo de projeto ótimo de painéis com um furo circular e elíptico. O objetivo é encontrar a melhor posição do furo, tamanho e orientação assim como minimizar a máxima deformação tangencial ao longo da circunferência do furo. O problema é tratado como um problema de otimização de forma e o método para parametrização do contorno do furo é desenvolvido de maneira a manter a forma requerida do furo. O problema de otimização neste artigo, é resolvido iterativamente usando elementos finitos com método p-adaptativo e programação linear. Como resultado, o artigo mostra exemplos numéricos que indicam substancial redução na deformação tangencial máxima pelo ajuste do tamanho, localização e orientação de um furo circular e elíptico.

Arora e Haug [1979] mostraram uma abordagem clássica dos conceitos de análise de sensibilidade citando três fundamentais abordagens para a análise de sensibilidade de projeto. Eles são o “*Virtual Load Method*”, o “*State Space Method*” e o “*Design Space Method*”. Uma análise destes três métodos indica que o “*State Space Method*” e o “*Design Space Method*” são mais gerais do que o “*Virtual Load Method*”, mas qualquer um destes procedimentos pode ser aplicado e incorporado dentro de um critério de ótimo ou um método de programação matemática para otimização estrutural.

Lund [1994] em sua tese de doutorado, desenvolveu, implementou e integrou métodos para análise estrutural, análise de sensibilidade e otimização, dentro de ambiente genérico para um processo iterativo de projeto estrutural, análise de projeto, análise de sensibilidade e otimização. Nesta tese Lund desenvolveu toda a biblioteca de elementos finitos para solução de problemas lineares, isto é, análise estática, frequência natural, análise térmica em regime permanente, termo-elástico, auto-valores e auto-vetores e flambagem. Para análise de sensibilidade, o método usado possui uma abordagem direta, é o método semi-analítico, onde a derivada de várias matrizes de elementos e vetores é baseada na diferenciação numérica exata. Um método genérico e flexível de formulação de problemas de programação matemática foi desenvolvido. A formulação matemática foi baseada sobre a formulação exata e a

implementação envolveu um método capaz de interpretar e executar a diferenciação da função definida pelo usuário. As condições necessárias de ótimo foram discutidas e algoritmos numéricos iterativos foram desenvolvidos e usados para a solução de problemas de projeto estrutural.

Capítulo 2

Otimização

2.1 Introdução

É importante no estudo sobre otimização verificar alguns conceitos relativo a este termo. Em otimização verifica-se que o problema está sempre concentrado na minimização ou maximização de uma função, a qual pode possuir restrições sobre as variáveis que a formam, Arora [1989]. Segundo Rao [1996] o conceito do termo otimização é mais abrangente e é descrito como o ato de obter-se o melhor resultado sob determinadas circunstâncias. Em qualquer projeto, construção, ou manutenção de qualquer sistema de engenharia, engenheiros têm que tomar decisões tecnológicas e gerenciais em vários estágios do projeto, e o objetivo de todas as decisões é minimizar o esforço requerido ou maximizar o benefício desejado. Portanto, o esforço requerido ou benefício desejado em qualquer situação prática pode ser expresso como uma função de certas variáveis que definem o modelo. Assim, otimização pode ser definida como um processo para encontrar as condições que dá o máximo ou mínimo valor de uma função. A fim de tornar mais claro estes conceitos, é ilustrado graficamente na figura 2.1, a diferença entre máximo e mínimo de uma função. Então, se um ponto x^* corresponde ao valor mínimo da função $f(x)$, o mesmo ponto também corresponde ao máximo valor da função negativa $-f(x)$. Então sem perda de generalidade, pode-se dizer que otimização é a minimização de uma função, desde que a maximização deva ser encontrada pelo mínimo do negativo desta mesma função.

Quanto ao aspecto matemático que envolve a formulação dos problemas de otimização, este está fundamentado nas teorias do cálculo. Uma ferramenta importante do cálculo é a aplicação da derivada para solução dos chamados problemas de extremo de uma função.

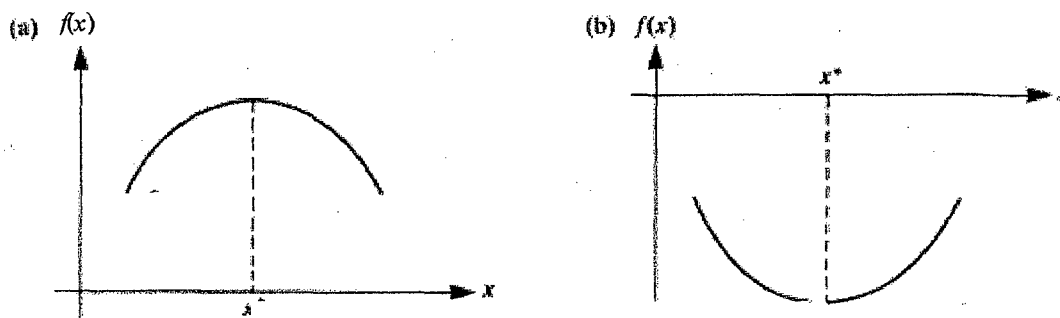


Figura 2.1 – Função unimodal

A aplicação dos conceitos fundamentais do cálculo em teoria de otimização pode permitir determinar vários fatores da engenharia, como por exemplo, qual a forma que deve ter um avião para que a resistência oferecida pelo ar seja a mínima? Qual a forma que deve ter um dique para que tenha a máxima resistência? Qual a forma que um reservatório deve ter para obter o máximo do volume com o mínimo material? Qual o número de determinado produto que uma fábrica deve produzir por dia para obter o máximo lucro? Estas e muitas outras questões do cotidiano são de suma importância para a modelagem matemática destes problemas de otimização, e permitem resolver grande parte das incógnitas no ambiente de projeto de engenharia.

2.2 Formulação do Problema de Otimização

A formulação de um problema de otimização envolve a tradução de um problema físico para um problema matemático bem definido. Esta formulação depende da definição de um grupo de parâmetros, aos quais são atribuídos valores descrevendo um modelo. Alguns destes parâmetros são selecionados para serem manipulados (variáveis de projeto), de modo a satisfazer os requisitos do problema. À medida que, as variáveis de projeto tenham valores que satisfaçam as restrições, pode-se então dizer que este projeto ou que estas variáveis pertencem a um domínio factível. Caso contrário, diz-se que as variáveis de projeto pertencem a um domínio inviável.

O problema de otimização também requer um critério para avaliar a eficiência de cada sistema. Este critério é aplicado a uma função (função objetivo) que depende das variáveis de projeto. Na evolução desta função, as variáveis de projeto devem respeitar limites, como dito anteriormente, pela imposição das condições de restrição. Estes limites são provenientes das condições exigidas pelo projeto, como por exemplo, normas técnicas, funcionalidade, manufaturabilidade, critério de resistência de materiais, etc.

A chamada otimização estrutural identifica o tipo de problema de projeto onde o grupo de parâmetros é subdividido em pré-fixados e variáveis de projeto. O problema consiste em determinar os valores ótimos das variáveis de projeto, tal que elas maximizem ou minimizem uma função específica chamada de função objetivo ou função custo, enquanto satisfazem o grupo de requisitos geométricos, chamado de restrições laterais e ou comportamental (físico), chamados de restrições de estado, que são especificados a priori para o projeto.

Resumidamente, pode-se descrever o processo de formulação de um problema de otimização em três etapas. A primeira etapa é a identificação das variáveis de projeto a partir do conjunto de parâmetros que definem a estrutura ou o componente. Esta fase é considerada a mais vital para a formulação do problema. Uma vez descritas as variáveis de projeto que definem o sistema, a etapa seguinte é a identificação da função objetivo, a qual avaliará o desempenho do projeto em relação aos parâmetros selecionados como variáveis de projeto. A última etapa na formulação é a identificação e o desenvolvimento de expressões matemáticas responsáveis pela imposição das restrições ao projeto da estrutura ou componente, que também dependem das variáveis de projeto. Matematicamente, a formulação de um problema de otimização consiste na definição de um conjunto de equações não lineares, e genericamente, pode-se representar o problema da seguinte forma:

$$\textit{minimizar: } f(x) \tag{2.1}$$

sujeito a:

$$g_j(x) \leq 0; \quad j = 1, \dots, m; \textit{ restrição de desigualdade} \tag{2.2}$$

$$h_p(x) = 0; \quad p = 1, \dots, l; \textit{ restrição de igualdade} \tag{2.3}$$

$$x_i^p \leq x_i \leq x_i^u; \quad i = 1, \dots, n; \quad \text{restrição lateral} \quad (2.4)$$

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}; \quad \text{vetor variáveis de projeto} \quad (2.5)$$

onde,

l , é o número total de restrições de igualdade;

m , é o número total de restrições de desigualdade;

n , é o número total de variáveis de projeto;

x_i^p , é o limite inferior;

x_i^u , é o limite superior.

A função objetivo dada pela equação (2.1), bem como as funções restrições definidas pelas equações (2.2) e (2.3) podem ser funções lineares ou não lineares e explícitas ou implícitas em x . As relações implícitas podem ser solucionadas por algum método analítico ou numérico. É importante salientar que, exceto para classes especiais de algoritmos de otimização, as funções devem ser contínuas e diferenciáveis em x .

A equação (2.5) define o vetor de variáveis de projeto e a equação (2.4), os limites laterais para as variáveis de projeto. Embora estas restrições laterais possam ser incluídas nas restrições de desigualdade, é conveniente tratá-las separadamente. As restrições (2.2), (2.3) e (2.4) dividem o domínio do projeto em duas partes, o domínio viável, onde as restrições são satisfeitas e o domínio inviável, onde pelo menos uma das restrições é violada. O problema declarado é chamado de problema de otimização com restrição.

2.3 Processo Iterativo Genérico para Otimização

Os algoritmos de otimização são entendidos como processos iterativos utilizados na solução de problemas lineares e não lineares. Estes processos requerem uma estimativa inicial x^0 , das variáveis de projeto. Iniciando deste ponto, é através de um processo iterativo com pequenos incrementos que um novo conjunto ou configuração é obtido para melhorar o valor da função objetivo $f(x)$, isto é, minimizando esta função, obedecendo as restrições impostas. A forma mais comum que representa o processo iterativo é dada por:

$$x_i^{k+1} = x_i^k + \Delta x_i^k \quad k = 0, 1, 2, \dots \quad i = 1, \dots, n \quad , \quad (2.6)$$

onde x representa as variáveis de projeto, k representa o número de iterações, i é o número da variável de projeto, Δx representa uma variação na variável de projeto. Pode-se definir Δx , como

$$\Delta x^k = \alpha_k d^k; \quad \alpha_k > 0, \quad (2.7)$$

tal que, α_k é um escalar positivo que define o passo, isto é, a distância que se move na direção d^k que é a direção de descida considerada. Portanto, substituindo (2.7) em (2.6) obtêm-se

$$x^{k+1} = x^k + \alpha_k d^k \quad (2.8)$$

Neste sentido, o processo de solução da equação (2.8), compreende dois subproblemas, o primeiro subproblema consiste na determinação da direção de descida (busca) d^k , e o segundo a determinação do comprimento α_k na direção de descida, que minimiza a função objetivo $f(x)$. Observa-se que existem muitos procedimentos para o cálculo de α_k e do vetor direção de busca d^k . Um importante aspecto, é que num problema n -dimensional, onde a direção de descida é conhecida, o problema torna-se um problema de busca em linha cuja variável é α .

O processo iterativo requer o cálculo de uma direção de descida em x^k , e a solução do problema de busca em linha, sendo então determinado o ponto ótimo x^{k+1} . Mas, caso x^{k+1} não atenda como ponto de mínimo, isto é, não satisfaça as condições de otimalidade, pode-se ter a possibilidade no processo iterativo de obter-se um outro ponto x^{k+2} , de modo a ter um valor

menor para a função objetivo do que o encontrado com x^{k+1} . Como resultado da busca unidimensional tem-se:

$$f(x^{k+1}) < f(x^k) \quad (2.9)$$

substituindo o termo x^{k+1} da equação (2.8) em (2.9) obtêm-se

$$f(x^k + \alpha_k d^k) < f(x^k) \quad (2.10)$$

expandindo o lado esquerdo da expressão linearmente em série de Taylor chega-se a:

$$f(x^k) + \alpha_k (\nabla f(x^k))^T \cdot d^k < f(x^k). \quad (2.11)$$

Logo, d^k é direção de descida e satisfaz a inequação:

$$\langle \nabla f(x^k), d^k \rangle < 0 \text{ e } \alpha_k > 0. \quad (2.12)$$

Este produto interno é a expressão que representa a condição para a direção de descida.

Assumindo d^k como sendo uma direção de descida conhecida, encontrar o passo α_k envolve a solução de um subproblema para minimizar $f(x^k + \alpha_k d^k)$ que é um problema de busca unidimensional com relação a variável α . Então

$$\bar{f}(\alpha_k) \equiv f(x^k + \alpha_k d^k) \quad (2.13)$$

onde, $\bar{f}(\alpha_k)$ é a nova função com α como variável independente. Note que se $\alpha_k = 0$, então $\bar{f}(0) = f(x^k)$, que é a própria função objetivo. A minimização deve ser sobre todos reais α_k não negativos ou todos α_k , isto é, $\alpha_k > 0$, tais que, $x^k + \alpha_k d^k$, é viável. Se d^k é a direção de descida tem-se:

$$f(x^{k+1}) = \bar{f}(\alpha_k) < \bar{f}(0) = f(x^k) \quad (2.14)$$

logo

$$f(x^{k+1}) = \min_{\alpha > 0} \bar{f}(\alpha_k) \tag{2.15}$$

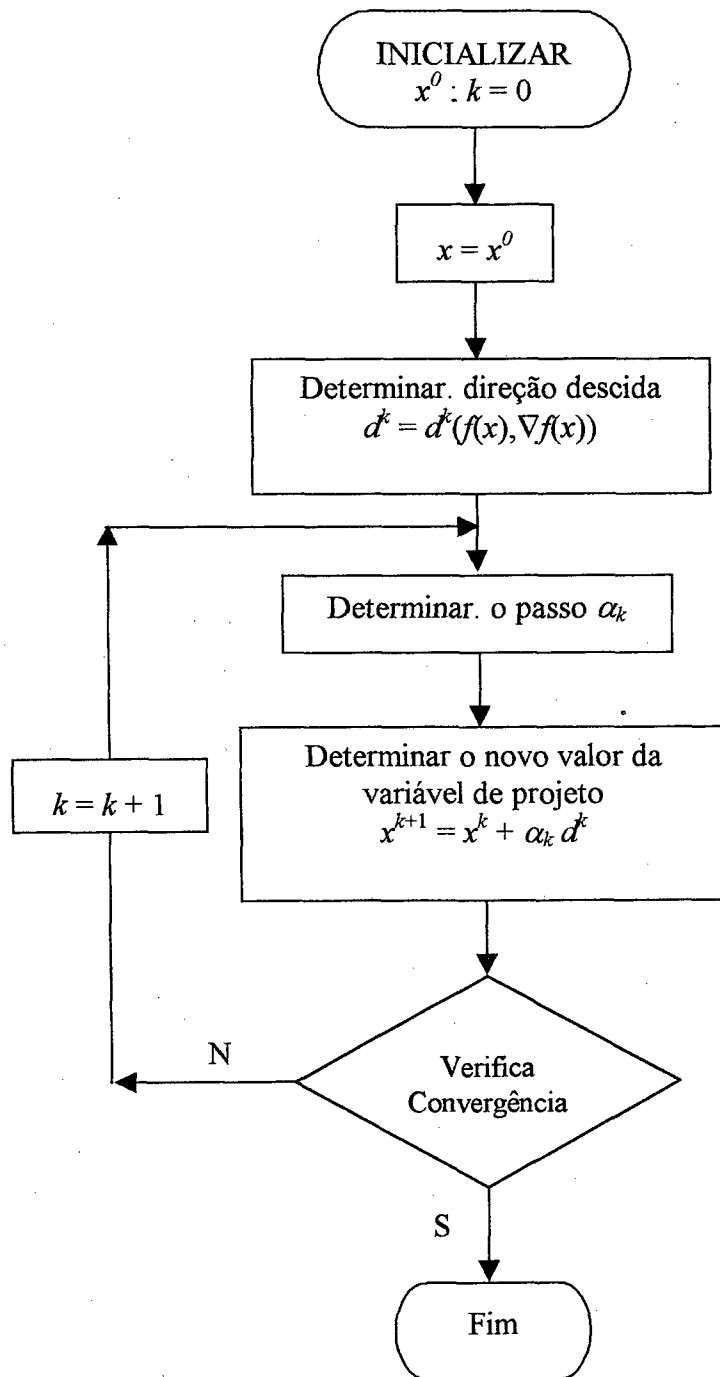


Figura 2.2 – Fluxograma para processo genérico de otimização.

Para o caso de funções simples pode-se calcular analiticamente α_k , aplicando as condições necessária e suficiente de otimalidade, então para $\alpha_k \equiv \alpha^*$ tem-se

$$\left. \frac{d\bar{f}(\alpha_k)}{d\alpha} \right|_{\alpha^*} = 0 ; \text{condição necessária} \quad (2.16)$$

$$\left. \frac{d^2\bar{f}(\alpha_k)}{d\alpha^2} \right|_{\alpha^*} > 0 ; \text{condição suficiente} \quad (2.17)$$

logo, diferenciando $\bar{f}(\alpha_k)$ da equação (2.13) obtêm-se

$$\left. \frac{d\bar{f}(\alpha)}{d\alpha} \right|_{\alpha^*} = \nabla\bar{f}(\alpha) \cdot d^k \quad (2.18)$$

conseqüentemente, a condição necessária para um valor ótimo de α_k implica em

$$\nabla\bar{f}(\alpha) \cdot d^k = 0 \leftrightarrow \nabla\bar{f}|_{\alpha^*} \perp d^k \quad (2.19)$$

Então dada a direção de descida, a determinação de α^* , requer a solução, em geral, da equação não linear

$$\nabla\bar{f}(\alpha) \cdot d^k = 0 \quad (2.20)$$

satisfazendo as condições de otimalidade

Estes aspectos caracterizam o método chamado *método de busca unidimensional*. Dentre as várias considerações no processo de otimização, sem dúvida um dos mais importantes é o critério de parada do processo iterativo. A busca é terminada quando nenhum progresso a mais pode ser feito para melhorar a condição de mínimo da função objetivo sem que seja violada algumas das restrições impostas. Em geral os processos verificam a condição de ótimo, que deve ser satisfeita no ponto de mínimo, são as condições de Kuhn Tucker. Pode-se salientar que nos métodos numéricos iterativos o critério de parada para o processo depende da precisão que se deseja. A influência do critério está na eficiência e confiabilidade do método utilizado. O processo iterativo é ilustrado na figura 2.2.

2.4 Métodos Numéricos

O processo de otimização aplicado à engenharia é, normalmente, formado por funções não lineares ou lineares, que expressam matematicamente o problema físico. Estas funções são: a função objetivo e as restrições. Tradicionalmente, estes problemas físicos, quando simples, são muitas vezes resolvidos através de métodos analíticos e gráficos. Entretanto, como os projetos tornam-se cada vez mais complexos, a exigência de recursos computacionais para a solução de problemas numéricos é inevitável. Pode-se enumerar algumas razões que tornam os métodos analíticos impróprios. Uma das razões é o grande número de variáveis de projeto e funções de restrições, onde as funções são em geral não lineares, tornando impraticável o uso de métodos analíticos para solução dos problemas de otimização. Segundo Vanderplaats [1984], pode-se enumerar algumas vantagens de uma abordagem numérica e programação matemática de otimização não linear:

- * Redução do tempo, com a utilização de algoritmos numéricos apropriados;
- * Procedimento de otimização lógico e sistêmico;
- * Capacidade de se considerar um grande número de variáveis de projeto;
- * Não se baseia na intuição e experiência de pessoas.

Apesar das inúmeras vantagens, encontram-se presente algumas limitações nos métodos numéricos:

- * Tempo computacional aumenta com o aumento de variáveis de projeto, neste caso pode-se produzir um mal condicionamento do problema;
- * Raramente se garante um ótimo global. Portanto, pode ser necessário reiniciar o processo para diferentes pontos, de modo a ter razoável segurança quanto ao resultado obtido como mínimo global;

- * Muitos algoritmos têm dificuldade em lidar com funções descontínuas. Também problemas altamente não lineares podem convergir muito lentamente ou mesmo não convergir. Isto requer um cuidado particular na formulação do programa;
- * Devido a alguns programas de análise não possuírem um processo automático de otimização, a adaptação dos programas requer uma reprogramação;
- * Se o problema não for corretamente definido, os resultados podem ser enganosos. Portanto, os resultados devem ser verificados cuidadosamente.

Devido a estas razões muitos métodos numéricos para otimização não linear foram desenvolvidos. Detalhes e teorias sobre alguns dos métodos utilizados neste trabalho são apresentados adiante. O método utilizado para resolução dos problemas descrito neste trabalho é o Método do Lagrangeano Aumentado, no qual a solução é obtida através da resolução de uma seqüência de problemas sem restrição. Como são necessários alguns procedimentos de calibragem do algoritmo, é importante o entendimento dos conceitos que são usados nos algoritmos, não somente para otimização com restrição, mas também para alguns algoritmos para otimização sem restrição.

2.4.1 Problemas Sem Restrição

Os métodos para resolver problemas sem restrição são também muito aplicados em problemas de engenharia. As técnicas a serem apresentadas podem também ser usadas na solução de sistemas lineares e não lineares. E o emprego destas técnicas para a solução de problemas de otimização sem restrição são também de grande importância para métodos indiretos, como Método de Penalidade e Método do Lagrangeano Aumentado para problemas com restrição.

Como se sabe, os problemas de otimização podem ser altamente não lineares, sendo muito difícil sua solução direta por métodos analíticos tradicionais. Sendo assim, muitos métodos numéricos têm sido desenvolvidos para a solução dessas equações. Estes métodos baseiam-se em procedimentos numéricos de busca, que funcionam de forma iterativa.

O estudo de problemas sem restrição é dividido em duas partes, que são a minimização de funções de uma variável e minimização de funções de várias variáveis. Grande parte dos métodos numéricos aplicados em otimização baseia-se no conceito de busca unidimensional, conforme já mostrado anteriormente. Os processos iterativos de otimização são separados em duas partes: a determinação da direção de descida d^k e a determinação do incremento de descida α . A seguir são apresentados uma série de métodos para determinação de α , considerando uma direção de descida conhecida. Em um processo completo, primeiramente deve-se determinar a direção de busca ou descida, mas estes serão abordados mais adiante. Para determinação do passo α , isto é, determinação do passo ótimo, é destacado neste trabalho o Método de “*Golden Section Search*”, Interpolação Quadrática, “*Quadratic Golden Section*”. Quanto aos métodos para a determinação da direção de descida ou de busca são verificados o “*Steepest Descent Method*” e também os métodos de segunda ordem, tais como o Método de *Newton* e *Quase Newton*.

2.4.1.1 Problemas de Minimização 1-D

I. Método “*Golden Section Search*”

Para muitos problemas uma função $\bar{f}(\alpha)$ pode não ser muitas vezes simples de resolver analiticamente, sendo necessário um método numérico para encontrar α^* , tal que,

$$\bar{f}(\alpha^*) = \min_{\alpha > 0} \bar{f}(\alpha) \quad (2.21)$$

Muitos métodos de busca unidimensional necessitam que a função seja unimodal no intervalo desejado. O Método “*Golden Section Search*” usado para minimização unidimensional é considerado uma das técnicas mais populares. Primeiramente, a função deve ser unimodal, porém não necessita ser diferenciável. Em segundo lugar, em relação a outras técnicas nesta, a taxa de convergência é conhecida. E por último, é que o Método “*Golden Section Search*” é facilmente programável. Os processos iterativos de busca unidimensional são na maioria técnicas baseadas na comparação dos valores das funções nos vários pontos ao longo da direção de busca.

Para o caso de uma função genérica, determina-se um intervalo de incerteza, no qual a função $\bar{f}(\alpha)$ tem um mínimo. O Método “*Golden Section Search*” inicia a partir do

estabelecimento de limites inferior e superior sobre a variável independente α . Numericamente, o que determina-se é o intervalo dentro do qual α minimiza $\bar{f}(\alpha)$. O intervalo (α_l, α_u) é chamado intervalo de incerteza(I), definido como

$$I = \alpha_u - \alpha_l, \tag{2.22}$$

tal que,

$$\alpha^* \in (\alpha_l, \alpha_u) \tag{2.23}$$

Basicamente, a técnica é dividida em duas partes: Primeiramente, é estabelecido um intervalo de incerteza inicial (α_l, α_u) , onde a função é calculada em cada um dos extremos, obtendo os valores de $\bar{f}(\alpha_l)$ e $\bar{f}(\alpha_u)$. Numa segunda fase, são selecionados pontos intermediários (α_1, α_2) , tal que, $(\alpha_1 < \alpha_2)$, e novamente a função é calculada nestes pontos obtendo $\bar{f}(\alpha_1)$ e $\bar{f}(\alpha_2)$.

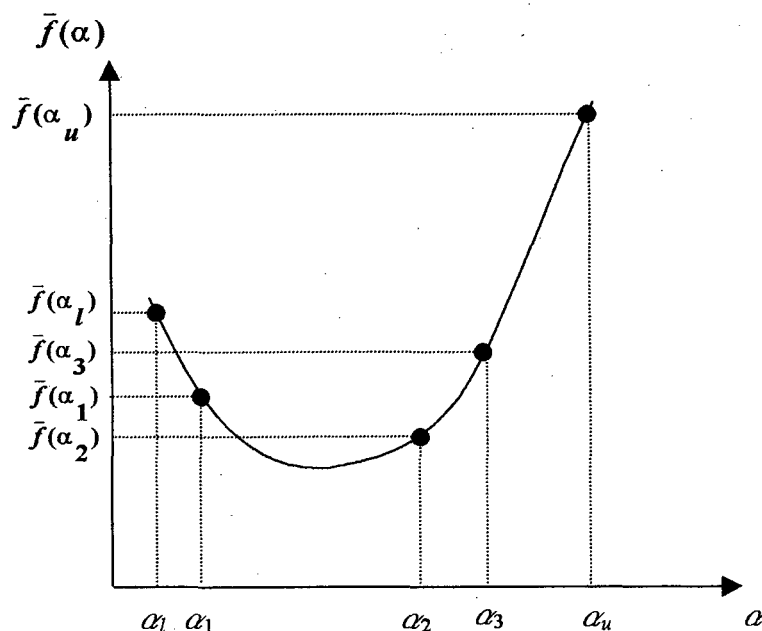


Figura 2.3 – Função unimodal – redução do intervalo de incerteza

Como foi assumido que a função é unimodal no intervalo desejado, segue que α_1 e α_2 , formarão o novo intervalo para o mínimo. Neste caso $\bar{f}(\alpha_1)$ é maior do que $\bar{f}(\alpha_2)$, então α_1 forma um

novo limite inferior, e agora tem-se um novo intervalo que é (α_1, α_u) . Do mesmo modo se $\bar{f}(\alpha_2)$ for maior do que $\bar{f}(\alpha_1)$, fica claro que (α_l, α_2) é um novo intervalo. Uma vez identificado o novo intervalo, pode-se então selecionar um ponto adicional $\alpha_3 \in (\alpha_1, \alpha_u)$, onde a função é novamente avaliada, obtendo $\bar{f}(\alpha_3)$. Comparando $\bar{f}(\alpha_2)$ e $\bar{f}(\alpha_3)$, pode-se ver que $\bar{f}(\alpha_3)$ é maior do que $\bar{f}(\alpha_2)$, então α_3 substitui α_u , como limite superior, e o intervalo fica (α_1, α_3) . Então através do processo iterativo, o intervalo de incerteza é refinado, estabelecendo intervalos menores, eliminando as regiões que não contém o mínimo até encontrar o valor mínimo da função de acordo com a tolerância estabelecida. Graficamente, a figura 2.3 mostra, conforme a situação explicada, como o Método “*Golden Section Search*” funciona. O algoritmo oferecido por Press et al [1992] fornece uma maneira rápida e eficiente de obter-se os intervalos para reduzir os limites de modo à encontrar o mínimo o mais rápido possível.

II. Método da Interpolação Quadrática

O método de interpolação quadrática requer pouca avaliação da função objetivo, além de não necessitar o cálculo da derivada. A técnica de interpolação quadrática permite por meio do intervalo de incerteza aproximar a função objetivo $\bar{f}(\alpha)$ por uma curva quadrática, a partir da determinação da função objetivo em três pontos distintos. Desta maneira pode-se determinar os coeficientes de um polinômio quadrático definido por:

$$q(\alpha) = a_0 + \sum_{i=1}^2 a_i \alpha^i \quad (2.24)$$

a_0, a_i - representam os coeficientes da função polinomial a serem determinados.

Necessariamente, deve-se assumir um intervalo cuja função $\bar{f}(\alpha)$ tem as características de funções unimodais, que são bem comportadas e que o intervalo de incerteza inicial (α_l, α_u) seja conhecido. Seja α_i um ponto intermediário qualquer dentro do intervalo de incerteza e $\bar{f}(\alpha_l), \bar{f}(\alpha_i), \bar{f}(\alpha_u)$ os valores das funções nos respectivos pontos, então $q(\alpha)$ tem o mesmo valor que a $\bar{f}(\alpha)$ nos pontos α_l, α_i e α_u , de onde obtém-se o sistema de equações como segue

$$\begin{aligned}
 q(\alpha_i) &= a_0 + a_1\alpha_i + a_2\alpha_i^2 = \bar{f}(\alpha_i) \\
 q(\alpha_i) &= a_0 + a_1\alpha_i + a_2\alpha_i^2 = \bar{f}(\alpha_i) \\
 q(\alpha_u) &= a_0 + a_1\alpha_u + a_2\alpha_u^2 = \bar{f}(\alpha_u)
 \end{aligned}
 \tag{2.25}$$

resolvendo este sistema de equações lineares simultaneamente, obtém-se a_0, a_1, a_2 .

O valor mínimo para a função quadrática $q(\alpha)$ na equação (2.24) é encontrado no ponto α^* , calculado pelas condições necessária e suficiente, de onde se obtém

$$\alpha^* = -\frac{1}{2 \cdot a_2} \cdot a_1 \quad ; \quad \text{se} \quad \frac{d^2 q(\alpha^*)}{d\alpha^2} = 2a_2 > 0
 \tag{2.26}$$

III. Método “*Quadratic Golden Section*”

Proposto por Yu et al [1991], o Método “*Quadratic Golden Section*” é uma combinação de três métodos que são o Método da Interpolação Quadrática, o Método “*Golden Section Search*” e um algoritmo de procura lateral. O resultado desta combinação é um algoritmo eficiente e bastante confiável. Este novo algoritmo aproveita o melhor de cada método descrito anteriormente, podendo a resolução ser dividida em duas fases. Primeiramente, através do Método de Interpolação Quadrática, o intervalo de incerteza é reduzido a um tamanho suficientemente pequeno até que os pontos de interpolação do intervalo subsequente calculados, atinjam a menor distância, obedecendo ao critério de tolerância. Na segunda fase, o Método “*Golden Section Search*” faz o processo de comparação da função calculada em cada um dos novos pontos dos novos intervalos e por conseguinte obtendo a solução. Do mesmo modo que o “*Golden Section Search*”, este algoritmo também parte de um intervalo inicial dentro do qual existe um ponto de mínimo.

2.4.1.2 Problema Sem Restrições *N-dimensional*

Anteriormente, a preocupação foi determinar α_k , considerando a direção de descida conhecida. Agora o alvo das atenções serão os métodos que permitem determinar a direção de descida d^* . Estes métodos são os que utilizam informações do gradiente da função a ser minimizada. A eficiência do método é adquirida as custas de informações do gradiente da função, que são obtidos analiticamente, ou numericamente pela aplicação do método das

diferenças finitas. No método de primeira ordem distinguem-se três subclasses que são vistos a seguir.

I. Método “*Steepest Descent*”

O Método “*Steepest Descent*” ou Método do Gradiente é o mais conhecido, no entanto é o de mais baixa eficiência. Este método pertence a classe de métodos de primeira ordem. O método proposto por Cauchy [1847], é de extrema importância do ponto de vista teórico, foi o ponto de partida para o desenvolvimento de métodos mais robustos e sofisticados. O método é definido por um processo iterativo, a partir da equação (2.8), sendo também usado o vetor gradiente neste processo. A direção de descida da função objetivo é representada pelo negativo do seu gradiente no ponto considerado. Sendo a função $f(x)$ diferenciável com relação a x , a direção de descida para função no ponto x^k é dada por:

$$d^k = -\nabla f(x^k) \quad (2.27)$$

A equação (2.27) dá a mudança de direção no domínio do projeto, onde d^k é usado na equação (2.8) para executar o processo de busca unidimensional, já tratado anteriormente. Logo,

$$x^{k+1} = x^k + \alpha_k d^k, \quad (2.28)$$

tal que, α_k é a solução do processo de busca unidimensional,

$$\bar{f}(\alpha) \equiv \min_{\alpha > 0} f(x^k + \alpha_k d^k), \quad (2.29)$$

portanto, da condição de descida, equação (2.12), se obtém:

$$\langle \nabla f(x^k), d^k \rangle = \langle \nabla f(x^k), -\nabla f(x^k) \rangle = -\|\nabla f(x^k)\| < 0, \quad (2.30)$$

isto é,

$$\begin{cases} -\|\nabla f(x^k)\| < 0 & \text{se } \|\nabla f(x^k)\| \neq 0 \\ -\|\nabla f(x^k)\| = 0 & \text{se } \|\nabla f(x^k)\| = 0 \end{cases}; \text{ condição necessária de otimalidade} \quad (2.31)$$

e o critério de convergência em geral é dado por:

$$\|\nabla f(x^{k+1})\| \leq tol \tag{2.32}$$

e

$$\frac{\|x^{k+1} - x^k\|}{\|x^{k+1}\|} \leq tol \tag{2.33}$$

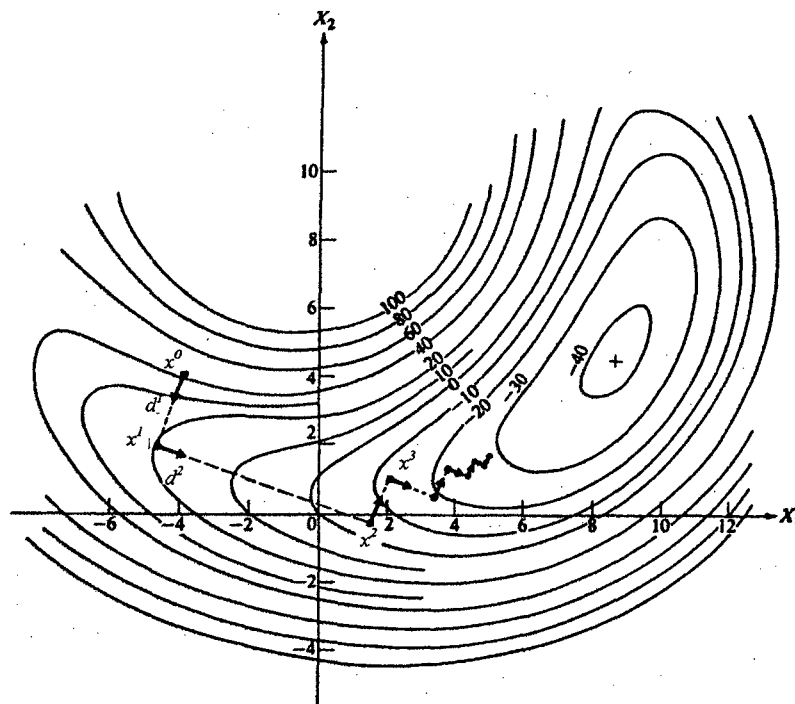


Figura 2.4 – Interpretação geométrica do Método “Steepest Descent” – Vanderplaats

O algoritmo básico para otimização pelo Método “Steepest Descent” é geometricamente interpretado e ilustrado na figura 2.4. Pode-se observar que o método possui uma taxa de convergência muito pobre e em cada etapa as direções são ortogonais entre si, ou seja, $\langle \nabla f(x^{k+1}), d^k \rangle = 0$.

II. Método de Newton

O Método de Newton é classificado como um método de segunda ordem, isto pelo fato de que, além das informações da função objetivo $f(x)$ e dos gradientes $\nabla f(x)$ tem-se também a necessidade de informações da derivada de segunda ordem, ou seja, da matriz Hessiana $[H]$. A idéia básica do Método de Newton, é usar uma expansão quadrática em séries de Taylor, para aproximar localmente a função objetivo $f(x)$. Portanto, a função aproximada em x^k pode ser expressa como:

$$f(x) \approx f(x^k) + \nabla f(x^k)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x^k) \Delta x, \quad (2.34)$$

na qual

$$\Delta x = (x - x^k) \quad (2.35)$$

e,

$$\nabla^2 f(x^k) = [H], \quad (2.36)$$

que é matriz Hessiana de $f(x)$ no ponto x^k .

Considerando que a matriz Hessiana $[H]$ seja positiva definida, então da equação (2.34) tem-se que:

$$\nabla f(x) = \nabla f(x^k) + [H] \Delta x \quad (2.37)$$

e impondo condições de otimalidade em x^{k+1} , isto é,

$$\left. \frac{\partial f(x)}{\partial x} \right|_{x=x^{k+1}} = 0, \quad \text{condição necessária de otimalidade} \quad (2.38)$$

$$\nabla f(x^k) + [H(x^k)] \Delta x \Big|_{x=x^{k+1}} = 0 \quad (2.39)$$

então,

$$\Delta x = -[H(x^k)]^{-1} \cdot \nabla f(x^k) \quad (2.40)$$

Para estimar um novo valor x^{k+1} , basta usar a equação (2.35), portanto,

$$x^{k+1} = x^k + \Delta x \quad (2.41)$$

$$x^{k+1} = x^k - [H]^{-1} \cdot \nabla f(x^k) \quad (2.42)$$

No entanto, como a equação (2.34) é uma aproximação da função objetivo em x^k , logicamente não se obtém uma precisão quanto ao ponto de mínimo de $f(x)$. Em cada iteração necessita-se calcular a matriz Hessiana e sua inversa, e isto requer um esforço computacional considerável, o que compromete a eficiência do método.

O Método de *Newton* possui, na proximidade do ponto ótimo, uma taxa de convergência quadrática. Isto torna o método atrativo para pontos suficientemente próximos do ponto ótimo. O algoritmo pode ser aprimorado através da introdução do parâmetro α e da utilização de um método de busca unidimensional para a determinação do parâmetro ótimo, isto é,

$$x^{k+1} = x^k - \alpha_k \cdot [H]^{-1} \cdot \nabla f(x^k) \quad (2.43)$$

na qual α_k , é solução ótima de

$$\min_{\alpha > 0} f(x^k - \alpha_k \cdot [H]^{-1} \cdot \nabla f(x^k)) \quad (2.44)$$

O uso deste parâmetro permite obter-se uma estabilidade e garantia de convergência, desde que a matriz Hessiana permaneça positiva definida em todas as iterações.

Entretanto, se considerado um ponto arbitrário, nem sempre se pode garantir que a matriz Hessiana seja positiva definida ou mesmo existir, então o método requer alguma modificação para contornar este tipo de problema. A solução adotada é o uso do Método “*Steepest Descent*”, isto é,

$$[H]^{-1} = [I] \quad (2.45)$$

onde, $[I]$ representa a matriz Identidade. Este procedimento é efetuado em situações em que a direção de descida d^k , dado por

$$d^k = -[H]^{-1} \cdot \nabla f(x^k), \quad (2.46)$$

viola a condição de descida abaixo,

$$\langle \nabla f(x^k), d^k \rangle < 0 \quad (2.47)$$

A principal dificuldade do método de Newton é que a matriz Hessiana $[H]$ pode ser singular, ou pelo menos não positiva definida, violando a condição de descida. A aplicação do Método de Newton se restringe a problemas cuja obtenção da matriz Hessiana, positiva definida, e $f(x)$ seja simples.

III. Método Variável Métrico – “*Quase Newton Methods*”

Anteriormente, foi verificado o uso de métodos que obtinham informações através do gradiente da função objetivo, cuja taxa de convergência é linear, este método foi o Método “*Steepest Descent*”. Com o Método de Newton, foi usado informações da matriz Hessiana, isto é, derivada de segunda ordem com boas propriedades de convergência, quando próximo do ponto ótimo. Entretanto, o método requer um esforço computacional impraticável, e tedioso para muitos problemas de engenharia. O Método Variável Métrico coloca-se como um método intermediário ao Método “*Steepest Descent*” e o Método de Newton. Considera-se aqui que a inversa da matriz Hessiana é impraticável e caro para ser calculada. A idéia básica do método é usar uma aproximação da inversa da matriz Hessiana, utilizando-se de informações de gradientes de pontos anteriores no processo de busca iterativa.

Particularmente, no caso de funções quadráticas a matriz Hessiana é constante e, à medida que, as iterações ocorrem, a matriz inversa da Hessiana é aproximada por uma matriz n -dimensional A^k , com a propriedade

$$\lim_{k \rightarrow \infty} A^k = [H]^{-1} \quad (2.48)$$

Como ponto de partida considera-se o processo iterativo do método de Newton, então a equação (2.37) é reescrita como:

$$\Delta x = A^k (\nabla f(x) - \nabla f(x^k)) \quad (2.49)$$

O Método Variável Métrico considera inicialmente, no processo iterativo a matriz identidade, isto é, $[H]^{-1} = [I] = A^0$, então o Método “*Steepest Descent*” é aplicado diretamente para a direção de busca. Então de forma genérica a direção de descida é dada por:

$$d^k = -A^k \cdot \nabla f(x^k); \quad k = 0, 1, 2, 3, \dots \quad (2.50)$$

A construção da matriz aproximada A^k deve ser feita de forma que a matriz permaneça positiva definida e simétrica. Assim, na proximidade do mínimo garante-se que a direção de descida moverá x^k para um ponto que decreta a função objetivo. Portanto, a atualização da matriz A^k tende exatamente a inversa da matriz Hessiana obtendo a convergência quadrática do método de Newton. A atualização da matriz A^k guarda informações das iterações anteriores, além disso, uma outra vantagem é que requer somente a determinação da primeira derivada. Este processo de atualização é parte de uma família do Método Variável Métrico, cujos métodos mais utilizados são:

- Método de *Davidon Fletcher Powell* (DFP);
- Método de *Broydon Fletcher Goldfard Shann* (BFGS).

a) Método de *Davidon Fletcher Powell* (DFP).

Um dos métodos para determinar a inversa da matriz Hessiana, foi introduzido por *Davidon* [1959], e posteriormente modificado por *Fletcher e Powell* [1963]. A atualização A^{k+1} por este método é dada pela expressão:

$$A^{k+1} = A^k + \frac{S^k \otimes S^{kT}}{(S^k \cdot Y^k)} - \frac{Z^k \otimes Z^{kT}}{(Y^k \cdot Z^k)} \quad (2.51)$$

na qual,

$$S^k = \alpha_k \cdot d^k = (x^{k+1} - x^k) \quad (2.52)$$

$$Y^k = \nabla f(x^{k+1}) - \nabla f(x^k) \quad (2.53)$$

$$Z = A^k \cdot Y^k \quad (2.54)$$

A matriz A^k , é positiva definida para todo k . Isto implica que o método sempre converge para um mínimo local.

b) Método Broydon Fletcher Goldfard Shann (BFGS)

Neste método a matriz Hessiana e sua inversa são atualizadas em toda iteração. Neste método a primeira iteração novamente se repete como no DFP. Para a atualização da aproximação da inversa da Hessiana no método BFGS, considera-se os mesmos termos iniciais do Método de DFP e adicionalmente acrescenta-se mais uma expressão, portanto a expressão toma a forma:

$$A^{k+1} = A^k + \frac{S^k \otimes S^{kT}}{(S^k \cdot Y^k)} - \frac{Z^k \otimes Z^{kT}}{(Y^k \cdot Z^k)} + [(Y^k \cdot Z^k)] \cdot C^k \otimes C^k \quad (2.55)$$

$$C^k = \frac{S^k}{(S^k \cdot Y^k)} - \frac{Z^k}{(Y^k \cdot Z^k)} \quad (2.56)$$

É interessante deixar claro que a direção de busca é garantida ser realmente de descida para a função objetivo $f(x^k)$, quando a matriz A^k for positiva definida. Para evitar que a aproximação da matriz Hessiana torne-se indefinida ou singular nos métodos numéricos, deve-se implementar desvios de segurança para assegurar uma satisfação numérica e convergência do método. Explicações mais detalhadas podem ser encontradas em várias literaturas, Arora [1989], Bazaraa e Shetty [1979], Chong e Zak [1996], Rao [1996] e McCormick [1983].

Portanto, de forma geral, no Método Variável Métrico a direção de descida é obtida pela equação (2.50), sendo A^k positiva definida, o que garante que d seja uma direção de descida.

Então o processo iterativo, segue com o método de busca unidimensional, computando o passo α e atualizando x^k através da equação (2.8), para chegar ao ponto ótimo, até a convergência $\|\nabla f(x^k)\| < \varepsilon$, na qual ε é a tolerância adotada.

2.4.2 Problemas com Restrição

Verificou-se anteriormente, alguns métodos numéricos usuais para minimização de problemas sem restrição. Agora será abordado o problema de mínimo de funções de n variáveis sujeitas a um grupo de restrições de igualdade e de desigualdade. Este problema foi definido genericamente pelas equações (2.1), (2.2), (2.3), (2.4) e (2.5), onde tanto a função objetivo, como a de restrições são funções não lineares. A filosofia apresentada aqui para resolver tais problemas é aplicar uma transformação do problema, isto é, a abordagem geral é resolver um problema de mínimo da função objetivo com restrição, através da solução de uma seqüência de problemas sem restrição. Esta abordagem é conhecida como *SUMT*, sigla em inglês que significa “*Sequential Unconstrained Minimization Techniques*”. O ponto de partida para o início desta abordagem é o Método da Penalidade Exterior e em seguida é verificado o Método do Lagrangeano Aumentado.

2.4.2.1 Método da Função Penalidade Exterior

O Método da Penalidade Exterior, é o mais fácil, nesta classe de problemas, de ser incorporado em um algoritmo, já que penaliza somente a função objetivo, quando as restrições são violadas. A técnica de penalização possui vantagens e desvantagens, principalmente, com relação ao mal condicionamento numérico. Neste sentido, para reduzir este problema faz-se uso do multiplicador de Lagrange. Para o problema de otimização genérico sujeito as restrições pode-se enunciar:

Encontrar um vetor $x=(x_1, x_2, \dots, x_n)$, o qual minimiza a função objetivo

$$f = f(x) \tag{2.57}$$

sujeito as restrições de igualdade

$$h_p(\mathbf{x}) = 0 ; \quad p = 1, \dots, l \quad (2.58)$$

e as restrições de desigualdade

$$g_j(\mathbf{x}) \leq 0 ; \quad j = 1, \dots, m \quad (2.59)$$

Portanto, a transformação do problema de otimização com restrição, definido nas equações (2.57), (2.58), (2.59), em um problema sem restrição é realizado através de uma abordagem clássica, criando uma função pseudo objetivo, da forma:

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) + \frac{1}{r} \cdot P(\mathbf{x}) \quad (2.60)$$

onde $f(\mathbf{x})$ é a função objetivo original, $P(\mathbf{x})$ é uma função penalidade, cuja parâmetro de controle da penalidade é dado por r .

Este método usa as funções de penalidade que transformam problemas com restrição em um problema sem restrição, o que possibilita a solução de um problema com restrições através da solução de uma seqüência de problemas sem restrição. As restrições são incorporadas na função objetivo através da função penalidade e permite a solução do problema com restrição através da penalização da violação das restrições. O termo “exterior” refere-se ao fato de que as penalidades são aplicadas somente no lado externo do domínio viável, isto é, o mínimo se aproxima pelo lado externo do domínio da função. O método de penalidade é de grande interesse, pelo fato de oferecer um simples e claro método para solucionar os problemas com restrição, e podem ser implementados com programas simples e que possuem bom grau de generalidade, porém com taxa de convergência lenta.

Considerando o problema de minimização com restrição de desigualdade dado nas equações (2.57), (2.59), a idéia básica do método de penalidade é substituir o problema por um problema sem restrição da forma:

Determinar x^*

$$x^* = \lim_{r \rightarrow 0} \bar{x}_r \quad (2.61)$$

onde, \bar{x}_r é a solução do problema:

Dado $r > 0$, determinar \bar{x}_r , tal que

$$\bar{x}_r = \operatorname{argmin} \phi(\mathbf{x}, r) \quad (2.62)$$

onde,

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) + \frac{1}{r} \sum_{j=1}^m (g_j(\mathbf{x})^+)^2 \quad (2.63)$$

a função $g_j(\mathbf{x})^+$, representa a parte positiva da função $g_j(\mathbf{x})$ e pode ser expressa como:

$$g_j(\mathbf{x})^+ \equiv \max\{0, g_j(\mathbf{x})\} ; \quad j = 1, \dots, m \quad (2.64)$$

A condição necessária de otimalidade para \bar{x}_r é dada por:

$$\nabla \phi(\bar{x}_r, r) = \nabla f(\bar{x}_r) + \frac{2}{r} \sum_{j=1}^m g_j(\bar{x}_r) \nabla g_j(\bar{x}_r)^+ = 0 \quad (2.65)$$

onde,

$$\nabla g_j(\bar{x}_r)^+ = \begin{cases} \nabla f(\bar{x}_r) ; & \text{se } g_j(\mathbf{x}) \geq 0 \\ 0 & ; \text{se } g_j(\mathbf{x}) < 0 \end{cases} \quad (2.66)$$

No caso de ser incluído restrições de igualdade obtém-se:

Seja \bar{x}_r a solução do problema (2.57), (2.58) e (2.59). Então, como anteriormente para restrição de desigualdade, tem-se

$$\mathbf{x}^* = \lim_{r \rightarrow 0} \bar{x}_r \quad (2.67)$$

onde, \bar{x}_r é a solução do problema:

Dado $r > 0$, determinar \bar{x}_r , tal que

$$\bar{x}_r = \arg \min \phi(x, r) \tag{2.68}$$

onde,

$$\phi(x, r) = f(x) + \frac{1}{r} \cdot P(x). \tag{2.69}$$

A função penalidade neste caso tem a forma:

$$P(x) = \sum_{j=1}^m [g_j(x)^+]^2 + \sum_{p=1}^l [h_p(x)]^2 \tag{2.70}$$

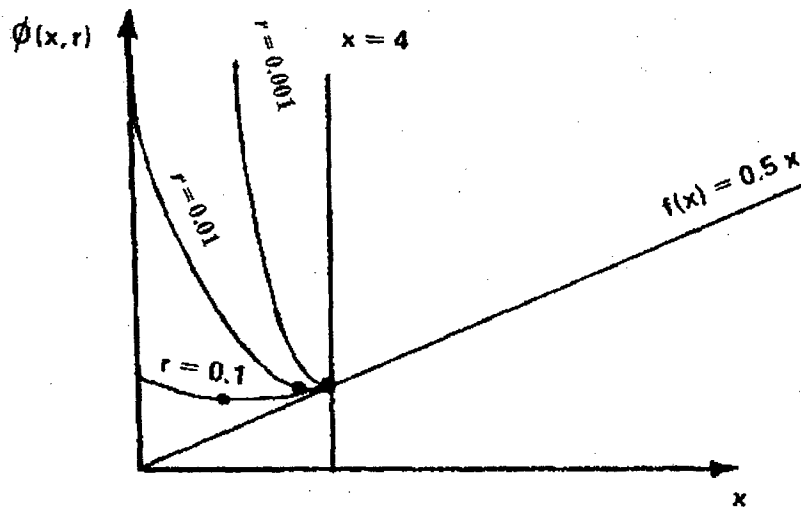


Figura 2.5 -Influência do parâmetro de penalização

Quanto ao parâmetro de penalidade r , este tem a responsabilidade de controlar a magnitude dos termos de penalidade. Se escolhido um valor relativamente pequeno para r , o resultado da função $\phi(x, r)$, é facilmente minimizado, mas não impede a violação das restrições. Por outro lado, um valor muito pequeno de r assegurará a satisfação de todas as restrições, mas em geral torna o problema numericamente mal condicionado. Normalmente, inicia-se com um valor relativamente pequeno para minimizar $\phi(x, r)$, então r é reduzido gradualmente, sendo $\phi(x, r)$, minimizado até obter-se um resultado satisfatório. Pode-se através da figura 2.5, ter idéia da evolução da função em relação ao parâmetro r . Pode-se notar que, como r é reduzido de

um valor inicial, então $\phi(x, r)$ move-se no intervalo fechado das restrições de contorno. Entretanto, a curvatura de $\phi(x, r)$, próximo do mínimo também aumenta. O valor alto da curvatura associada com o valor muito reduzido de r , muitas vezes conduz a dificuldade numérica. Usando-se uma seqüência de valores de r , a localização do mínimo pode ser obtido a partir de um valor inicial de r , como ponto de partida para a busca, sendo o seu valor reduzido gradativamente até a posição de mínimo for identificada, quando então o valor de r é limitado de forma que, $r < r_{min.}$. Assim o mal condicionamento associado com a curvatura grande é contrabalançado pela disponibilidade de um bom ponto de início.

Agora [1994], enumera algumas vantagens e desvantagens do Método de Penalidade:

- 1- É aplicável para problemas genéricos com restrição, considerando tanto restrições de igualdade quanto de desigualdades;
- 2- Arbitrariedade na escolha do ponto inicial;
- 3- O método itera através do domínio inviável onde o problema pode estar indefinido;
- 4- Se o processo iterativo termina prematuramente, o ponto de mínimo pode não ser viável.

2.4.2.2 Método do Lagrangeano Aumentado – M.L.A.

Anteriormente, foi verificado uma abordagem clássica para resolver um problema de mínimo com restrição pelo uso de funções de penalidades exterior. Apesar de ser considerado um método eficiente, Powell [1979] verificou que o processo de otimização pode ser melhorado significativamente incluindo os multiplicadores de Lagrange. Este fato permite reduzir a dependência da escolha do parâmetro de controle da penalidade, bem como sua atualização, que como visto anteriormente, pode causar problemas de mal condicionamento numérico. O método apresentado com esta característica é conhecido como A.L.M., em inglês, “*Augmented Lagrange Multiplier*” ou “*Multiplier Methods*”, e em português M.L.A. - Método do Lagrangeano Aumentado. Este método pode ser visto como uma combinação do uso de multiplicadores de Lagrange com as funções de penalizações.

Primeiramente, é apresentado o método para as condições de restrições de igualdade. Em seguida, é feita uma extensão para o problema com restrição de desigualdade. Finalmente, a combinação de ambos para a solução de um problema genérico de otimização com restrição.

I. Problemas com Restrição de Igualdade

O M.L.A. foi inicialmente desenvolvido para problemas com restrição de igualdade cuja forma é expressa como:

Minimizar

$$f(x) \tag{2.71}$$

sujeito à

$$h_p(x) = 0 ; \quad p = 1, \dots, l \tag{2.72}$$

Define-se então a função Lagrangeana associado ao problema com a forma:

$$L(x, \lambda) = f(x) + \sum_{p=1}^l \lambda_p h_p(x) \tag{2.73}$$

Nota-se que se existir algum λ^* , para o qual x^* é solução do problema sem restrição, tal que,

$$\min_x L(x, \lambda^*), \tag{2.74}$$

enquanto for satisfeito a condições $h_p(x)=0$, cujo x^* é solução do problema descrito pelas equações (2.71) e (2.72). Portanto o problema

$$\min_x L(x, \lambda), \quad \lambda \text{ conveniente}, \tag{2.75}$$

sujeito à

$$h_p(x) = 0 ; \quad p = 1, \dots, l \tag{2.76}$$

equivale ao problema original. Agora fazendo uso do Método da Penalidade Exterior, incorporando à função Lagrangeana o termo de penalidade, permitindo criar uma função pseudo-objetiva expressa por:

$$\Psi(x, \lambda, r) = f(x) + \sum_{p=1}^l \lambda_p h_p(x) + \frac{1}{r} \cdot \sum_{p=1}^l [h_p(x)]^2 ; \quad r \rightarrow 0 \quad (2.77)$$

que é denominada função Lagrangeana aumentada. Portanto, dado λ e r , a solução do problema sem restrição $x(\lambda, r)$, consiste em aplicar um algoritmo que resolva a equação (2.77). Os parâmetros λ e r são convenientemente reajustados e o processo segue iterativamente até a convergência.

A função Lagrangeana aumentada apresentada possui algumas características, que podem ser citadas:

1) Se $\lambda_p = 0$, então a equação (2.77) reduz-se ao Método de Penalidade Exterior. Ao descrever r gradualmente a seqüência de soluções $x(r) \rightarrow x^*$, as soluções intermediárias não são viáveis e de um modo geral, o problema fica mal condicionado se $r < r_{critico}$. Portanto, é utilizado um valor mínimo para o parâmetro de penalização: $r_{min.} = r_{critico}$.

2) Se $\lambda_p = \lambda_p^*$, onde λ_p^* é o vetor dos multiplicadores de Lagrange associados ao critério de otimalidade em x^* , isto é, as condições necessárias de Kuhn-Tucker (KT), então o mínimo da função (2.77) independe do valor de r . Assim a condição de otimalidade em x^* é

$$\nabla_x \Psi(x, \lambda^*, r) \Big|_{x^*} = \underbrace{\nabla_x f(x^*) + \sum_{p=1}^l \lambda_p^* \nabla_x h_p(x^*)}_{KT} + \frac{2}{r} \cdot \sum_{p=1}^l h_p(x^*) \nabla_x h_p(x^*) = 0 \quad (2.78)$$

Estas características permitem avaliar de um modo prático a questão quanto a atualização de λ_p , até atingir-se λ_p^* . Como foi visto anteriormente, a determinação do mínimo depende da escolha do multiplicador de Lagrange, de modo a não se necessitar de um valor muito baixo para r , que levaria ao mal condicionamento. O método é baseado em uma estimativa de λ , quando a estimativa for boa, o mínimo é atingido sem aplicar um valor muito baixo para r . Portanto, para

obter uma estimativa para os multiplicadores de Lagrange compara-se a condição de otimalidade da função (2.77), dada por

$$\nabla_x \Psi(x, \lambda, r) = \nabla_x f(x) + \sum_{p=1}^l \left(\lambda_p + \frac{2}{r} h_p(x) \right) \cdot \nabla_x h_p(x) = 0 \quad (2.79)$$

Com a condição de estacionaridade de $L(x, \lambda)$ em (x^*, λ^*) dada por:

$$\nabla_x L(x^*, \lambda^*) = \nabla_x f(x^*) + \sum_{p=1}^l \lambda_p^* \cdot \nabla_x h_p(x^*) = 0 \quad (2.80)$$

e obtendo, no limite onde $x(\lambda, r) \rightarrow x^*$, a expressão

$$\lambda_p + \frac{2}{r} \cdot h_p(x) \rightarrow \lambda_p^* \quad (2.81)$$

Com esta observação Hestenes [1969], sugeriu a seguinte expressão como estimativa para λ_p , associados as restrições de igualdade.

$$\lambda_p^{k+1} = \lambda_p^k + \frac{2}{r^k} \cdot h_p(x^k); \quad p = 1, \dots, l \quad (2.82)$$

II. Problemas com Restrição de Desigualdade

O problema considerado é:

Minimizar

$$f(x) \quad (2.83)$$

sujeito à :

$$g_j(x) \leq 0; \quad j = 1, \dots, m \quad (2.84)$$

onde o conjunto de restrições inclui também as restrições laterais, equação (2.4).

Pode-se no entanto, reescrever o problema de forma equivalente a um problema com restrição de igualdade. Esta conversão é realizada mediante a introdução de uma variável dita variável de relaxação (folga) na condição de restrição. Logo, o problema toma a forma:

minimizar

$$f(x) \tag{2.85}$$

sujeito a:

$$g_j(x) + z_j^2 = 0 \quad ; \quad j = 1, \dots, m \tag{2.86}$$

onde, z_j é a variável de relaxação associada à restrição de desigualdade.

Deste modo tem-se agora um problema na forma de um problema com restrição de igualdade. Com base nos resultados associados as restrição de igualdade da equação (2.77), podem ser aplicados as restrição de desigualdade para gerar uma nova função lagrangeana aumentada na forma :

$$\Psi(x, \lambda, z, r) = f(x) + \sum_{j=1}^m \lambda_j (g_j(x) + z_j^2) + \frac{1}{r} \sum_{j=1}^m (g_j(x) + z_j^2)^2, \tag{2.87}$$

cujo $\lambda_j \geq 0$, é o multiplicador de Lagrange associado a restrição de desigualdade. Pode-se notar que houve um aumento considerável no número de variáveis de projeto com inclusão da variável de relaxação z_j . Porém, a imposição de estacionaridade da equação (2.87) com relação a z_j , permite a eliminação desta variável. Para fazer isto, aplica-se a condição necessária de ótimo para a função Lagrangeana aumentada derivando-a em relação a z_j então:

$$\nabla_{z_j} \Psi(x, \lambda, z, r) = 0 \quad ; \quad j = 1, \dots, m \tag{2.88}$$

de onde se conclui que

$$z_j^2 = 0 \quad \text{ou} \quad z_j^2 = -\frac{r\lambda_j}{2} - g_j(x) \tag{2.89}$$

Portanto, se obtém

$$z_j^2 = \max \left[0 ; -\frac{r\lambda_j}{2} - g_j(x) \right] \quad (2.90)$$

Esta condição satisfaz a condição necessária de otimalidade e z_j torna-se uma variável independente. Pode-se então eliminar z_j de $\Psi(x, \lambda, z, r)$, utilizando a equação (2.90), somando $g_j(x)$ em ambos os lados, tal que,

$$g_j(x) + z_j^2 = \max \left[g_j(x) ; -\frac{r\lambda_j}{2} \right] ; \quad j = 1, \dots, m \quad (2.91)$$

Fazendo a substituição da equação (2.91) na equação (2.87), se obtém:

$$\Psi(x, \lambda, r) = f(x) + \frac{1}{r} \sum_{j=1}^m M_j(x, \lambda_j, r) \quad (2.92)$$

na qual,

$$M_j(x, \lambda_j, r) = \begin{cases} g_j(x) (g_j(x) + \lambda_j r) & ; \quad \text{se } g_j(x) \geq -\frac{r\lambda_j}{2} \\ -\left(\frac{r\lambda_j}{2}\right)^2 & ; \quad \text{se } g_j(x) < -\frac{r\lambda_j}{2} \end{cases} \quad (2.93)$$

A equação (2.92) possui derivadas de primeira ordem contínuas com relação a x , assim $\Psi(x, \lambda, r)$ é resolvida como no caso do problema com restrição de igualdade.

O multiplicador de Lagrange é atualizado pela expressão

$$\lambda_j^{k+1} = \max \left[0, \lambda_j^k + \frac{2}{r^k} g_j(x^k) \right] ; \quad j = 1 \dots m \quad (2.94)$$

III. Problema Genérico

A combinação dos dois últimos problemas, isto é, restrição de igualdade e restrição de desigualdade permite criar o Método do Lagrangeano Aumentado para o caso genérico expresso por

$$\Psi(x, \lambda, r) = f(x) + \frac{1}{r} \sum_{j=1}^m M(x, \lambda_j, r) + \sum_{p=1}^l \left\{ \lambda_{p+m} h_p(x) + \frac{1}{r} (h_p(x))^2 \right\} \quad (2.95)$$

E a atualização dos multiplicadores de Lagrange, são dados por

$$\lambda_j^{k+1} = \max \left[0, \lambda_j^k + \frac{2}{r^k} g_j(x^k) \right] \quad ; \quad j = 1 \dots m \quad (2.96)$$

$$\lambda_{p+m}^{k+1} = \lambda_{p+m}^k + \frac{2}{r^k} \cdot h_p(x^k) \quad ; \quad p = 1, \dots, l \quad (2.97)$$

A figura 2.6, apresenta o fluxograma para o algoritmo do Método do Lagrangeano Aumentado genérico. Vanderplaats [1984], destaca algumas características atrativas do método:

- 1- o método é relativamente independente do valor de r , isto é, não é necessário que r tenda a zero;
- 2- precisar $g_j(x)=0$ e $h_p(x)=0$ é possível;
- 3- a aceleração da convergência é realizada pela atualização dos multiplicadores de Lagrange;
- 4- o ponto inicial pode estar em qualquer região viável ou inviável;
- 5- no ponto ótimo $\lambda_j^* \neq 0$, automaticamente identifica o conjunto de restrições ativas.

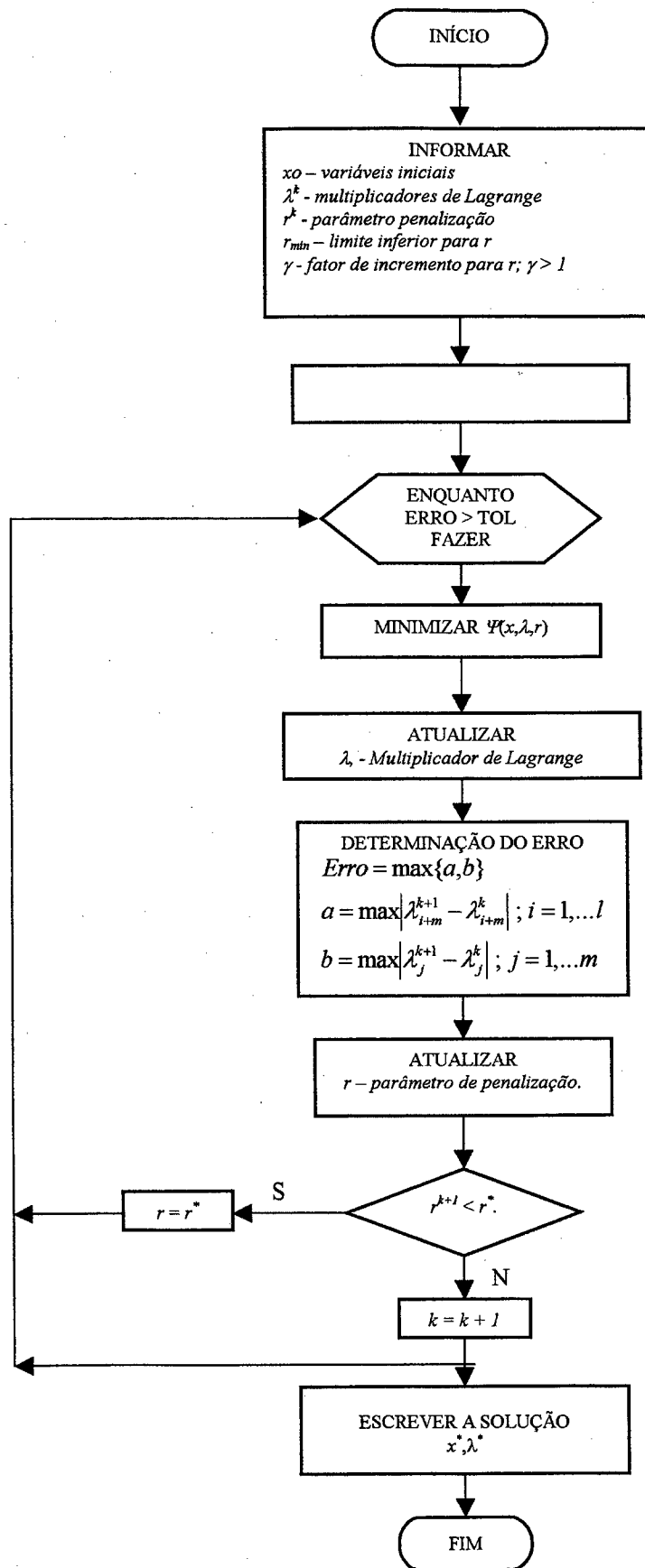


Figura 2.6 – Fluxograma para o Método do Lagrangeano Aumentado.

Capítulo 3

Análise de sensibilidade

3.1 Introdução

Atualmente, o Método de Elementos Finitos (MEF) tornou-se a técnica mais popular no ambiente de engenharia para soluções de problemas complexos. Inicialmente, quase todo o desenvolvimento do MEF foi direcionado para problemas específicos de análise estrutural, entretanto o uso da técnica expandiu-se para aplicações nos mais diversos ramos da engenharia. Hoje com o advento dos computadores digitais, o MEF representa uma das ferramentas de análise mais poderosas usada praticamente em todos os campos da engenharia. O MEF é um procedimento numérico para resolver problemas da mecânica do contínuo com precisão aceitável num ambiente industrial. O MEF tem uma aplicação bastante intensa na modelagem matemática de sistemas físicos permitindo calcular as suas respostas, quando submetido a alguma variação. O uso do Método de Elementos Finitos em análise estrutural fornece uma ferramenta bastante confiável para execução de projetos estruturais. O MEF é usado para identificar problemas técnicos, mas também dá ao engenheiro uma ajuda para identificar meios de modificar o projeto para eliminar problemas ou melhorar a qualidade. Usando a informação da análise de sensibilidade que pode ser gerada por métodos que exploram a formulação do MEF, o engenheiro pode realizar uma análise sistemática de otimização e melhorar o projeto. Na discussão relativa a análise de sensibilidade de projeto, adota-se o MEF, como o modelo matemático para resolver os problemas da elasticidade linear.

Para os algoritmos de otimização, é requerida uma análise de como se comporta a estrutura ou componente diante das mudanças ocorridas nas variáveis de projeto. Esta análise

permite ao engenheiro de projeto avaliar qual a melhor alternativa de soluções de projeto, mediante os valores das variáveis de projeto adotadas. Para a formulação do problema de otimização, em geral, o engenheiro deve estabelecer qual o objetivo (função objetivo), que pode ser, por exemplo, redução do peso de uma peça ou da estrutura. Portanto, para o engenheiro chegar à melhor condição de projeto (projeto ótimo), deve ser determinado um conjunto de variáveis de projeto (x_i), que permita alcançar o objetivo, satisfazendo as restrições impostas ao projeto. As restrições podem ser, por exemplo, processo de fabricação, normas, espaço físico, e o próprio comportamento do material quanto às tensões e deformações, além dos limites estabelecidos às variáveis de projeto.

O comentário acima descreve o problema de otimização definido no capítulo anterior pelas equações (2.1) a (2.5). Para utilizar os algoritmos de otimização, é necessário calcular ao menos os valores da primeira derivada da função objetivo $f(x)$ e funções restrições $g(x)$ com relação as variáveis de projeto (x_i). Estas derivadas são conhecidas na vasta literatura como análise de sensibilidade. A análise de sensibilidade é conceituada segundo Haug, Choi, Konkov [1986], como a relação entre as variáveis de projeto que os engenheiros têm a sua disposição e a resposta estrutural ou das variáveis de estado que são determinadas pelas leis da mecânica. A atenção do presente trabalho é restrita à mecânica estrutural linear elástica.

3.2 O Método de Elementos Finitos

Para apresentar os conceitos matemáticos básicos que envolvem o MEF, isto é, que envolvem a transformação do problema de equilíbrio na sua forma original (formulação forte) para uma forma simplificada (formulação fraca), faz-se uso das equações definidas na teoria da elasticidade linear, que pode ser encontrada nas várias literaturas indicadas na referência [25], [33].

O primeiro passo é formular o problema de valor de contorno para elasticidade, como segue. Dado $f: \Omega \rightarrow \mathbb{R}$, $t: \Gamma_T \rightarrow \mathbb{R}$, encontrar $u_i: \Omega \rightarrow \mathbb{R}$, tal que:

$$\operatorname{div} \mathbf{T}(\mathbf{x}) + \mathbf{f}(\mathbf{x}) = \mathbf{0} ; \quad \forall \mathbf{x} \in \Omega \quad (3.1)$$

$$\bar{\mathbf{t}} = \mathbf{T} \cdot \hat{\mathbf{n}} ; \quad \forall \mathbf{x} \in \Gamma_T \quad (3.2)$$

$$u_i = \bar{u} ; \quad \forall \mathbf{x} \in \Gamma_u \quad (3.3)$$

Donde pode-se verificar a equação de equilíbrio (3.1) e as condições de contorno (3.2) e (3.3), dadas em termos do vetor unitário normal $\hat{\mathbf{n}}$, componentes do vetor tração $\bar{\mathbf{t}}$, e pelo vetor deslocamento u . A figura 3.1, permite uma visualização de um corpo genérico ocupando um volume Ω , delimitado por uma superfície Γ . O contorno é particionado em duas regiões Γ_u , na qual o deslocamento é prescrito, como indicado na equação (3.3), e Γ_T , no qual a tração é prescrita equação (3.2).

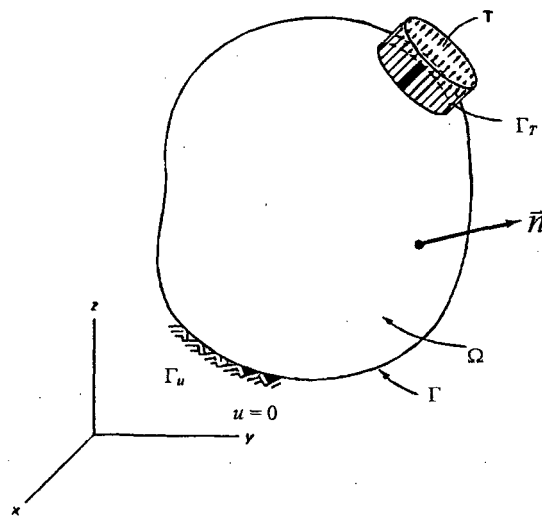


Figura 3.1 – Sólido genérico e condições de contorno – Belegundu & Chandrupatla.

Para que o problema tenha solução deve-se eliminar os movimentos de corpo rígido. E a equação de equilíbrio, com as condições de contorno, deve ser satisfeita em todo ponto do domínio Ω .

3.2.1 Princípios Matemáticos para o MEF

Definido a formulação forte com as condições de contorno prescritas para um corpo genérico, pode-se agora avaliar os métodos alternativos para encontrar as soluções. Para

problemas com simplicidade geométrica, por exemplo, um modelo de barras, a integração analítica é muito simples, no entanto, a grande maioria dos casos reais onde se tem um domínio geométrico arbitrário, o processo analítico é impraticável. Neste sentido, o método numérico surge para viabilizar a solução destes tipos de problemas. O método aqui aplicado é o método de elementos finitos, que permite a aproximação de soluções de diversos tipos de equações diferenciais que descrevem ou modelam matematicamente problemas físicos da mecânica do contínuo. Para utilizar o método de elementos finitos é interessante analisar o problema de equilíbrio através de uma formulação matemática denominada de princípio dos trabalhos virtuais (PTV), e mais precisamente com o Método de Galerkin.

I. Princípios dos Trabalhos Virtuais para Elasticidade Tridimensional.

A idéia consiste em transformar a formulação forte descrito nas equações (3.1), (3.2) e (3.3) em uma formulação fraca ou integral do problema de equilíbrio. Ao contrário do problema original (formulação forte), onde se procura o conjunto de soluções que satisfizessem a equação diferencial de equilíbrio, necessita-se agora soluções que satisfaçam uma equação integral de equilíbrio denominada Princípio dos Trabalhos Virtuais. Este princípio está interessado na relação que existe entre o conjunto de forças externas atuando no corpo e as correspondentes forças internas originadas, que juntas satisfazem a condição de equilíbrio e também com um conjunto de deslocamentos e o correspondente componente de deformação, satisfazendo as condições de compatibilidade da teoria da elasticidade. Em linhas gerais o princípio pode ser assim enunciado:

“Um corpo está em equilíbrio se o trabalho virtual feito pelas forças externas é igual ao trabalho virtual interno absorvido pela estrutura para todo campo de deslocamento cinematicamente admissível.” Belegundu [1991].

Define-se, nesta seção, os seguintes conjuntos:

$$Kin_u = \{u(x) \mid u(x), \text{ suficientemente regular, } u(x) = \bar{u}(x) \text{ em } \Gamma_u\} \text{ e}$$

$$Var = \{v(x) \mid v(x), \text{ suficientemente regular, } v(x) = 0 \text{ em } \Gamma_u\},$$

que são respectivamente, o conjunto dos deslocamentos admissíveis e o conjunto das variações dos deslocamentos admissíveis. Então, a integração da equação (3.1) é dada por:

$$\int_{\Omega} [\text{div} T(x) + f(x)] \cdot v(x) d\Omega = 0, \quad \forall v(x) \in Var \quad (3.4)$$

usando a regra do produto para a integral tem-se

$$\int_{\Omega} \text{div}(T^T \cdot v) d\Omega - \int_{\Omega} T \cdot \nabla v d\Omega + \int_{\Omega} f \cdot v d\Omega = 0, \quad \forall v(x) \in Var \quad (3.5)$$

usando o teorema da divergência e como o tensor de tensão T (tensor tensão de Cauchy) é simétrico, e

$$T \cdot \nabla v = T \cdot \varepsilon(v), \quad (3.6)$$

logo,

$$\int_{\Gamma} T \bar{n} \cdot v d\Omega - \int_{\Omega} T \cdot \varepsilon(v) d\Omega + \int_{\Omega} f \cdot v d\Omega = 0, \quad \forall v(x) \in Var \quad (3.7)$$

Porém, como $v(x) \in Var$ e $v(x) = 0$ em Γ_u , logo

$$\int_{\Gamma} T \bar{n} \cdot v d\Omega = \int_{\Gamma_r} T \bar{n} \cdot v d\Gamma \quad (3.8)$$

Por outro lado, sabe-se que $T \bar{n} = \bar{t}$, o que implica

$$\int_{\Gamma_r} \bar{t} \cdot v d\Gamma + \int_{\Omega} f \cdot v d\Omega = \int_{\Omega} T \cdot \varepsilon(v) d\Omega, \quad \forall v(x) \in Var \quad (3.9)$$

formulando assim o princípio dos trabalhos virtuais, tal que,

trabalho das forças externas:

$$W_E = \int_{\Gamma_r} \bar{t} \cdot v d\Gamma + \int_{\Omega} f \cdot v d\Omega, \quad \forall v(x) \in Var \quad (3.10)$$

trabalho das forças internas:

$$W_I = \int_{\Omega} T \cdot \varepsilon(v) d\Omega, \quad \forall v(x) \in Var \quad (3.11)$$

Para o problema da elasticidade linear, introduz-se a equação constitutiva, para um sólido elástico linear e isotrópico. Assim, dado um deslocamento $u(x)$ no corpo, é possível determinar a deformação $\varepsilon(u(x))$ e a tensão deste corpo através das relações constitutivas que são expressas por:

$$T(u(x)) = 2 \cdot \nu \cdot \varepsilon(u(x)) + \beta \cdot (tr \varepsilon) \cdot I \quad (3.12)$$

na qual

$$\nu = \frac{E}{2(1+\eta)} \quad ; \quad \beta = \frac{\eta \cdot E}{(1+\eta) \cdot (1-2\eta)} \quad (3.13)$$

são os parâmetros de Lamé, com o módulo de elasticidade E e coeficiente de Poisson η , incorporados na expressão. Outra forma usual de escrever a relação constitutiva é

$$T(u(x)) = C \cdot \varepsilon(u(x)) = C \left(\frac{\nabla u(x) + \nabla u(x)^T}{2} \right) = C \cdot \nabla^s u(x) \quad (3.14)$$

onde C é o tensor constitutivo da teoria da elasticidade, com o módulo de elasticidade e coeficiente de Poisson nele incorporados, tal que,

$$C = \frac{E}{(1+\eta) \cdot (1-2\eta)} \begin{bmatrix} (1-\eta) & \eta & \eta & 0 & 0 & 0 \\ \eta & (1-\eta) & \eta & 0 & 0 & 0 \\ \eta & \eta & (1-\eta) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\eta)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\eta)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\eta)}{2} \end{bmatrix} \quad (3.15)$$

$$\varepsilon = \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_1} & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \quad (3.16)$$

Portanto, o problema de equilíbrio se resume em procurar o deslocamento $u(x)$, que produz um estado de tensão interno $T(u(x))$, que equilibra os esforços externos (f, \bar{t}) . Formalmente, a formulação fraca do problema de equilíbrio é escrita na forma:

Dado $f: \Omega \rightarrow R$, $\bar{u}: \Gamma_u \rightarrow R$ e $\bar{t}: \Gamma_T \rightarrow R$, determinar $u(x) \in Kin$, tal que,

$$\int_{\Gamma_T} \bar{t} \cdot v \, d\Gamma + \int_{\Omega} f \cdot v \, d\Omega = \int_{\Omega} C \cdot \varepsilon(u) \cdot \varepsilon(v) \, d\Omega, \quad \forall v(x) \in Var \quad (3.17)$$

II. Método de Galerkin

Será analisado agora o método de obtenção de uma solução aproximada do problema real de valor do contorno. Inicialmente, formulou-se a condição forte e fraca do problema de equilíbrio, baseado no PTV, o qual é base para a formulação matemática do método dos elementos finitos.

Se for analisado o conjunto Var , verifica-se duas propriedades fundamentais, primeiramente, Var é um espaço linear de funções. Em segundo, é que Var tem dimensão infinita, ou seja, são necessárias infinitas funções para representar o conjunto Var . O mesmo conceito serve para o conjunto Kin , já que, $Kin_u = Var_u + \{u_p\}$, onde u_p é uma função particular de Kin_u . Basicamente, a ideia do método é aproximar a solução por um conjunto de dimensão finita Var^h . Seja $v^h(x) \in Var^h$, então

$$v^h(x) = \sum_{i=1}^n v_i \cdot \varphi_i(x) ; \quad v_i \in R \text{ arbitrário} \quad (3.18)$$

As funções φ_i , definem o subespaço n-dimensional Var^h de Var , onde cada $v^h(x)$ em Var^h , é determinado por uma combinação linear das funções base $\varphi_i(x) \in Var$. Considerando o método de Galerkin para determinação de soluções aproximadas para o problema do valor de contorno, baseado na formulação fraca definida pela equação (3.17), então $u(x) = u^h(x) + w(x)$, na qual $u^h(x) \in Var^h$ e $w(x) \in Kin^h$, onde

$$u^h(x) = \sum_{i=1}^n u_i \cdot \varphi_i(x) \quad (3.19)$$

Desta forma o problema pode ser descrito na forma:

Seja $u(x) = u^h(x) + w(x)$, onde $w(x) \in Kin^h$ é conhecido. O problema consiste então em determinar $u^h(x) \in Var^h$, tal que,

$$\int_{\Omega} C \cdot \varepsilon(u^h) \cdot \varepsilon(v^h) d\Omega = \int_{\Gamma_r} \bar{t} \cdot v^h d\Gamma + \int_{\Omega} f \cdot v^h d\Omega - \int_{\Omega} C \cdot \varepsilon(w) \cdot \varepsilon(v^h) d\Omega , \quad \forall v^h(x) \in Var^h \quad (3.20)$$

o problema então consiste na determinação de $u_i \in R$.

Uma outra forma de se representar a equação (3.20) é através da forma clássica de representação matricial. Pode-se então reescrever a equação (3.16) na forma

$$\varepsilon(u^h) = \nabla u^h = DNU \quad (3.21)$$

onde, N representa a matriz das funções base $\varphi_i(x) \in Var$, que dependem da formulação do elemento finito utilizado, U é o vetor deslocamento e D representa a matriz de gradientes dada por

$$D = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \end{bmatrix} \quad (3.22)$$

logo, se

$$B = D.N \quad (3.23)$$

onde, B é a matriz que representa o gradiente do conjunto das funções base $\varphi_i(x) \in Var$, então pode-se reescrever a equação (3.21) na forma

$$\varepsilon(u^h) = \nabla u^h = DNU = BU \quad (3.24)$$

do mesmo modo para v^h se tem

$$\varepsilon(v^h) = \nabla v^h = DNV = BV \quad (3.25)$$

Portanto, fazendo as devidas substituições na equação (3.20), chega-se ao seguinte problema:

Dado $f: \Omega \rightarrow R$, $\bar{u}: \Gamma_u \rightarrow R$ e $\bar{t}: \Gamma_T \rightarrow R$, determinar $u(x) \in R$, tal que,

$$\int_{\Omega} C.B.U.B.V d\Omega = \int_{\Gamma_T} \bar{t}.N.V d\Gamma + \int_{\Omega} f.N.Vd\Omega - \int_{\Omega} C.\varepsilon(w).B.V d\Omega, \quad \forall V \in R^n \quad (3.26)$$

isolando V e reordenando os termos da equação integral chega-se a forma:

$$\underbrace{\left[\int_{\Omega} B^T \cdot C \cdot B \, d\Omega \right]}_K \cdot U = \underbrace{\int_{\Gamma_T} N^T \cdot \bar{t} \, d\Gamma + \int_{\Omega} N^T \cdot f \, d\Omega - \int_{\Omega} B^T \cdot C \cdot \varepsilon(w) \, d\Omega}_F \quad (3.27)$$

Assim, resumidamente tem-se:

$$K \cdot U = F \quad (3.28)$$

onde, a matriz $K=[k_{ij}]$ é usualmente referida como matriz de rigidez, o vetor F é conhecido como vetor de carga e U o vetor deslocamento.

Para a consideração de problemas não lineares, somente ocorrerá alteração na matriz K , que se apresenta de forma genérica na equação (3.28), já que podem ser incorporados elementos que determinam a não linearidade do material, ou ainda a não linearidade geométrica, e também no vetor de forças pode ocorrer alteração. Aqui se procurou dar uma idéia básica sobre a matemática que envolve muito dos programas comerciais de elementos finitos, além do que, para abordar os aspectos relativos aos gradientes de funções e análise de sensibilidade no processo de otimização é importante ter também estes conceitos. Para maiores detalhes acerca deste tema pode-se encontrar uma vasta bibliografia, dentre as quais são citadas algumas nas referências deste trabalho.

3.3 Gradiente da Função Objetivo e Restrições

Quando se adota, por exemplo, como função custo o peso total da estrutura, o problema de estabelecer o melhor projeto consiste em determinar as variáveis de projeto de tal modo que a estrutura tenha o menor peso e ainda satisfaça a equação de equilíbrio $K U = F$, que é considerada uma das restrições de igualdade do problema de otimização estrutural. Existem ainda outras restrições que são devido a deslocamentos, deformações e tensões. Assim, para resolver um problema de mínimo necessita-se determinar ao longo do processo iterativo as seguintes informações: $f(x)$ e $g(x)$, e seus respectivos, $\nabla f(x)$ e $\nabla g(x)$. Pode-se então fazer uso de técnicas de programação matemática para obter uma rotina sistemática de cálculo que resolva o problema de otimização. Para o cálculo das derivadas existem alguns métodos os quais se

classificam em analíticos, semi-analíticos e numéricos. Este processo de determinação da resposta do sistema com relação as variáveis de projeto é conhecido como Análise de sensibilidade.

A seguir são descritos alguns dos métodos mais usuais para análise de sensibilidade, que são: método direto, método adjunto, semi-analítico e o método das diferenças finitas. Este último tem sua importância neste trabalho pela simplicidade de implementação e flexibilidade com relação a equação de estado, o que torna-o bastante genérico. Inicialmente, é dado de forma genérica uma abordagem dos métodos analíticos e o semi-analíticos por sua importância no contexto.

3.3.1 Método Direto

O método direto é classificado como um método analítico e para ilustrá-lo é tomado como ponto de partida a equação de estado definida anteriormente na equação (3.28), onde estão embutidos os parâmetros físicos, geométricos, e mecânicos. Assim como, a matriz de rigidez, onde estão todas as características da estrutura, que depende das variáveis de projeto, o vetor carga, também pode depender das variáveis de projeto, por exemplo, quando é considerado o peso próprio da estrutura. Deste modo pode-se reescrever a equação (3.28) como

$$K(x) \cdot U(x) = F(x) \quad (3.29)$$

A discussão é iniciada através da definição da função objetivo do problema, que se apresenta, genericamente da seguinte forma

$$f(x) = f(x, U(x)) \quad (3.30)$$

Usando a regra da cadeia para a diferenciação da equação (3.30) para obter o gradiente da função objetivo, então

$$\frac{df(x_i)}{dx_i} = \frac{\partial f(x_i)}{\partial x_i} + \frac{\partial f(x_i)}{\partial U} \frac{\partial U}{\partial x_i} ; \quad i = 1, \dots, m \quad m - \text{núm. de variáveis de projeto} \quad (3.31)$$

reescrevendo tem-se

$$\nabla f = \frac{\partial f}{\partial x} + [\nabla U]^T \frac{\partial f}{\partial U} \quad (3.32)$$

Logo, para calcular o gradiente da função objetivo é necessário calcular o gradiente dos deslocamentos. Para calcular $\nabla_x U$, isto é, a resposta de U com relação as variáveis de projeto x_i , utiliza-se a equação de estado definida pela equação (3.29), e diferenciando ambos os lados desta equação com relação as variáveis de projeto produz-se

$$[K(x_i)] \left\{ \frac{\partial U}{\partial x_i} \right\} + \left[\frac{\partial K(x_i)}{\partial x_i} \right] U = \left[\frac{\partial F}{\partial x_i} \right] \quad (3.33)$$

reordenando os termos fica

$$[K(x_i)] \left\{ \frac{\partial U}{\partial x_i} \right\} = \left\{ \frac{\partial F}{\partial x_i} \right\} - \left[\frac{\partial K(x_i)}{\partial x_i} \right] U, \quad (3.34)$$

pode-se então verificar que ∇U é dado por

$$\nabla_{x_i} U = \left\{ \frac{\partial U}{\partial x_i} \right\} = [K(x_i)]^{-1} \left[\left\{ \frac{\partial F}{\partial x_i} \right\} - \left[\frac{\partial K(x_i)}{\partial x_i} \right] \cdot U \right] \quad (3.35)$$

onde, $[K(x)]^{-1}$ é o inverso do operador linear (matriz de rigidez global em análise estrutural). O método direto consiste em resolver o sistema de equações dado pela equação (3.34). A solução do sistema expresso por $\frac{\partial U}{\partial x_i}$, juntamente com os termos $\frac{\partial f}{\partial U_k}$ e $\frac{\partial f}{\partial x_i}$, permite determinar a derivada total de $f(x)$. O mesmo deve ser feito para $g_j; j = 1, \dots, n$ (número de restrições).

Para obter-se as expressões analíticas para o cálculo dos gradientes é requerido detalhe das expressões da matriz de rigidez da equação de estado. Entretanto, em muitos programas comerciais esta informação detalhada da estrutura do programa não está disponível para o usuário. Além do mais, a estrutura da equação (3.29) são muitas vezes apresentadas de forma complexa o que torna inviável para a proposta deste trabalho. Portanto, o cálculo exato pela derivada analítica utilizando o método direto é tipicamente implementado por desenvolvedores de programas de elementos finitos, os quais tem profundo conhecimento dos detalhes da

estrutura do programa. Um outro método analítico, o método adjunto, abordado a seguir, também apresenta esta mesma restrição para uso no procedimento proposto.

3.3.2 Método Adjunto

Seja μ a variável adjunta, tomando como base a equação (3.32) e $g(x, U(x))$ a função restrição, então para determinar $\nabla g(x)$, parte-se da equação (3.29), cuja diferenciação é dada pela equação (3.33), da qual após reordenar os termos se obtém a expressão

$$[K(x_i)] \left\{ \frac{\partial U}{\partial x_i} \right\} = \left\{ \frac{\partial F}{\partial x_i} \right\} - \left[\frac{\partial K(x_i)}{\partial x_i} \right] U \quad (3.36)$$

então, com a inclusão da variável adjunta se tem

$$\left\langle \mu, [K] \left\{ \frac{\partial U}{\partial x_i} \right\} \right\rangle = \left\langle \mu, \left\{ \frac{\partial F}{\partial x_i} \right\} \right\rangle - \left\langle \mu, \left[\frac{\partial K}{\partial x_i} \right] U \right\rangle ; i = 1 \dots m \text{ (num. variáveis de projeto)} \quad (3.37)$$

portanto,

$$\left\langle [K]^T \mu, \left\{ \frac{\partial U}{\partial x_i} \right\} \right\rangle = \left\langle \mu, \left\{ \frac{\partial F}{\partial x_i} \right\} \right\rangle - \left\langle \mu, \left[\frac{\partial K}{\partial x_i} \right] U \right\rangle ; i = 1 \dots m \quad (3.38)$$

o que se quer determinar é

$$\frac{dg_j(x_i)}{dx_i} = \frac{\partial g_j(x_i)}{\partial x_i} + \frac{\partial g_j(x_i)}{\partial U} \frac{\partial U}{\partial x_i} ; i = 1, \dots, n ; \text{ núm. de variáveis de projeto.} \quad (3.39)$$

$j = 1, \dots, m ; \text{ núm. de inequações.}$

Sendo assim, pode-se definir que

$$[K]^T \mu^j = - \frac{\partial g_j}{\partial U} \Rightarrow \left\langle \frac{\partial g_j}{\partial U}, \frac{\partial U}{\partial x_i} \right\rangle \equiv - \left\langle [K]^T \mu^j, \frac{\partial U}{\partial x_i} \right\rangle = - \left\langle \mu^j, \frac{\partial F}{\partial x_i} \right\rangle + \left\langle \mu^j, \left[\frac{\partial K}{\partial x_i} \right] U \right\rangle \quad (3.40)$$

logo a equação (3.39), torna-se

$$\frac{dg_j}{dx_i} = \frac{\partial g_j}{\partial x_i} - \mu_s^j \left\{ \frac{\partial F_s}{\partial x} \right\} + \mu_s^j \left[\frac{\partial K_{sk}}{\partial x_i} \right] U_k ; j = 1, \dots, m$$

$$s = 1, \dots, l$$

$$k = 1, \dots, l$$

$$i = 1, \dots, n$$
(3.41)

mas o que deve ser determinado é

$$[K(x_i)]^T \mu^j = - \frac{\partial g_j}{\partial U}$$
(3.42)

O método adjunto também é classificado como o um método analítico.

Observações:

- Se o número de variáveis de projeto x_i , $i = 1, \dots, n$ é bem menor que o conjunto de restrições definindo a região factível de projeto, então o método direto é mais eficiente;
- Por outro lado, se g_j , onde $j = 1, \dots, m$ é um conjunto pequeno, então o método adjunto é mais eficiente;

A determinação da derivada pelo método analítico é o ideal para obter-se o valor exato da derivada para as funções objetivo e restrições. Como, as funções são implícitas em relação a variável de estado u , isto requer o conhecimento do programa de elementos finitos. Mas, até mesmo nos casos em que as funções objetivo e restrições são explicitamente expressas nas variáveis de estado e variáveis de projeto, a utilização da forma acima para determinar os gradientes das funções pode não ser prático já que se tem que computar a primeira derivada do operador linear (matriz de rigidez global) K da equação de equilíbrio, que pode se apresentar de forma bastante complexa.

3.3.3 Método Semi Analítico

O método semi-analítico faz uso do método analítico e do método de diferenças finitas para determinar o gradiente das funções objetivo e restrições. Este método também faz uso da

equação (3.29), entretanto, a diferenciação da matriz K e do vetor F com relação as variáveis de projeto para determinar $\frac{\partial U}{\partial x_i}$ é feito por diferenças finitas. Portanto, este método tem a vantagem de ser mais fácil de implementar do que o método analítico (direto e adjunto), mas ainda requer o conhecimento de detalhes do programa de elementos finitos. Um exemplo, interessante que ajuda a entender este método pode ser o caso de uma estrutura de casca 3-D. Neste caso a derivada analítica da matriz K com relação a uma coordenada nodal de um nó pode não ser obtida numa forma simples. Para a formulação de casca no método de elemento finito se usa um sistema de coordenada local associada a cada elemento finito que é definido pelas coordenadas locais dos quatros nós dos vértices do elemento. A matriz de rigidez do elemento K_{local}^e é transformada para um K_{global}^e no sistema de coordenadas global através da matriz R , isto é,

$$K_{global}^e = R^T K_{local}^e R \quad (3.43)$$

onde R é a matriz transformação do sistema de coordenadas local para o sistema de coordenadas global. Assim, a primeira derivada da matriz de rigidez é computada por

$$\frac{\partial K_{global}^e}{\partial x_i} = \left(\frac{\partial R}{\partial x_i} \right)^T K_{local}^e R + R^T \frac{\partial K_{local}^e}{\partial x_i} R + R^T K_{local}^e \frac{\partial R}{\partial x_i} \quad (3.44)$$

esta é, a primeira derivada de ambas as matrizes de transformação de coordenadas e de rigidez calculada analiticamente. Entretanto, na prática, não é viável computar a derivada da matriz de rigidez K_{global}^e de forma analítica. Isto conduz ao uso de diferenças finitas (*central*, *“forward”*, ou *“backward”*) para computar a derivada do operador K da equação de estado, tal que,

$$\frac{\partial K}{\partial x_i} = \frac{K|_{x+\Delta x e_i} - K|_{x-\Delta x e_i}}{2 \cdot \Delta x}; \quad D.F. Central \quad (3.45)$$

onde, Δx é uma mudança suficientemente pequena nas variáveis de projeto corrente. Do mesmo modo tem-se para o vetor de carga

$$\frac{\partial F}{\partial x_i} = \frac{F|_{x+\Delta x e_i} - F|_{x-\Delta x e_i}}{2 \cdot \Delta x} \quad (3.46)$$

Esta aproximação leva a um método semi-analítico para computar a sensibilidade de projeto. Então, da equação (3.29), determina-se $\nabla_x U$, e $\nabla_x f$ é obtido por

$$\nabla_{x_i} f = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial U} \cdot \frac{\partial U}{\partial x_i} \approx \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial U} \cdot \left([K(x_i)]^{-1} \cdot \left(\frac{F|_{x_i+\Delta x_i, e_i} - F|_{x_i-\Delta x_i, e_i}}{2 \cdot \Delta x_i} - \frac{K|_{x_i+\Delta x_i, e_i} - K|_{x_i-\Delta x_i, e_i}}{2 \cdot \Delta x_i} \cdot U \right) \right) \quad (3.47)$$

Este método, do mesmo modo que o método direto e adjunto necessita de acesso aos detalhes do código fonte do MEF para computar de forma eficiente o cálculo de $\nabla_x U$, permitindo determinar $\nabla_x f$ e $\nabla_x g$.

3.3.4 Método de Diferenças finitas

Se considerar o que foi exposto nos capítulos anteriores, para o procedimento genérico de otimização integrado ao programa comercial, pode ser muito mais simples aplicar a aproximação por diferenças finitas para computar a sensibilidade das funções objetivo e restrições, pois não se tem a necessidade dos detalhes do código fonte do programa comercial. E também, a forma explícita da função f e g em x e U não é requerida, bem como a primeira derivada do operador linear K da equação de estado não necessita ser calculada. A desvantagem deste método é precisar de no mínimo duas análises do programa comercial por variável de projeto para computar o gradiente da função objetivo, se for utilizado o método “forward” ou “backward”, e três para *diferenças finitas centrais*. Assim, se n funções e m variáveis de projeto estão envolvidos no problema, $(2m+1)n$ análises são necessárias para computar a derivada das funções restrições e objetivo. Se o número de variáveis de projeto é grande como em problemas de otimização dimensional para estruturas aeroespaciais, o método de diferenças finitas não é uma boa prática. Entretanto, se o número de variáveis de projeto é um tanto reduzido, ele se torna poderoso. O que se encontra na prática, é que em muitos problemas de projeto mecânico o número de variáveis de projeto é pequeno, devido ao custo para controle das variações.

É notado que não é preciso nenhum desenvolvimento especial para que o método das diferenças finitas seja aplicado, enquanto os métodos analíticos e semi-analítico necessitam considerável esforço para o desenvolvimento do código e implementação, além da necessidade de conhecer o código de elementos finitos. Além disso, o uso de diferenças finitas permite trabalhar com problemas multidisciplinares, por exemplo, se a forma de uma estrutura é

otimizada pela minimização do arraste gerado pelo fluxo de um fluido externo a estrutura, considerando os efeitos da máxima tensão e deformações da estrutura, existe a necessidade de se calcular os gradientes das funções nas duas diferentes disciplinas, isto é, nas duas diferentes equações de estado. Então, como os programas de análises para tensão e fluido são em geral não integrados num único programa, logo a determinação dos gradientes de f e g pelos métodos analíticos e semi-analítico torna-se muito complexo para atender aos dois problemas, já que são fontes de dados diferentes. Entretanto, o método de diferenças finitas pode ser aplicado para os mais diferentes tipos de equação de estado, juntamente com um programa comercial existente no ambiente de projeto, permitindo determinar diretamente as derivadas das funções objetivo e restrições sem grandes complicações na implementação das rotinas que efetuam os cálculos. A flexibilidade do método de diferenças finitas para o cálculo das derivadas é muito significativa.

Além disso, num ambiente de rede, pode ser possível usar vários computadores simultaneamente, para computar os cálculos das derivadas, Kikuchi e Horimatsu [1994]. Isto implica na possibilidade de usar dois ou três computadores para fazerem as análises ao mesmo tempo, como um processamento paralelo. Por exemplo, em um dos computadores se faria a análise da equação de estado na perturbação do projeto $x+\Delta x$, enquanto um outro computador analisa na configuração $x-\Delta x$. Deste modo, a determinação das derivadas se efetua sem aguardar por um longo tempo que, normalmente, é preciso quando usado somente um computador.

Kikuchi e Horimatsu [1994], enumeram algumas condições nas quais o método de diferenças finitas é o mais apropriado para calcular as derivadas das funções, considerando alguns problemas práticos de otimização estrutural:

- o tempo de computação pode ser excessivo, quando o número de variáveis de projeto é grande, isto é, maior que 10 variáveis de projeto;
- o sistema de otimização de projeto pode ser muito flexível, porque pode ser usado qualquer tipo de programa de análise de engenharia, isto é, para qualquer equação de estado;
- a dimensão do programa a ser desenvolvido para o sistema de otimização de projeto é muito pequena e de fácil manutenção;

- o computador usado para modelagem e modificação de projeto não necessita ser o mesmo usado para os cálculos das derivadas num ambiente de rede.

O gradiente das funções feito pelo método de diferenças finitas, usa o seguinte procedimento:

Seleciona o método de diferenças finitas:

(1) *central*; (2) *forward*.

$x = x^0$

Calcula $f_v = f(x^0)$

Define o incremento: h

Calcula $x_{ph} = x^0 + h$

Caso 1:

Calcula $x_{ps} = x^0 - h$

$h = x_{ph} - x_{ps}$

Loop 1:

$x(j) = x_{ph}(j)$

Calcula $f_h = f(x)$

Calcula $x(j) = x_{ps}(j)$

Calcula $f_s = f(x)$

Calcula $gradf(j) = [f_h - f_s] / h(j)$

$x(j) = x^0(j)$

fim loop1

Caso 2:

$h = x_{ph} - x^0$

Loop 2:

$x(j) = x_{ph}(j)$

Calcula $f_h = f(x)$

Calcula $gradf(j) = [f_h - f_v] / h(j)$

$x(j) = x^0(j)$

Fim loop2

Neste procedimento estão implementados dois dos métodos de diferenças finitas, o método “*forward*” e o central. No primeiro caso, a expressão para determinar a derivada é dada por:

$$\frac{\partial f}{\partial x_i} = \frac{f(x + \Delta x e_i) - f(x)}{\Delta x} ; \quad i = 1, \dots, n \text{ número de variáveis de projeto} \quad (3.48)$$

Conhecido o valor no ponto corrente x_i , este método requer o cálculo da função no ponto $x_i + \Delta x_i$, para encontrar o valor da derivada. Entretanto, para o procedimento de otimização proposto foi

aplicado o método de diferenças finitas central. Pois, embora este método necessite mais tempo de computação, a diferença quanto à precisão, em relação ao método “forward” é significativa. Para calcular a derivada parcial de f por diferenças finitas central é dada a expressão:

$$\frac{\partial f}{\partial x_i} = \frac{f(x + \Delta x e_i) - f(x - \Delta x e_i)}{2 \cdot \Delta x} ; \quad i = 1, \dots, n \text{ número de variáveis de projeto} \quad (3.49)$$

esta expressão requer além do cálculo da função em x_i , que ela seja calculada mais duas vezes, uma vez em $x_i + \Delta x_i$, e outra em $x_i - \Delta x_i$. Portanto, isto requer a chamada do programa comercial 3 vezes para computar o gradiente da função. A preferência por este método se explica pela importância que a direção de descida tem no processo de otimização e como já foi verificado anteriormente, a direção depende fundamentalmente de uma boa precisão no cálculo da derivada numérica.

O valor da perturbação Δx , é muito importante para o cálculo dos gradientes, normalmente, para um valor de 1% se tem trabalhado muito bem. O trabalho de Gill, Murray e Wright [1981], dá maiores detalhes sobre a especificação de valor da perturbação para as variáveis de projeto.

Capítulo 4

Procedimento de Otimização

4.1 Introdução

Ainda hoje, pode-se encontrar nas indústrias metodologias de otimização de projeto sendo trabalhada ainda com o processo manual, sendo muitas vezes baseado na intuição e na experiência de pessoas, como também através de testes experimentais, figura 4.1. Sabemos também que a redução de custo se faz presente a todo instante, bem como a qualidade, isto é, procura-se cada vez mais desenvolver produtos com menor custo e de boa qualidade e no menor tempo.

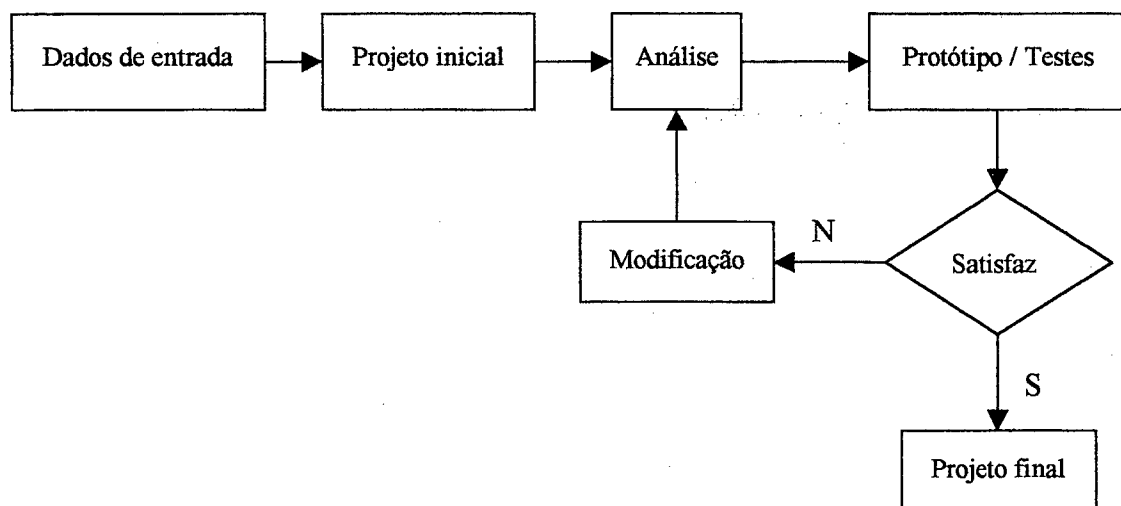


Figura 4.1 – Processo de otimização tradicional.

Neste sentido, comete-se alguma contradição, pois para conseguir os objetivos almejados pela indústria, não se pode continuar a trabalhar com base na intuição, mas sim sobre dados e fatos concretos e cientificamente fundamentados. Testes são necessários para certificar-se do que está errado e para validação de modelo, mas quando usado como meio principal para obter respostas sobre cada alteração intuitiva proposta em um projeto, este método de trabalho acaba tornando-se inviável devido a quantidade de protótipos que garanta uma estatística confiável, tempo de resposta e conseqüentemente pelo custo.

Sabe-se também que na indústria pode-se encontrar muitos programas que o engenheiro usa para auxiliá-lo no desenvolvimento de peças, porém estas ferramentas encontram-se muitas vezes dispersas trabalhando de forma isolada. No início da década passada a disponibilidade de programas comerciais de elementos finitos e métodos de otimização, embora ainda limitada, permitiu a inicialização destes recursos matemáticos para o desenvolvimento de projeto no meio industrial. Portanto, muito se investiu na aquisição de sofisticados programas. Muito destes programas foram adquiridos pelas indústrias, que montaram toda uma infra-estrutura baseada nos recursos existentes naquele momento, sendo que hoje seria inviável sua troca, o que torna mais viável o desenvolvimento de um procedimento de otimização de projeto que possa utilizar a tecnologia existente, tal que não seja requerido um custo muito alto, bem como esforço adicional para familiarização com novos programas de elementos finitos. Outro fator é que cada organização pode ter sua própria preferência e larga experiência no uso de um determinado programa comercial de elementos finitos, e o procedimento de otimização de projeto pode ser capaz de integrar-se este programa já existente e em uso na indústria. Isto significa que o sistema teria facilidade de adaptar-se, mediante utilização de algumas rotinas, que permitem adequar-se a um programa qualquer, tendo toda flexibilidade possível, além do baixo custo de implementação.

Neste ambiente onde se tem um processo manual, mesmo com tecnologias sofisticadas, o processo de otimização torna-se impraticável. Assim, não existe garantia que uma alteração no projeto do componente, baseada em uma análise intuitiva forneça uma peça melhor em relação à original, ou no caso de um novo projeto se ele é confiável ou não, isto é, se a resistência poderia ter sido aumentada, ou que o peso (custo) poderia ter sido substancialmente diminuído.

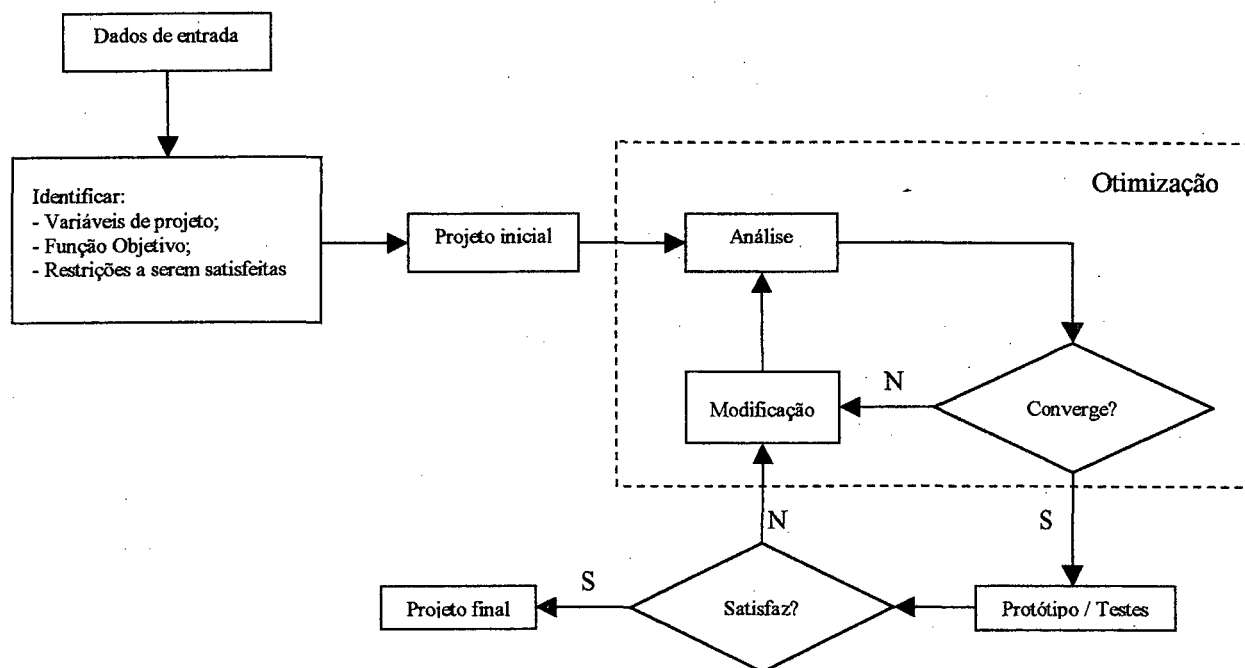


Figura 4.2 – Processo de otimização integrado.

Quando do desenvolvimento de um processo automático de otimização, o uso de um programa comercial de elementos finitos tem que ser acoplado a um método de análise com uma eficiente estratégia de otimização, figura 4.2. Muitas estratégias de otimização têm, resumidamente, a seguinte estrutura: define-se um conjunto de variáveis de projeto em seguida determina-se uma direção de descida da função objetivo. Nesta direção é executada uma série de iterações até obter-se o mínimo desejado para a função objetivo.

A integração de algoritmos de otimização com programas de elementos finitos pode seguir três linhas de abordagem. Primeiramente, pode-se ter a íntima integração com o programa de elementos finitos, quando do uso de programas comerciais que já possuem algoritmos de otimização integrados. Ou ainda pelo uso de rotinas de otimização desenvolvidas como macros dentro do programa comercial, se este permitir. A outra maneira de promover a integração é através de uma conexão externa com o programa comercial, isto é, o algoritmo de otimização invoca o programa de elementos finitos, o qual resolve o problema com as variáveis de projeto corrente e a rotina de otimização utiliza as respostas para proceder às análises necessárias.

Sendo assim, no ambiente industrial onde um programa comercial de elementos finitos tem sido usado por anos, poderia preferir uma segunda opção, que consiste na interface de um

programa de elementos finitos com um algoritmo de otimização num ambiente integrado e com grande flexibilidade. Nesta abordagem, parametrização, e algoritmo de otimização são externos ao programa comercial, onde operando com rotinas específicas para manipulação de arquivos, pode-se integrar programas comerciais de análise por elementos finitos com rotinas de otimização.

4.2 Desenvolvimento Teórico

Para o desenvolvimento de um procedimento são abordados alguns aspectos teóricos necessários para o entendimento de um sistema de otimização. Fundamentalmente, um sistema de otimização consiste de quatro módulos que podem ser representados pelo esquema indicado na figura 4.3.

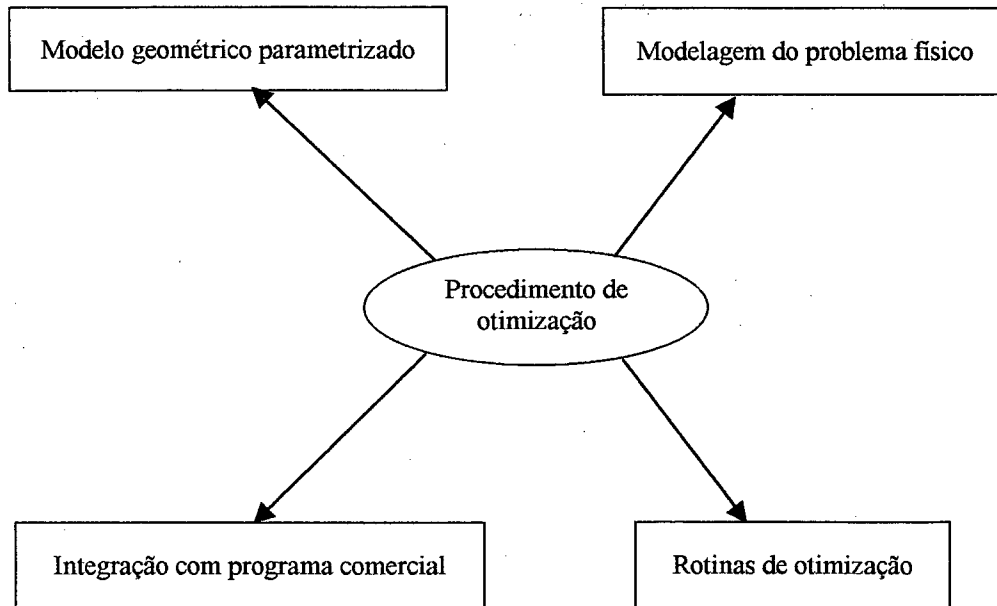


Figura 4.3 – Módulos para um procedimento de otimização.

Deseja-se uma faixa de aplicabilidade abrangente de modo que se deve desenvolver um sistema que integre todos estes quatro módulos concatenando ou apagando partes de arquivos, de modo a gerar os dados de entrada para o sistema, tornando o procedimento genérico. Vários

módulos e algoritmos foram implementados de modo a gerar o ferramental básico necessário para cumprir os requisitos do trabalho:

4.2.1 Modelo Geométrico Parametrizado

A parametrização de geometrias é a característica mais avançada introduzida em programas de CAD da atualidade. A grande flexibilidade e habilidade de aplicar dimensões precisas em rascunhos grosseiros, bem como símbolos paramétricos auxiliaram muito o dia a dia do projetista ou engenheiro na indústria, principalmente, levando-se em conta o grande número de alterações de projeto necessárias até se obter o projeto final.

A utilização de programas de CAD com rotinas de parametrização integradas a esses programas torna-se essencial no processo de um sistema de otimização, no qual existe um conhecido e definido grupo de parâmetros num projeto que deve ser manipulado e modificado dentro de limites estabelecidos. Antes de iniciar um processo de otimização de projeto, deve-se definir alguns parâmetros que são as variáveis de projeto associadas ao modelo paramétrico. As variáveis de projeto são quantidades independentes que podem ser variadas para encontrar a melhor condição de projeto, isto é, o projeto ótimo. Limites superiores e inferiores são especificados para as variáveis de projeto servindo como restrições.

A integração promovida pelo presente procedimento que está sendo proposto foi realizada a partir de um programa de elementos finitos, como o ANSYS, que possui um pré-processador que permite a entrada de dados por um arquivo dito arquivo de entrada. Este arquivo de entrada é um arquivo editável, escrito a partir de comandos do ANSYS, onde estão todas as informações relativas as características do problema que se deseja resolver. A vantagem de se ter um arquivo deste tipo está na flexibilidade que se tem em manipular as várias partes do arquivo de modo a definir as variáveis de projeto, e as saídas que se deseja. Portanto, aproveitando o conhecimento do programa de elementos finitos, o engenheiro ou o projetista, pode gerar os arquivos que representam o problema a ser resolvido, e com as rotinas integradas, proceder à otimização. Uma outra possibilidade de se obter o arquivo de entrada para o programa de elementos finitos é através da utilização de um programa de CAD, para gerar a geometria a ser otimizada. Para ilustrar, é interessante comentar um pouco mais sobre a tecnologia CAD. Existe hoje no mercado uma infinidade de programas de CAD, cada qual com suas características. Por

exemplo, o Pro-Engineer (PRO/E) que é um modelador sólido, paramétrico, podendo gerar o projeto de peças e montagens, agregando características construtivas ao sólido, podendo também criar relações paramétricas entre duas ou mais dimensões. Pode-se ainda destacar muitos outros, como EUCLID-IS, PATRAN, IDEAS, CATIA, os dois últimos bastante difundidos na indústria automobilística. Todos esses programas têm uma boa performance e possuem muitos recursos, alguns dos quais incluem a parte de elementos finitos, cada qual com suas particularidades e suas potencialidades.

4.2.2 Modelagem do Problema Físico

Quando se necessita executar uma análise numérica de um problema físico, seja ele um problema de estrutura, térmico ou de fluidos, utiliza-se uma formulação matemática que traduza o fenômeno físico. O método de elementos finitos é um método numérico bastante utilizado para resolver os problemas de engenharia. Para proceder à análise computacional pelo método de elementos finitos é importante estabelecer algumas condições que permitam esta tarefa ser executada de forma eficaz. Primeiramente, é importante que se defina qual o tipo de problema que está sendo formulado, se é um problema estático ou dinâmico, problemas que envolvem plasticidade, problemas térmicos em regime permanente ou transiente, e muitos outros problemas de engenharia. Um outro fator muito importante na definição do problema são as condições de contorno do sistema, pois são estas condições que definem o problema físico, e a sua definição é primordial para que a resposta tenha um bom grau de confiabilidade. A figura 4.4 sintetiza as fases para a modelagem de um problema de engenharia. Para proceder a solução ou resolução das equações matemáticas que descrevem o problema de engenharia utiliza-se métodos matemáticos dos quais tem grande destaque o método de elementos finitos. O método de elementos finitos é uma técnica numérica para obtenção da solução aproximada de um problema físico, portanto, a discretização do modelo é fundamental, como também a escolha do tipo de elemento, isto é, a formulação do elemento que será utilizado para discretizar o modelo e o tipo de problema que se está abordando. Um exemplo clássico é o problema que envolve casca, cuja formulação do elemento é fundamental para se ter uma descrição precisa do perfil da resposta de tensão ao longo da espessura, haja visto as várias teorias e formulações estudadas. A discussão anterior é muito interessante para se ter idéia da complexidade que envolve a solução de um problema de engenharia, que na prática pode muitas vezes parecer trivial, mas a modelagem requer o conhecimento teórico apurado para não ser cometido erros que podem levar a resultados

muitas vezes catastróficos. Detalhes a respeito deste assunto podem ser encontrados na referência [44], onde são abordadas as fases da modelagem do sistema estrutural, os fatores que influenciam a análise numérica e os tipos de erros embutidos nos resultados. A evolução da tecnologia computacional levou a implementação dos vários métodos, o que levou ao surgimento de inúmeros programas comerciais.

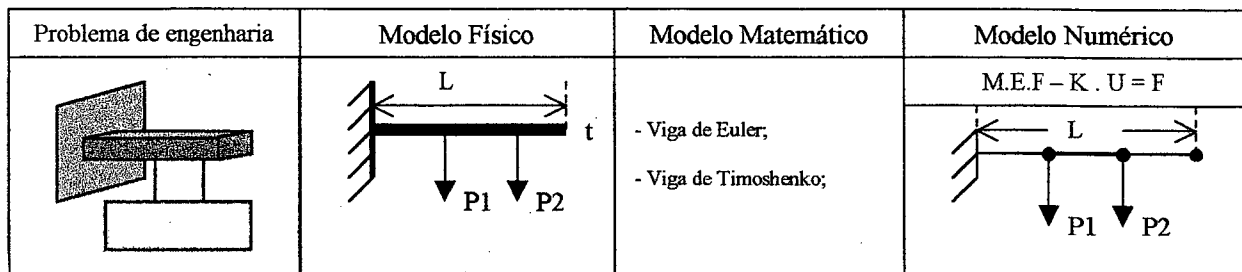


Figura 4.4 – Fases da modelagem de um problema de engenharia.

Quanto aos programas comerciais de elementos finitos, destaca-se, o IDEAS que também é um programa de CAD, tendo todas capacidades, incluindo análise de sensibilidade; o PATRAN que como modelador de elementos finitos tem grande destaque. O MSC/NASTRAN largamente usado, possui módulos para análise de sensibilidade e otimização como opcionais avançados do programa. O programa ANSYS que também tem grande destaque no contexto mundial possui uma ampla gama de aplicações, como por exemplo, análise eletromagnética, problemas de fluidos, problemas de dinâmica linear e não linear, estática linear e não linear, problemas de contato, térmicos permanentes e transientes entre outros. Existem também alguns aplicativos como ANSYS/PROFEA, que é integrado ao PRO/E, para análise linear e otimização, e o MECHANICA, também incorporado ao PRO/E. Existem muitos outros programas comerciais para análise de engenharia que são aplicáveis ao estudo de projetos otimizados, mas apenas como ilustração estão sendo citados nesta dissertação os mais conhecidos. É notado que existe uma forte tendência de se ter maior integração entre as tecnologias CAD e CAE haja visto a grande flexibilidade em se ter tudo sendo executado em um único ambiente, facilitando um reprojeto. Mesmo com o uso de um programa comercial, ainda continua sendo necessária a abordagem estabelecida inicialmente, isto é, existe a necessidade do usuário saber modelar o problema de engenharia, de modo que os programas gerem uma resposta confiável mediante a interpretação do engenheiro ou técnico.

É interessante ser comentado que a idéia do procedimento de otimização não se restringe a um programa que utiliza o método de elementos finitos, ou seja, pode ser utilizado outro método, tais como, elementos de contorno ou diferenças finitas.

4.2.3 Rotinas de Otimização

Outra parte fundamental de um procedimento de otimização é a escolha do algoritmo para solução do problema de otimização. Outra vez o fator importante para o problema é a sua formulação. Por exemplo, se é desejado minimizar a massa de uma estrutura, sujeito a algumas restrições de projeto, isso requer um aporte matemático para formular corretamente o problema. Pode-se notar na prática, que muitos dos erros encontrados na solução de problemas de otimização está nesta etapa, e isto só é amenizado com o entendimento do problema e experiência adquirida pelo engenheiro a cada novo problema a ser resolvido.

Existe uma série de métodos matemáticos aplicáveis ao desenvolvimento de algoritmos de otimização, existem também vários algoritmos já desenvolvidos com seus códigos fonte disponíveis na rede da “internet” e em bibliografia específica. Neste trabalho foi feita a implementação do algoritmo baseado no Método do Lagrangeano Aumentado (M.L.A.). Esta escolha deve-se ao fato do algoritmo ser bastante eficiente e poder ser programado de forma a ter convergência global, isto é, para qualquer valor inicial das variáveis de projeto, determina-se uma solução que satisfaz a condição necessária de ótimo do problema.

A determinação dos gradientes é a ferramenta imprescindível para um problema de otimização. Existem vários métodos aplicáveis para determinação, tais como o método de diferenças finitas, semi-analítico, e o método analítico. Os dois últimos requerem o conhecimento dos detalhes do programa de elementos finitos. No método analítico pode ser destacado: o método direto, e o método adjunto. O método semi-analítico foi desenvolvido de modo a melhorar a ineficiência do método de diferenças finitas, com relação ao custo computacional - tempo, e a dificuldade de implementação do método analítico. O método de diferenças finitas é um método de fácil implementação e bastante genérico, porém alguns autores o consideram ineficiente devido ao tempo, normalmente limitado pelo número de variáveis. Neste trabalho não é feita a análise de sensibilidade na sua forma clássica, como é verificado com os métodos analíticos, na realidade, é usado diretamente o método de diferenças finitas para

efetuar numericamente o cálculo dos gradientes da função objetivo e restrições com base nas respostas geradas pelo programa comercial de elementos finitos, o ANSYS. Desta forma tem-se a possibilidade de utilizar um programa comercial, no qual não se tem acesso aos detalhes necessários para a determinação da sensibilidade analítica. Mahmoud & Holzleitner [1994] com o método semi-analítico, descrevem uma técnica para implementar a análise de sensibilidade sem conhecer os detalhes do programa comercial de elementos finitos.

4.2.4 Integração com Programa Comercial

Quando se propõe um procedimento de otimização que integre um programa comercial de elementos finitos externo, é importante definir uma condição de gerenciamento dos dados e chamadas do programa de análise de modo automático. Estas rotinas estão embutidas dentro do algoritmo de otimização, e são responsáveis pela determinação da função objetivo e restrições, pela determinação dos gradientes numéricos, e pela manipulação de arquivos de entrada, manutenção do arquivo de variáveis de projeto, e leitura dos dados gerados pelo programa comercial. É importante deixar claro que a abordagem deste trabalho pode permitir a utilização de um outro programa comercial além do ANSYS, pois não está se manipulando com detalhes do programa fonte, e sim com comandos usuais de modelagem que é característica de cada programa comercial e que são de conhecimento do usuário. Portanto, a implementação de outros programas no sistema integrado de otimização pode ser bastante simples desde que o usuário formule o problema de otimização e conheça os procedimentos do programa comercial para acesso e geração de arquivos de entrada e saída do mesmo de forma automática.

4.3 Abordagem Genérica do Procedimento

Com base nos conceitos dos módulos que compõem o procedimento de otimização dado anteriormente, é colocado agora uma descrição que permite uma visão geral de como o procedimento foi implementado para resolver problemas de otimização pela integração de um algoritmo de otimização e um programa comercial de elementos finitos, que é executado fora das rotinas de otimização. Este procedimento é flexível, e de fácil manutenção, e baixo custo.

Baseado então no método de diferenças finitas para o cálculo dos gradientes das funções, esta abordagem permite o desenvolvimento de um procedimento de otimização de projeto com o uso de programas comerciais, dos quais não se conhece detalhes do programa. Este procedimento pode minimizar o peso, custo entre outras funções objetivo estando a estrutura sujeita a restrições de tensões, deslocamentos e restrições geométricas de suas variáveis de projeto. É considerado neste trabalho tanto a otimização de dimensão, quanto à otimização de forma das estruturas. Uma das vantagens do procedimento a ser adotado é a sua flexibilidade. Esta flexibilidade permite considerar várias classes de problemas não convencionais, tais como, posicionamento de suportes, determinação de carregamentos admissíveis, problemas de dinâmica, térmicos e fluidos entre outros.

O ponto chave do procedimento de otimização consiste de um programa comercial existente no ambiente de projeto, e sua integração com rotinas de otimização que chama este programa e controla o fluxo para proceder aos cálculos dos gradientes e algebrismo necessário para uma otimização. O método escolhido para o cálculo dos gradientes foi o método de diferenças finitas. A rotina de otimização necessita rodar pelo menos $(n+1)$ vezes, n -número de variáveis de projeto, o programa comercial, caso o método de diferenças finitas seja o “*forward*” e $(2n+1)$ vezes para o caso de diferença finita central. Todo este processo se dá sem a necessidade de acesso interativo com o programa comercial (ANSYS), já que isto é feito de forma automática pela sub-rotina de integração dentro da rotina de otimização.

Se há intenção que se chame módulos diferentes de modeladores geométricos, como também outros programas de elementos finitos, e otimizadores, existe a completa liberdade de poder ser incorporado rotinas que chamem estes outros módulos. No entanto, no aspecto geral o sistema de otimização de projeto permanece inalterado. Com a ajuda deste procedimento, o engenheiro pode executar uma análise de otimização de projeto contando com a sua experiência e o seu conhecimento prévio em um programa comercial de elementos finitos, o qual é chamado de fora da estrutura principal das rotinas de otimização, sem conhecimento de detalhes de programação do programa comercial que se deseja integrar.

4.4 Descrição do Programa

4.4.1 Aspectos Gerais

O programa para o procedimento de otimização estrutural proposto possui uma série de sub-rotinas, entre as quais englobam basicamente as sub-rotinas de otimização, sub-rotina com comandos de manipulação de arquivos e controle para ativar o programa comercial de elementos finitos. O fluxograma genérico do procedimento indicado na figura 4.5, mostra a disposição das várias sub-rotinas. As implicações com relação à implementação das sub-rotinas foram várias, principalmente, o aspecto da integração do programa comercial. Primeiramente, com relação à escolha do programa ANSYS, que é devido a sua disponibilidade, pois é o programa utilizado pela Multibrás S.A., e também está disponível no Grante (UFSC). Em segundo, é pela sua condição de trabalho, que permite que o programa seja iniciado por uma linha de comando com os parâmetros de entrada, que indicam o modo em que o ANSYS será executado, e com o arquivo de entrada e saída.

O processo iterativo de otimização é iniciado com a definição de um vetor variável de projeto inicial, o qual é reescrito em um formato ANSYS, e em seguida com o arquivo de entrada é ativado um processo ANSYS em modo “batch”. A integração com o programa de elementos finitos é feita dentro das rotinas de otimização através de sub-rotina específica que utiliza recursos do *Fortran 90*. No momento em que o programa ANSYS é invocado, a rotina de otimização recebe um “pause”, ficando temporariamente parada até que o ANSYS termine o processo e gere os arquivos de resultados e o arquivo de saída (*filename.out*), este último sendo usado como verificador do processo ANSYS. Sendo assim, a integração proporciona a manipulação de arquivos, formado por comandos ANSYS. A manipulação de arquivos promove a escrita, leitura, concatenação e eliminação, através de comandos do *Fortran 90*, que permitem usar recursos do sistema operacional. Com isto forma-se o arquivo de entrada (*input file*) para o ANSYS executar a análise por elementos finitos e gerar os arquivos de resultados a serem utilizados pelo procedimento de otimização. Este processo é repetido a cada nova iteração ou em todo momento que se deseja determinar a função objetivo e restrições, seja no processo de busca unidimensional, ou na determinação dos gradientes. Na seqüência do programa de otimização,

após o término do processo ANSYS, é determinado a função objetivo, a qual pode ser obtida pela leitura do arquivo de saída, por exemplo, o valor da massa ou através de uma função pré-definida pelo usuário que necessite ou não dos dados fornecidos pelo programa de elementos finitos. Do mesmo modo, é feito com a determinação das restrições, no entanto, os dados para obtenção destes valores são obtidos a partir dos arquivos de resultados (deslocamentos e tensões) da análise por elementos finitos. Após a obtenção da função objetivo e das restrições, o programa de otimização monta a função Lagrangeana Aumentada.

O programa foi implementado de modo a proporcionar uma lógica simples para as sub-rotinas, caracterizando a fácil manutenção do procedimento através da implementação de outros métodos matemáticos de otimização. Neste sentido, o programa pode permitir, por exemplo, incorporar outros métodos de “*line search*” (busca unidimensional), que permitem melhorar a determinação da direção de descida, que é o ponto fundamental no processo de otimização, ou ainda pode permitir o uso de outros programas comerciais de elementos finitos. A condição apresentada no programa reflete a situação mais desfavorável em relação à obtenção dos gradientes, já que é usado um programa fechado, que não permite acesso aos detalhes de programação do programa. Contudo, pode-se utilizar programas comerciais com arquiteturas abertas, o que permitiria o uso de métodos analíticos ou semi-analíticos para o cálculo dos gradientes de forma mais precisa, ao contrário do método de diferenças finitas que é mais adequado para a situação de programas fechados. Foram implementadas duas condições para o cálculo dos gradientes: o método de diferença finita central e “*forward*”, entretanto após alguns testes prevaleceu o uso do método de diferença central por sua melhor precisão, apesar do tempo maior para o processo.

A arquitetura do programa ANSYS também permite ao usuário criar rotinas que possibilitam ter uma condição customizada adequada ao ambiente de projeto do usuário. Esta condição possibilita ao usuário o desenvolvimento de suas próprias rotinas de otimização dentro do programa de elementos finitos, criando um ambiente customizado e específico para o seu tipo de problema. O processo de customização do ANSYS requer um profundo conhecimento e experiência com o programa de elementos finitos, pois se trata de um processo muito delicado o que pode levar o usuário menos experiente a cometer erros. Além disso, o usuário estaria limitado somente aos recursos oferecidos pelo ANSYS. Por isso, é interessante se ter outros meios que permitam ao engenheiro diversificar as técnicas aplicadas para solução de problemas

de otimização. Uma alternativa está sendo proposta neste trabalho, onde o programa ANSYS é ativado através de uma sub-rotina específica do programa de otimização.

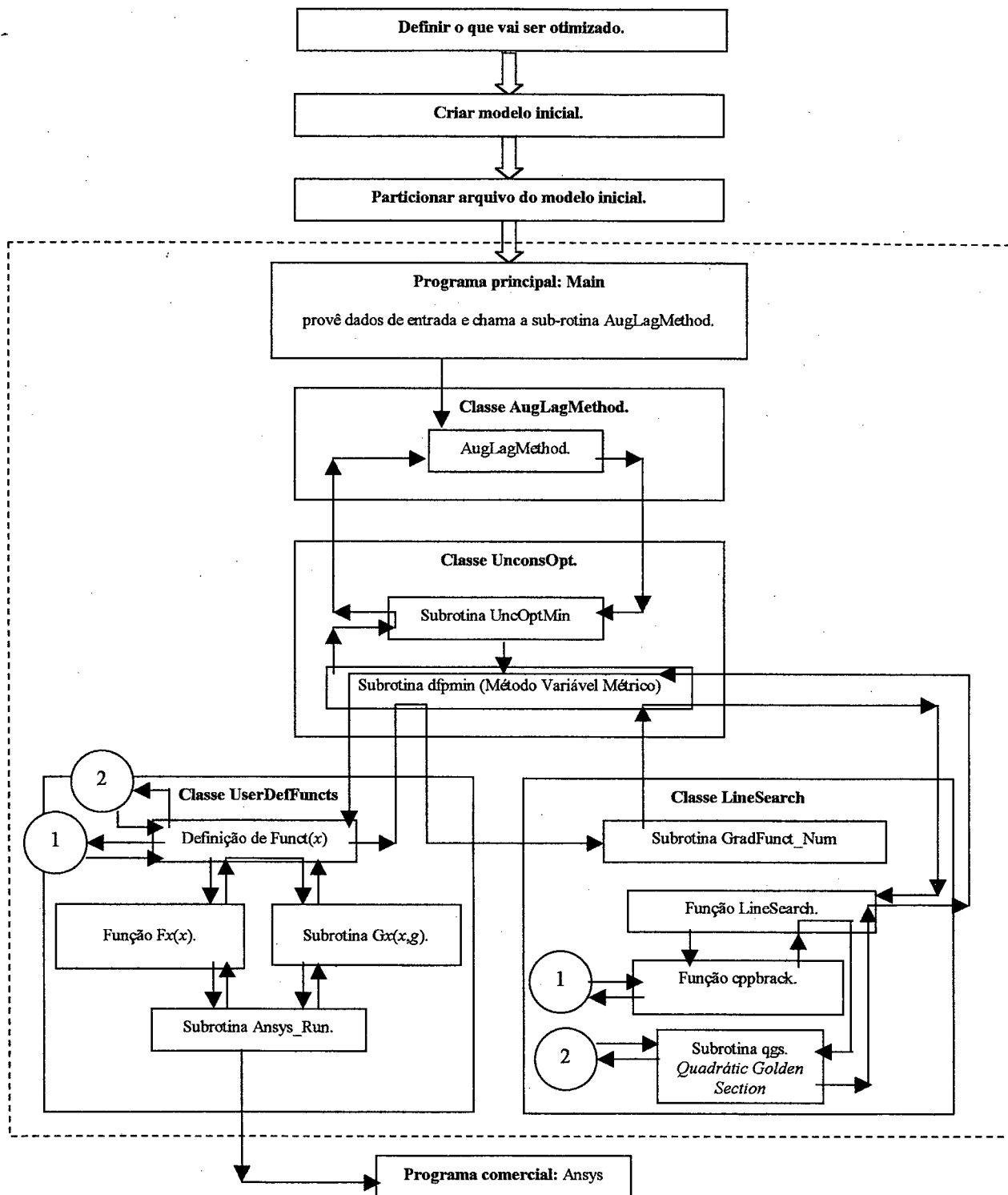


Figura 4.5 – Fluxograma genérico do processo de otimização proposto.

Existe uma etapa do procedimento que não faz parte do corpo do programa que é a preparação dos arquivos de entrada para o programa ANSYS, bem como a sua partição de acordo com a formulação do problema que se deseja resolver. Esta etapa também poderia ser implementada de modo a fornecer subsídios ao usuário menos experiente a formatar os dados de entrada numa linguagem entendida pelo ANSYS. Mas para usuários mais experientes isto pode ser tedioso, uma vez que conhecendo os comandos do ANSYS basta escrever o arquivo diretamente, e particioná-lo da maneira que melhor atenda os objetivos de problema proposto, permitindo ao usuário maior controle e segurança na manipulação de dados.

4.4.2 Preparação do Arquivo de Entrada

A entrada de dados para a análise estrutural é realizada por meio de um arquivo chamado arquivo de entrada (*input file*), que possui comandos em formato ANSYS, e são produzidos através de um editor de texto. Inicialmente, é descrita de forma genérica e rápida a definição do objetivo e como buscar informações que permitam formular o problema da melhor maneira possível em um ambiente de projeto na indústria. Em seguida, é descrito o arquivo de entrada para o programa ANSYS, que descreve o modelo a ser analisado e a partição deste arquivo, que permitirá a manutenção do arquivo responsável pela definição das variáveis de projeto.

i) Definir o que será otimizado.

Esta fase é um tanto subjetiva e requer a experiência do engenheiro ou do técnico envolvido no projeto. Na realidade, é nesta fase onde o engenheiro irá formular matematicamente o problema de otimização. Para fazer isto em um ambiente industrial, o engenheiro deve cercar-se de todas as informações relevantes ao projeto. Um exemplo bastante prático, é o processo de engenharia simultânea, onde o grupo ou equipe de projeto é formado por pessoas de diferentes áreas e especialidades, que durante a fase conceitual e de desenvolvimento, fazem freqüentes reuniões para discutir a cerca de cada componente a ser projetado. Como exemplos, pode-se citar na fase de desenvolvimento: as reuniões de análise de manufaturabilidade (RAM), onde são discutidos os aspectos relativos a condição de fabricação do componente e montagem em algum conjunto, as reuniões de análise crítica, para discutir aspectos intrínsecos do componente, bem como assuntos que envolvem os aspectos funcionais do componente no conjunto. Na fase conceitual: as reuniões nas quais é elaborada a matriz de

QFD¹, para traduzir os anseios do consumidor na forma de função que componente deve desempenhar e as reuniões onde são elaborados os FMEA's² (projeto e processo), os quais permitirão através da função do componente detectar as possíveis falhas, medir os riscos e seus reflexos no componente, no conjunto e principalmente, no consumidor final. Nesta etapa, o engenheiro de projeto adquire a sensibilidade para definir a função objetivo e as variáveis factíveis de serem otimizadas, como também as restrições a serem impostas ao projeto, levando em consideração a experiência de toda equipe de projeto.

ii) Criação do Modelo Inicial – Arquivo de Entrada para o ANSYS

Um dos passos para o procedimento de otimização consiste na criação de um arquivo que descreve o modelo de elementos finitos com as características do projeto inicial. O arquivo é escrito com comandos específicos do programa ANSYS, descrevendo a geometria com todas as características necessárias para proceder à análise. Os comandos do ANSYS permitem atribuir propriedades geométricas, propriedades materiais, atribuir parâmetros geométricos, tais como coordenadas que definem posição de K-pontos, e definição de linhas, áreas, volumes, definindo portanto a geometria a ser analisada. Neste arquivo são também definidos os parâmetros para geração de malha e definição das condições de contorno. Todas as definições fazem parte das etapas de pré-processamento (/PREP7), solução (SOLUTION) e pós-processamento (/POST1) do programa ANSYS. A seguir é apresentado parte do arquivo de entrada do ANSYS com os comandos relativos a definição do modelo e outros comandos necessários para auxiliar na construção e controle do modelo. O arquivo apresentado refere-se ao utilizado no exemplo 4 - caso prático: otimização de espessura de prateleira de refrigerador. Para melhor entendimento, a forma de apresentação do arquivo de entrada foi separada nas etapas de pré-processamento, solução e pós-processamento.

Pré-processamento - nesta etapa o arquivo mostra detalhes de comandos do programa ANSYS, os quais indicam que o programa irá rodar em "batch". É observado também a indicação da fase de pré-processamento /PREP7, e a definição do tipo de elemento a ser usado. O arquivo mostra as características do modelo geométrico e de elementos finitos, indicando as propriedades materiais, definidas pelo comando UIMP.

¹QFD – Quality Function Deployment;
²FMEA – Failure Mode Effect Analysis.

Também está indicado a definição da geometria através da definição de k-pontos, linhas e áreas, além de outros comandos relativos a manipulação, como seleção, cópia, e eliminação de entidades geométricas. Outros comandos relativos a controle e geração de malha estão presentes (ESHAPE, LESIZE, AMESH), e colocação das condições de contorno (D,ALL,UX,0,,,,UY,UZ e SFA,ALL,1,PRES, 0.0009696969). A seguir é apresentado uma parte do arquivo de entrada relativo a fase de pré-processamento e para os principais comandos é apresentado um comentário ao lado descrevendo o que este comando faz.

```

BATCH
/COM,ANSYS RELEASE 5.3 UP071096 12:09:56 11/05/1996
!
/AUX15
IOPTN,IGES,0
IOPTN,MERG,0
IOPTN,SOLID,0
IOPTN,GTOLER,DEFA
IGESIN,PRAT1,IGS,c:\estudo~1\pratel~1\
!*
!* Design parameters

!* geometry properties
t1=0.265898508347D+01
!* geometry properties
t2=0.200001491881D+01

/PREP7
/pnum,kpoint,1
NUMMRG,ALL,,01,,01,
NUMCMP,ALL
ET,1,SHELL93
!*
!* Design parameters
!* geometry properties
R,1,t1
!* geometry properties
R,2,t2
!*
ASEL,s,area,,9
CM,fundo,AREA
AATT,1,1,1,0
ASEL,invert,,,
AATT,1,2,1,0
ALLSEL,ALL
!*
UIMP,1,EX,,3500,
UIMP,1,NUXY,,0.35,
UIMP,1,DENS,,1,
!*
LESIZE,ALL,20,,1,
ESHAPE,1,0
AMESH,ALL
!*
SFA,9,1,PRES,-0.0007721510
/PBC,ALL,,1
LSEL,S,line,,17,,,1
!*
D,ALL,UY,0,,,,UZ
ALLSEL,ALL
NSEL,S,LOC,X,-1,1
!*
DSYM,SYMM,X,
!*
D,ALL,UX,0,,,,
FINISH
    
```

- * Indica o modo que o programa vai ser executado.
- * Indica a versão do programa utilizada.
- * Início do módulo de leitura do arquivo IGES. com comandos para calibrar a leitura
- * Commando utilizado para ler arquivo IGES.
- * Variável de projeto do grupo geométrico 1.
- * Variável de projeto do grupo geométrico 2.
- * Início da fase de pré-processamento.
- * Comando para acertar a conectividade geométrica.
- * Reordenação da numeração das entidades.
- * Definição do tipo de elemento – triangular de 6 nós.
- * Definição da propriedade geométrica de espessura para grupo 1.
- * Definição da propriedade geométrica de espessura para grupo 2.
- * Associa os grupos de propriedades materiais e geométricas a área.
- * Definição de propriedade material isotrópica – módulo de elasticidade.
- * Coeficiente de poisson.
- * Densidade.
- * Definição da dimensão para discretização.
- * Forma do elemento – triangular.
- * Executa malha de todas as areas.
- * Condição de carregamento de pressão sobre área número 9.
- * Colocação de condição de deslocamento zero na direção UY e UZ.
- * Colocação de condição de simetria no plano normal ao eixo X.
- * Colocação de condição de deslocamento zero na direção UX.

Fase de solução - uma outra fase que consta no arquivo de entrada é a fase de solução a qual é identificada pelo comando /SOLU. Nesta fase é definido o tipo de análise e são calibrados alguns aspectos que dependem do tipo de análise, por exemplo, aspectos de não linearidade de material, não linearidade geométrica, entre outras. O uso do comando /SOLVE, faz com que o ANSYS inicie a análise no modo definido pelo usuário “*interactive*” ou “*batch*”. Os comandos referentes a esta fase estão indicados abaixo na porção do arquivo de entrada referente a esta fase:

/SOLU	* Início da fase de solução.
/STATUS,SOLU	
ANTYPE,STATIC	* Identifica que o tipo de análise é estática.
NLGEOM,ON	* Identifica que o problema possui não linearidade geométrica.
SOLVE	* comando para iniciar a resolução do problema.
FINISH	

Pós-processamento - esta é a última fase no processo de análise numérica por elementos finitos e é iniciada a partir do comando /POST1. Nesta fase são gerados os arquivos dos resultados de tensão e de deslocamento através do comando /OUTPUT, conforme indicado abaixo na porção do arquivo de entrada. Outros comandos também são usados para auxiliar a formatação dos resultados e seleção dos nós ou elementos. Parte do arquivo de entrada é apresentado abaixo representando a fase de pós-processamento:

/POST1	* Início da fase de pós-processamento.
SET, LAST	* Comando para compilar os resultados.
!* ALLSEL, ALL	* Seleção de todas as entidades geométricas e elementos finitos.
NSORT, S, EQV, ,	
/OUTPUT, SIGMA, TXT, C:\ESTUDO_06_05_00\prateleira,	* Geração dos dados de tensão para um arquivo (SIGMA.TXT).
/FORMAT, , F, 18, 12, ,	* Formatação dos resultados.
PRNSOL, S, PRIN	* Comando para gerar lista de tensões principais.
/OUTPUT, , , ,	
NSORT, U, Y, , 1	
/OUTPUT, SIGMA, TXT, C:\ESTUDO_06_05_00\prateleira, APPEND	* Geração dos dados de deslocamentos para arquivo (SIGMA.TXT)
/FORMAT, , F, 18, 12, ,	
PRNSOL, U, SUM,	* Comando para gerar lista de deslocamentos.
!* FINISH	

Uma outra maneira de se obter o arquivo de entrada é utilizando um pré-processador, que permita gerar o arquivo para o ANSYS. Neste caso, pode-se destacar o programa Pro-Engineer, que possui um aplicativo para geração de um modelo de elementos finitos. Outra forma é a partir da criação de um arquivo no padrão IGES obtido por um programa de CAD (Pro-Engineer), como foi apresentado no exemplo acima. O arquivo IGES é então incorporado ao corpo principal do arquivo de entrada por comandos específicos do ANSYS, e posteriormente à leitura do arquivo IGES são usados comandos ANSYS para fazer alguns ajustes, e complementação de modo a obter-se o arquivo de entrada.

O arquivo descrito acima representa o modelo geométrico e o modelo de elementos finitos mostrados na figura 5.18 e 5.19, respectivamente.

iii) Partição do Arquivo de Entrada

Considerando as três fases, pré-processamento, solução e pós-processamento, elaborase a partição do arquivo de entrada, de modo que se tenha em cada fase os arquivos com a discriminação dos comandos ANSYS. Por exemplo, pode-se ter um arquivo de propriedades de materiais, um arquivo com as entidades geométricas, um arquivo das variáveis de projeto, sendo o mais importante na estrutura principal da fase de pré-processamento. A fase de solução contém somente os comandos que definem o tipo de análise e para rodar o ANSYS. Na fase de pós-processamento, pode-se dividir os comandos em arquivos, de modo que se pode ter um arquivo de comandos ANSYS que geram a saída de tensões, e um outro arquivo para gerar os deslocamentos nodais da estrutura, ou mesmo de algum nó específico. A separação do arquivo de entrada se faz necessário, pois no caso do arquivo que contém as variáveis de projeto, este sofrerá uma manipulação constante, com várias mutações provocadas pelo algoritmo de otimização. A manutenção deste arquivo ocorre automaticamente dentro da rotina de otimização, através de comandos da linguagem de programação (*Fortran 90*), que escrevem o arquivo de variáveis de projeto com os comandos específicos do programa ANSYS. Um exemplo do arquivo de variáveis de projeto é dado a seguir:

```
!* Design parameters  
!* geometry properties
```

```
R,1,0.20000000D+01  
R,2,0.40000000D+01
```

Neste exemplo, tem-se as definições das propriedades geométricas para um elemento de casca que é a espessura, onde R define a constante do tipo real, 1 define o grupo geométrico, 2.0 define o valor da espessura, conforme unidade na qual o usuário está trabalhando, do mesmo modo para a linha seguinte. Para o caso de elementos de viga, pode-se ter, por exemplo, o momento de inércia, altura da viga, espessura e área, como propriedades a serem definidas.

4.4.3 Descrição das Sub-rotinas

O procedimento para otimização de projeto proposto foi apresentado na figura 4.5, que mostra de maneira genérica a disposição das sub-rotinas para o procedimento de otimização integrando um programa comercial, o ANSYS. Além das rotinas que foram desenvolvidas, outras rotinas de otimização que integram o procedimento podem ser encontradas na rede da “internet” ou em bibliografia específica, referências [41], [42], as quais após revisões e adaptações, puderam ser usadas no programa. Nesta seção serão descritos com mais detalhes algumas das rotinas mais importantes no processo de otimização proposto.

1- Rotina Principal: **Main**

Nesta rotina o usuário irá inicializar alguns parâmetros de controle do processo de otimização, tais como: número de variáveis de projeto; número de inequações; parâmetro de penalização e o limite inferior (r_{\min}); multiplicador de Lagrange; tolerância para convergência; e inicialização das variáveis de projeto. Após as inicializações, é chamado a sub-rotina **AugLagMethod**. A figura 4.6 mostra o fluxograma da rotina **Main**.

2- Sub-rotina **AugLagMethod**

Esta sub-rotina, que está ilustrada na figura 4.7, tem a responsabilidade de iniciar o processo iterativo para minimização de um problema com restrições do tipo

$$\min f(x) \tag{4.1}$$

sujeito à

$$g_j(x) \geq 0; \quad j = 1, \dots, m \tag{4.2}$$

Esta sub-rotina através de um “loop” chama a sub-rotina **dfpmin**, para otimização sem restrição, de uma função Lagrangeana Aumentada para valores do multiplicador de Lagrange (v_g) e o parâmetro de penalidade (r). Após o retorno a esta sub-rotina é chamado a sub-rotina **Gx(x,g)**, determinando as restrições e em seguida é executado a atualização do multiplicador de Lagrange, determinando o novo valor (v_{gn}). O erro para o multiplicador de Lagrange é determinado por:

$$\text{erro} = \text{maxval} \left(\frac{\text{dabs}(\text{vgn} - \text{vg})}{(1 + \text{dabs}(\text{vgn}))} \right) \quad (4.3)$$

e a atualização do parâmetro de penalidade (r) é dado por:

$$r = \frac{1}{2}r \quad (4.4)$$

e para verificar se $r \geq r_{\min}$, tem-se

$$r = \text{dmax1}(r, r_{\min}) \quad (4.5)$$

para evitar o mal condicionamento numérico. O “loop” desta sub-rotina continua enquanto o erro se mantém maior do que a tolerância definida para o problema.

3- Sub-rotina **dfpmin**

O objetivo principal desta sub-rotina, ilustrada na figura 4.8, é executar uma otimização sem restrição através do método variável métrico DFP (Davidon Fletcher Powell), caracterizado pelo modo de atualização da matriz Hessiana. A sub-rotina inicia com a determinação da função no ponto x_0 , com a função **funct(x)** e em seguida é determinado, numericamente, o gradiente através da função **Gradfunct_Num(x)**. É nesta sub-rotina onde se inicia o processo de otimização propriamente dito, ela é o ponto de partida para o processo de otimização. Daqui parte a chamada para outras sub-rotinas e funções que permitirão determinar o valor ótimo do problema, tais como determinação da direção de busca e o passo α . Isto caracteriza o processo de “*line search*”. Primeiramente, a matriz Hessiana é igualada a matriz identidade determinando a direção de busca unitária. Através desta sub-rotina inicia-se um “loop” que é o processo de “*line search*” com a chamada da função **LineSearch** determinando o valor de α e em seguida volta para a sub-rotina **dfpmin**, determina-se o valor do novo ponto **pnew**, conforme equação (3.8), e a função no novo ponto é determinada por **funct(pnew)**. Antes de proceder ao cálculo do gradiente faz-se a verificação para o intervalo Δx com a tolerância adotada, tal que,

$$\text{maxval} \left(\frac{\text{dabs}(\Delta x)}{\max(\text{dabs}(x), 1.0)} \right) < \text{tolerância}_x \quad (4.6)$$

O cálculo do gradiente é feito para o novo ponto e então é averiguado o critério de convergência para o problema, então,

$$\maxval\left(\frac{dabs(g) \cdot \max(dabs(x), 1.0)}{den}\right) < \text{tolerância} \quad (4.7)$$

onde

$$den = \max(f(p_{new}), 1.0), \quad (4.8)$$

e a seguir procede-se os algebrismos para atualização da matriz Hessiana. O método utilizado para atualizar a Hessiana é dado pela equação (2.51). Muitos autores consideram esta abordagem como a melhor condição para atualização da inversa da matriz Hessiana. Com a determinação da nova matriz Hessiana pode-se determinar a nova direção de descida, equação (2.50), e se verifica se é a direção de descida, caso contrário utiliza-se novamente a identidade como feito inicialmente. Assim, iterativamente o processo se repete até ser atingido a convergência ou a máxima iteração for alcançada.

4- Função **Func(x)**

Esta sub-rotina é onde se determina o valor da função Lagrangeana Aumentada (**func**), Isto é feito através do valor que retorna da função **Fx(x)** e da determinação do vetor de restrições determinado pela sub-rotina **Gx(x,g)**. O fluxograma é mostrado na figura 4.9.

5- Função **Fx(x)**

Esta função, que está ilustrada no fluxograma da figura 4.10, é responsável pela determinação da função objetivo f . Esta função chama a sub-rotina **ANSYS_Run**, que executará o programa comercial. Outro aspecto é a geração em linguagem ANSYS do arquivo das variáveis de projeto, que é feito de forma automática com a linguagem *Fortran 90*. Após o término do programa ANSYS, que gera os arquivos de resultados, que contém os deslocamentos e tensões, e um arquivo de saída, que possui a massa da estrutura ou componente, procede-se a leitura do arquivo de saída para buscar o dado referente a função objetivo. Outra forma de obter o valor da função objetivo é através de comandos do ANSYS que podem estar descritos no

arquivo de entrada que direcionam o valor da massa ou alguma outro valor da função, para um arquivo qualquer através do comando /OUTPUT. O arquivo de saída costuma ser muito grande em função disso uma porção do arquivo de saída é apresentada a seguir para deixar mais claro de onde pode se obter, por exemplo, a massa da estrutura:

```

LOAD STEP OPTIONS

LOAD STEP NUMBER. . . . . 1
TIME AT END OF THE LOAD STEP. . . . . 1.0000
NUMBER OF SUBSTEPS. . . . . 1
STEP CHANGE BOUNDARY CONDITIONS . . . . . NO
PRINT OUTPUT CONTROLS . . . . . NO PRINTOUT
DATABASE OUTPUT CONTROLS. . . . . ALL DATA WRITTEN
                                      FOR THE LAST SUBSTEP

***** CENTROID, MASS, AND MASS MOMENTS OF INERTIA *****

CALCULATIONS ASSUME ELEMENT MASS AT ELEMENT CENTROID

TOTAL MASS = .31964E+06

CENTROID                MOM. OF INERTIA          MOM. OF INERTIA
                        ABOUT ORIGIN          ABOUT CENTROID
XC = -133.78            IXX = .2116E+11         IXX = .5010E+10
YC = 18.192             IYY = .2863E+11         IYY = .6872E+10
ZC = -224.03           IZZ = .7713E+10         IZZ = .1887E+10
                        IXY = .8027E+09         IXY = .2481E+08
                        IYZ = .1357E+10         IYZ = .5417E+08
                        IZX = -.9565E+10         IZX = .1483E+08

*** MASS SUMMARY BY ELEMENT TYPE ***

TYPE    MASS
  1    319635.

```

6- Sub-rotina ANSYS_Run

A função desta sub-rotina é fazer a concatenação dos arquivos particionados e formar o arquivo de entrada do ANSYS, também chamado de “*input file*”. Então através de recursos da linguagem de programação (*Fortran 90*) se faz a chamada do ANSYS por uma linha de comando que possui alguns parâmetros que identificam que o programa ANSYS executará no modo “*batch*”, e gerará um arquivo de saída com um nome que o usuário pré definiu. Como dito anteriormente, também se pode através de comandos ANSYS dentro do arquivo de entrada ter outras opções para gerar um arquivo de onde se obtém a função objetivo. As funções restrições são obtidas a partir dos arquivos de resultados obtidos por comandos ANSYS, que geram saídas, por exemplo, deslocamentos e ou tensões. Após o programa ANSYS terminar de executar ele retorna à função $F(x)$ ou à sub-rotina $G(x,g)$, e estas para a função $Funct(x)$. A figura 4.11 mostra o fluxograma para esta sub-rotina.

7- Sub-rotina $Gx(x,g)$

Esta sub-rotina é responsável pela determinação do vetor de restrições para o problema. A sub-rotina $Gx(x,g)$, figura 4.12, também chama a sub-rotina `ANSYS_Run`, como também possui comandos do *Fortran 90* que geram o arquivo das variáveis de projeto de forma automática. Com a leitura dos arquivos de resultados (deslocamentos e tensões) e com a formulação do problema pode-se determinar o vetor das restrições. Os arquivos de resultados são gerados pelo comando `/OUTPUT`, e fica a critério do usuário dar o nome e direcioná-lo para o diretório de sua preferência, de onde serão lidos os dados para os cálculos das restrições. Para ilustrar é apresentado abaixo um arquivo de resultados do ANSYS, com os dados de tensão e deslocamentos dos nós. Primeiramente, é apresentado parte do arquivo de resultados denominado `SIGMA.TXT`, com os valores de tensão, obtido pelo comando:

```
/OUTPUT,SIGMA,TXT,C:\ESTUDO_06_05_00\prateleira,
```

```
CHARACTERS IN INTEGER FIELD   =   7
OUTPUT DISPLAY TYPE           =   F
CHARACTERS PER OUTPUT FIELD   =  18
CHARACTERS AFTER DECIMAL POINT =  12
LINES PER PAGE                =  24
CHARACTERS PER LINE           =  80
```

```
PRINT S   NODAL SOLUTION PER NODE
```

```
***** POST1 NODAL STRESS LISTING *****
```

```
LOAD STEP=      1  SUBSTEP=      1
TIME=      1.0000  LOAD CASE=      0
SHELL NODAL RESULTS ARE AT TOP
```

NODE	S1	S2	S3	SINT	SEQV
21	9.510825414517	2.776858164021	.754187473767	8.756637940750	7.940913665232
4	2.636555503363	.410257280347	-4.198889399089	6.835444902452	6.038374943013
8	6.157880264843	3.673086839373	.880472937705	5.277407327138	4.572959592407
61	2.148304259855	-.101968352988	-3.071429453471	5.219733713326	4.534702090404
24	4.307024970644	.073110079129	-.048937916590	4.355962887234	4.296239268058
531	.935170597447	-.222548152431	-3.443899616893	4.379070214340	3.930246423796
599	.990906891377	.581255273378	-2.856229294810	3.847136186187	3.659547201481
562	.000393531709	-1.642705476160	-4.158014695886	4.158408227595	3.627596574030
92	1.870453518902	.712909819489	-2.163068397939	4.033521916841	3.597252922496
539	1.285337488071	.081849685742	-2.745596943698	4.030934431769	3.584080755641
6	.888654171231	-.157295979091	-3.075189529653	3.963843700883	3.558101416093
145	3.087793817172	.103560743824	-.823821859274	3.911615676446	3.540227586527
64	1.620618831534	-1.399312254069	-2.257209063148	3.877827894683	3.527996589964
306	3.559831524759	2.113470995996	-.379917486688	3.939749011447	3.451852196860
533	.000225104832	-2.074937087689	-3.906048404365	3.906273509197	3.385132161968
1	3.022913380736	.431831073332	-.752653195704	3.775566576440	3.344518910133
688	1.395524569546	.797225860875	-2.102674248560	3.498198818107	3.240738878751
127	1.069619211243	-.198266802132	-2.606997583235	3.676616794478	3.234735690298
690	1.639895021442	.933855599563	-1.886695542507	3.526590563949	3.231937608928
537	1.887637670388	.525098882219	-1.775451084400	3.663088754788	3.206810044548
541	1.467487631249	.258191362164	-2.149459838447	3.616947469696	3.189160608120
1573	.111766098005	.069992048346	-3.066600701190	3.178366799195	3.157687022006
1593	.129202268314	.051000891701	-3.048180521799	3.177382790113	3.139012765504
1595	.087276586735	.049851704974	-3.041728901908	3.129005488642	3.110461912528

Uma outra parte do arquivo SIGMA.TXT com valores de deslocamentos, é gerado pelo comando: /OUTPUT,SIGMA, TXT,C:\ESTUDO_06_05_00\prateleira,APPEND.

```
CHARACTERS IN INTEGER FIELD    =    7
OUTPUT DISPLAY TYPE           =    F
CHARACTERS PER OUTPUT FIELD   =   18
CHARACTERS AFTER DECIMAL POINT =   12
LINES PER PAGE                =   24
CHARACTERS PER LINE          =   80
```

PRINT U NODAL SOLUTION PER NODE

***** POST1 NODAL DEGREE OF FREEDOM LISTING *****

```
LOAD STEP=      1  SUBSTEP=      1
TIME=      1.0000  LOAD CASE=    0
```

THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN GLOBAL COORDINATES

NODE	UX	UY	UZ	USUM
615	.000000000000	-6.995628194973	.014210049331	6.995642627222
898	-.000395352593	-6.983298158449	.013776884101	6.983311759376
897	-.000594246687	-6.982789288668	.014334146090	6.982804026379
616	.000000000000	-6.980627401291	.013689277987	6.980640823877
1016	-.000966214780	-6.970950110175	.013874498183	6.970963984545
614	.000000000000	-6.960113459525	.014718939925	6.960129022989
700	-.001209644517	-6.950215004098	.014415321820	6.950230058633
896	-.000638332151	-6.933113746395	.014839445331	6.933129656732
701	-.000762182614	-6.929585240261	.013323311844	6.929598090338
899	-.000379124092	-6.927106196086	.013255421935	6.927118888961
1018	-.001744787869	-6.922854994037	.013860163764	6.922869088528
617	.000000000000	-6.914984125020	.013168742662	6.914996664140
1020	-.001494518543	-6.900583716933	.013329869570	6.900596753427
1017	-.001992086713	-6.878075363237	.014519188429	6.878090976253
613	.000000000000	-6.874924834572	.015212532081	6.874941665364
1013	-.001370124651	-6.874585106078	.014957037602	6.874601513609
1112	-.002487814968	-6.851208788971	.013919548463	6.851223380764
1019	-.001016332867	-6.849364776687	.012908028366	6.849377015047
734	-.002236900154	-6.840282995266	.013262363029	6.840296217951
900	-.000368823608	-6.820516158082	.012739125423	6.820528064894
895	-.000757855260	-6.819733007151	.015370469133	6.819750370394
1042	-.001726272617	-6.818211282845	.012890332009	6.818223686430
618	.000000000000	-6.799081518381	.012654482275	6.799093294662
1015	-.002233270373	-6.776138193763	.015041866693	6.776155256945
756	-.002827060375	-6.773485077668	.014553639353	6.773501302730
1113	-.003102205348	-6.758544164742	.013407197987	6.758558174892
612	.000000000000	-6.741497983010	.015678960779	6.741516215566
699	-.001566358628	-6.739792604083	.015470637758	6.739810541860
709	-.001268207480	-6.738394582576	.012488532190	6.738406274652
1044	-.002251519078	-6.727267916557	.012594012306	6.727280081848
1176	-.003472112371	-6.713151327520	.014032786096	6.713166892073
902	-.000631109392	-6.702657969067	.012347346914	6.702669371644
1114	-.003013366953	-6.689859445594	.012711070252	6.689872200086
894	-.000829940186	-6.668302764198	.015824780440	6.668321592986
1041	-.001796197281	-6.648186266942	.012232671868	6.648197763646
1172	-.003146180468	-6.644177861684	.015077289962	6.644195713615
619	.000000000000	-6.633442546908	.012149558336	6.633453673231

8- Sub-rotina GradFunct_Num

A partir dos valores da função $Funct(x)$ calculados em x , em $x+\Delta x$ e em $x-\Delta x$, procede-se o cálculo da derivada pelo método das diferenças finitas central. A seguir, retorna para a sub-rotina $dfpmin$, de onde se chama a função $LineSearch$ para o processo de busca unidimensional, onde se obtém o valor de alfa (α). O fluxograma mostrado na figura 4.13 ilustra esta sub-rotina.

9- Função **LineSearch**

Nesta função é selecionado o método **cppbrack**, para fazer o “*bracketing*” determinando os intervalos de incerteza e o método **gqs** - “*Quadratic Golden Section*”, como o método para minimização 1-D, para determinar o valor mínimo de alfa (α). A figura 4.14 mostra o fluxograma da função.

10- Função **cppbrack** e Sub-rotina **gqs**

Como já foi dito, estas duas sub-rotinas fazem parte do processo de busca unidimensional, e isto também é realizado através da determinação das funções **Fx(x)** e sub-rotina **Gx(x,g)**. Portanto, existe uma troca freqüente de informações entre as sub-rotinas **cppbrack** e **gqs** com a função **Funct(x)**. Após ser atingido o critério de convergência na **cppbrack** o processo retorna para a função **LineSearch**, e esta vai para a sub-rotina **gqs**, que retorna o valor para **LineSearch**, e esta atribui o valor para a alfa (α) em **dfpmin**.

4.4.4 Fluxograma das Principais Sub-rotinas

Rotina Main

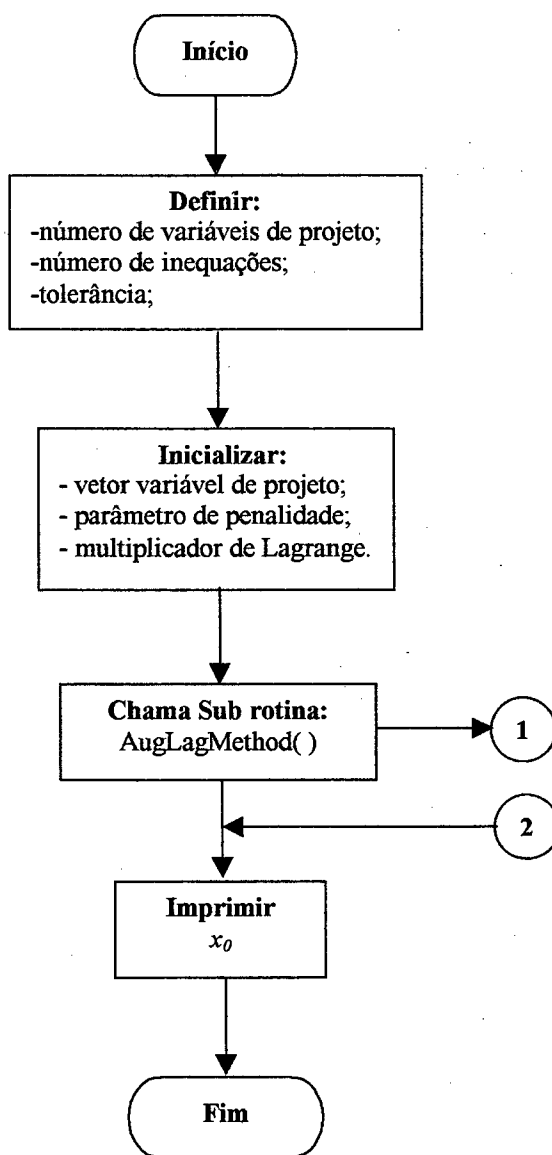


Figura 4.6 – Fluxograma da rotina Main.

Sub-rotina AugLagMethod

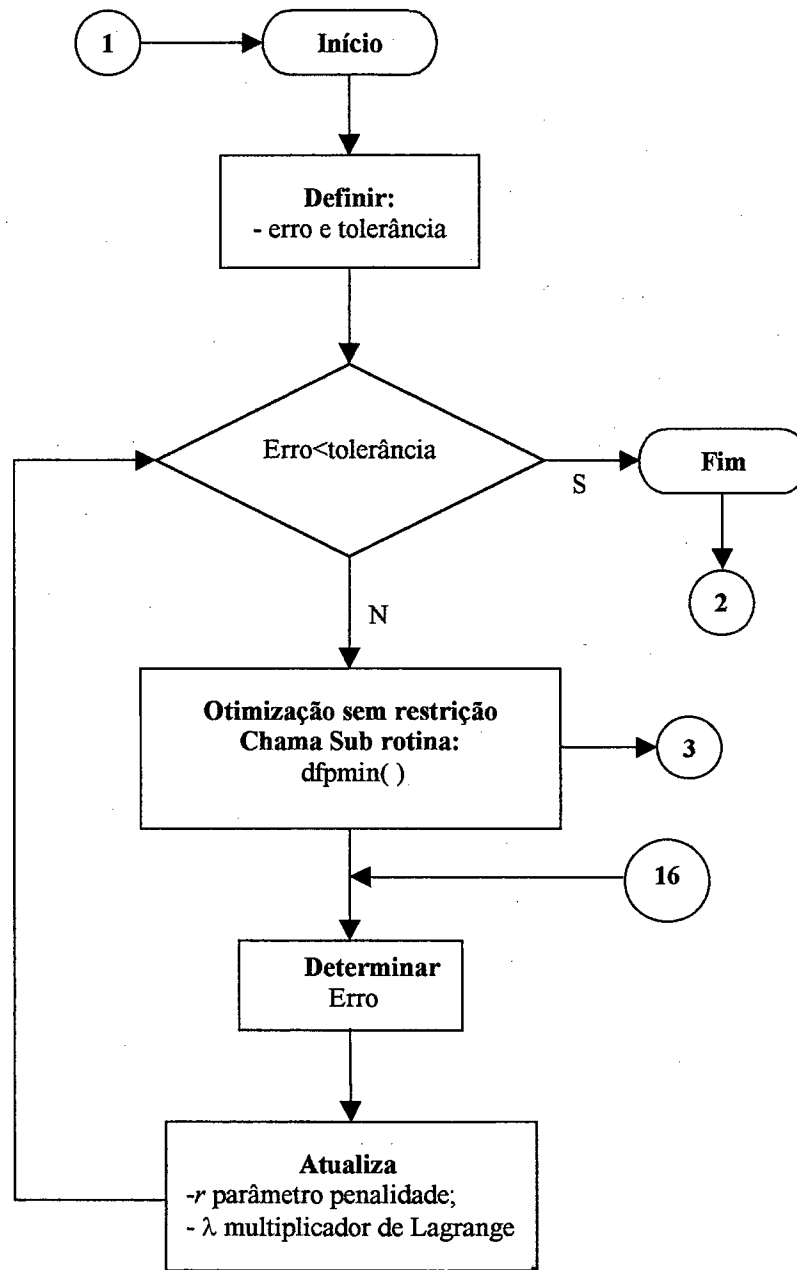


Figura 4.7 – Fluxograma da sub-rotina AugLagMethod.

Sub-rotina **dfpmin** – Método Variável Métrico

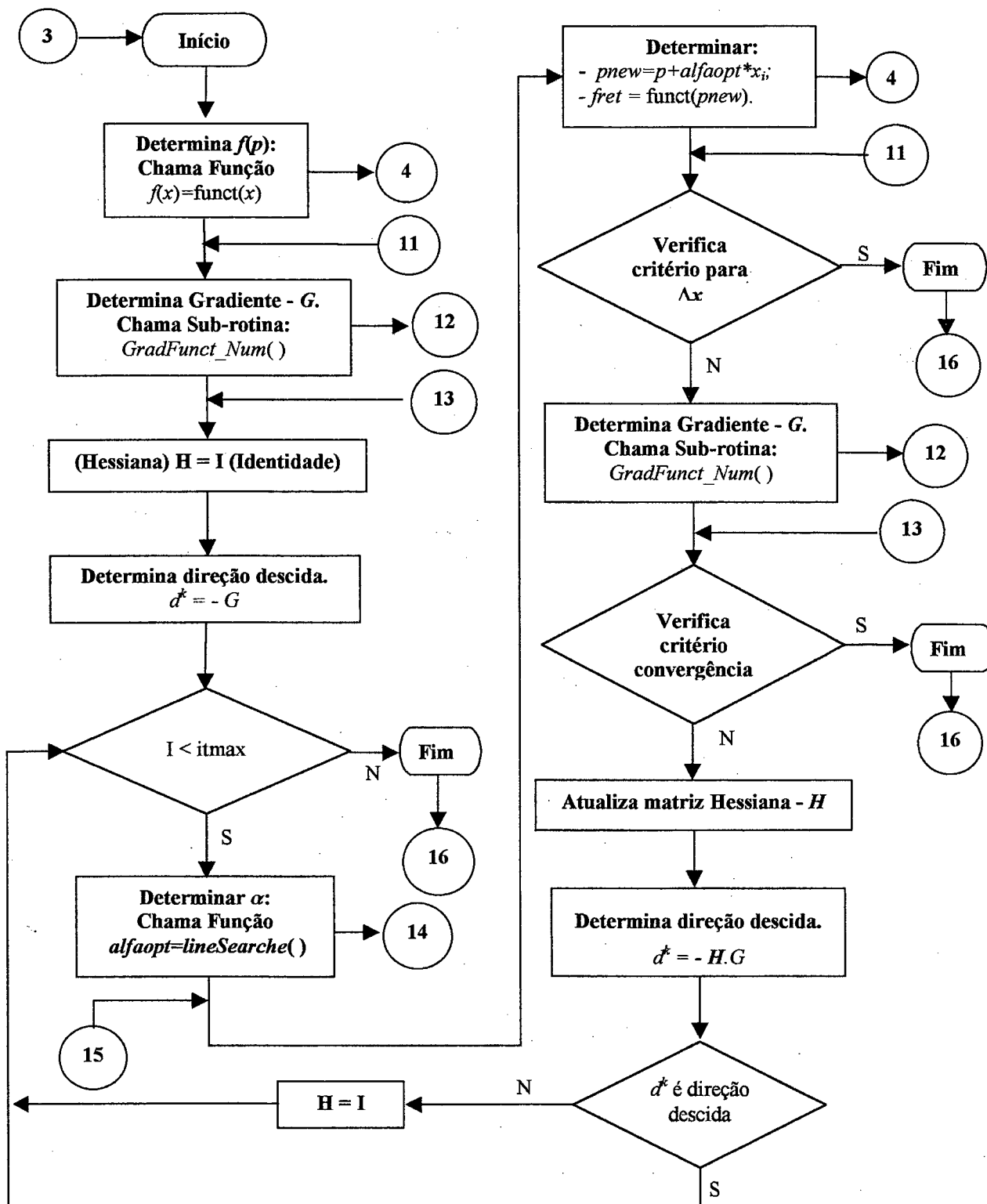


Figura 4.8 – Fluxograma da sub-rotina **dfpmin**.

Função $\text{Funct}(x)$

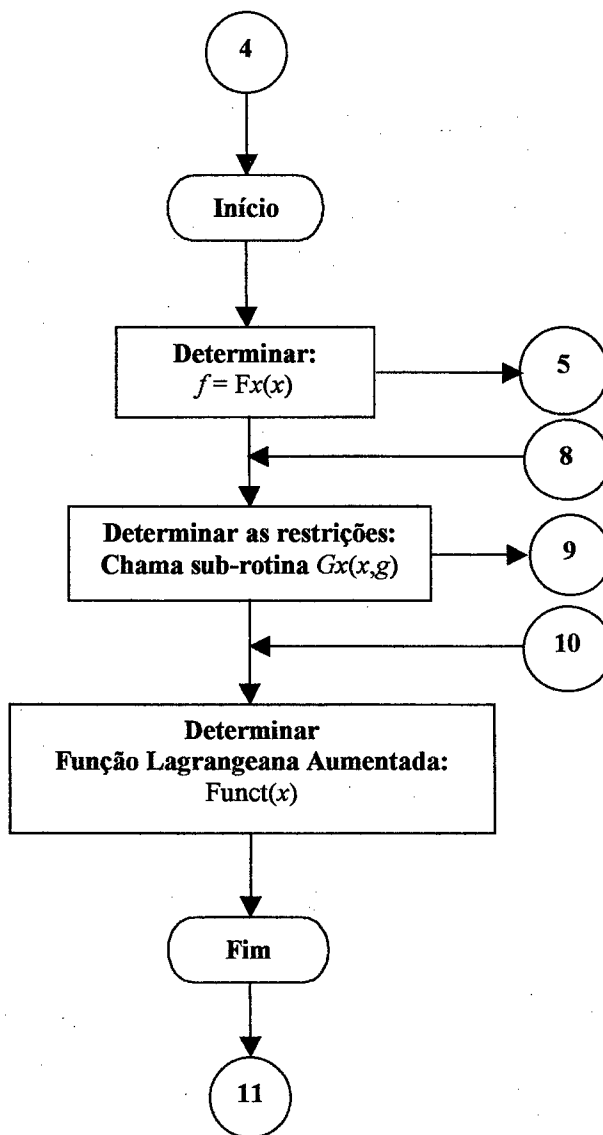


Figura 4.9 - Fluxograma da função $\text{Funct}(x)$.

Função $Fx(x)$

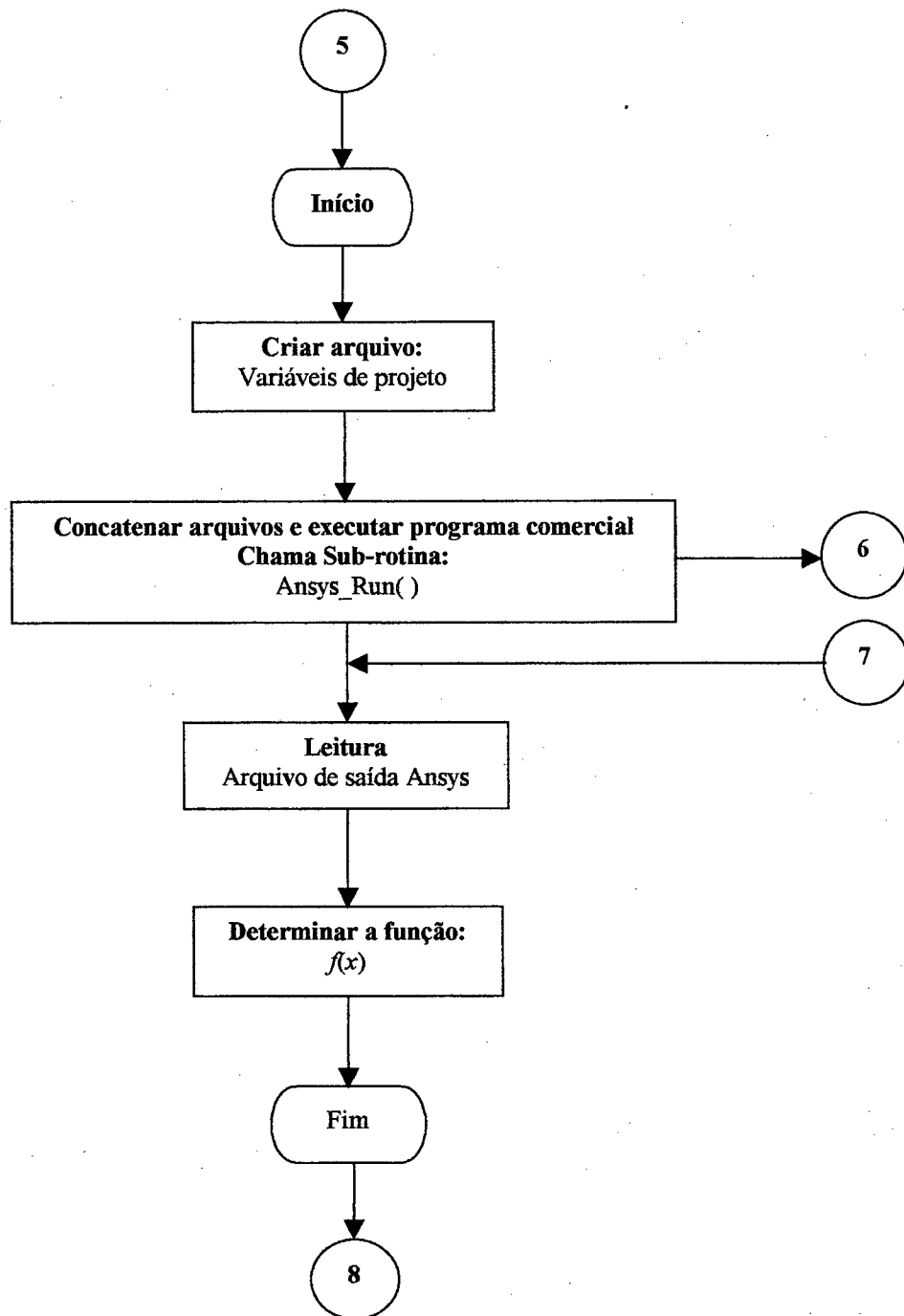


Figura 4.10 – Fluxograma da função $Fx(x)$.

Sub-rotina Ansys_Run

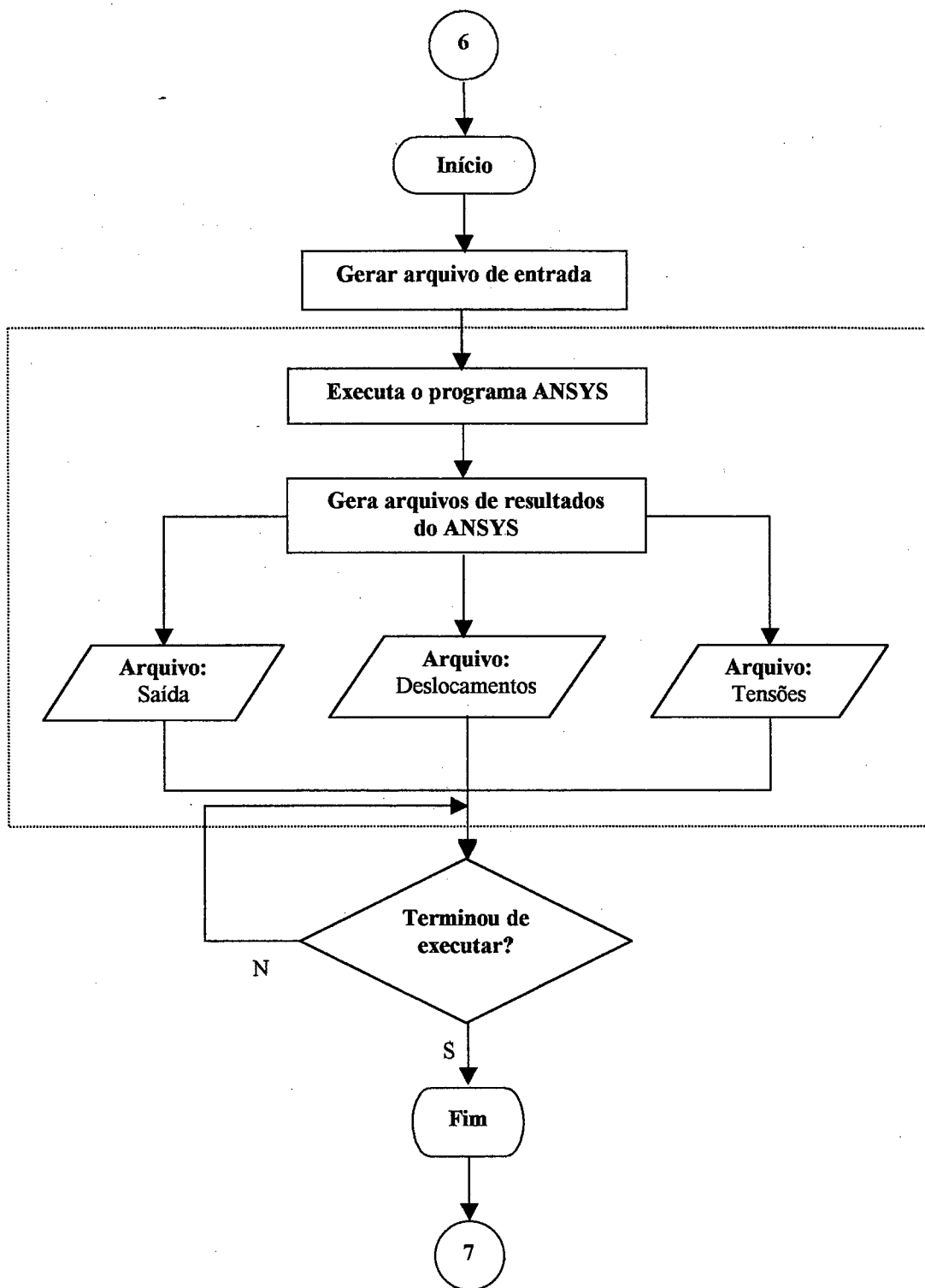


Figura 4.11 – Fluxograma da sub-rotina Ansys_Run.

Função $Gx(x,g)$

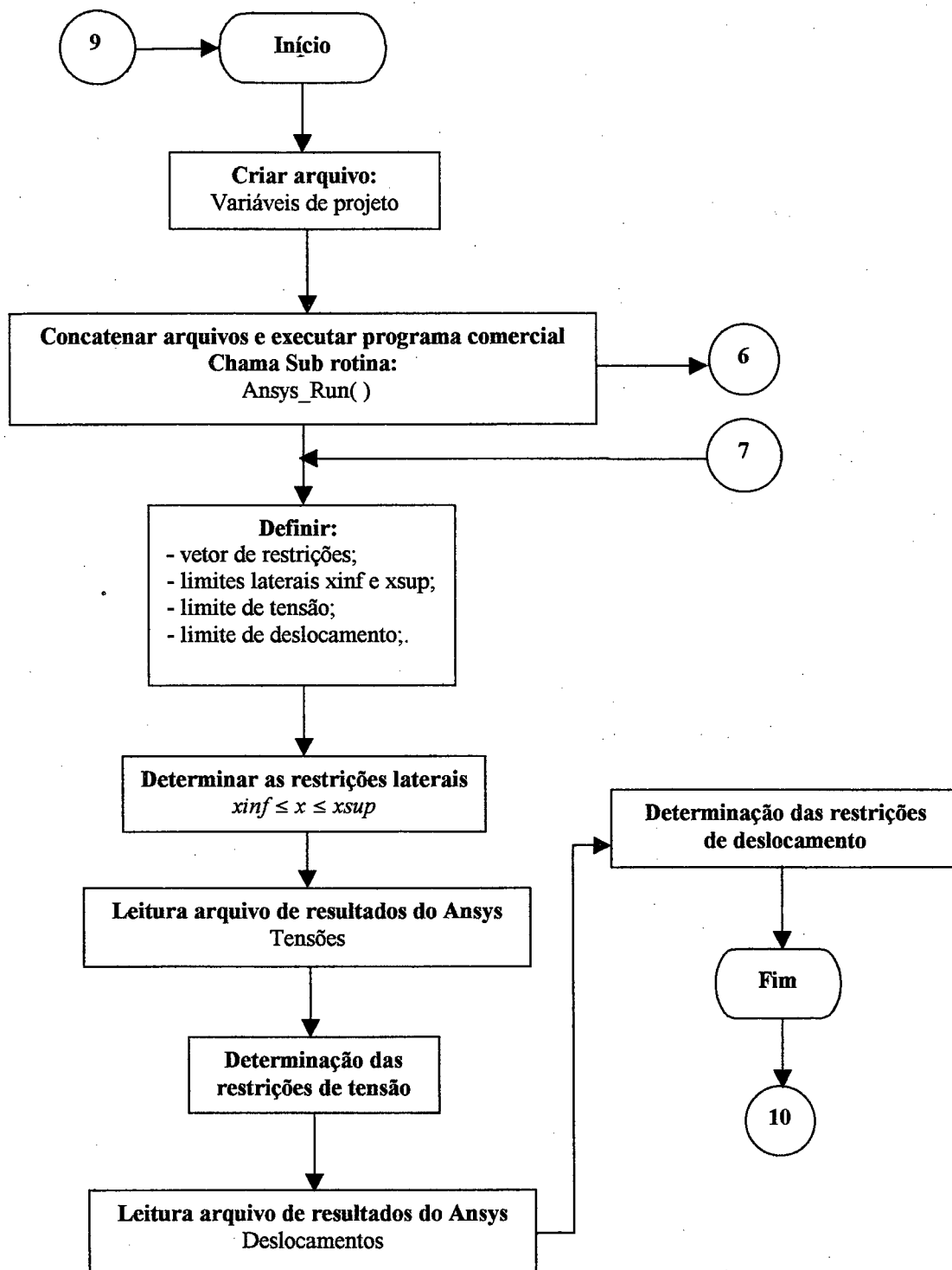


Figura 4.12 – Fluxograma da sub-rotina $Gx(x,g)$.

Sub-rotina GradFunct_Num

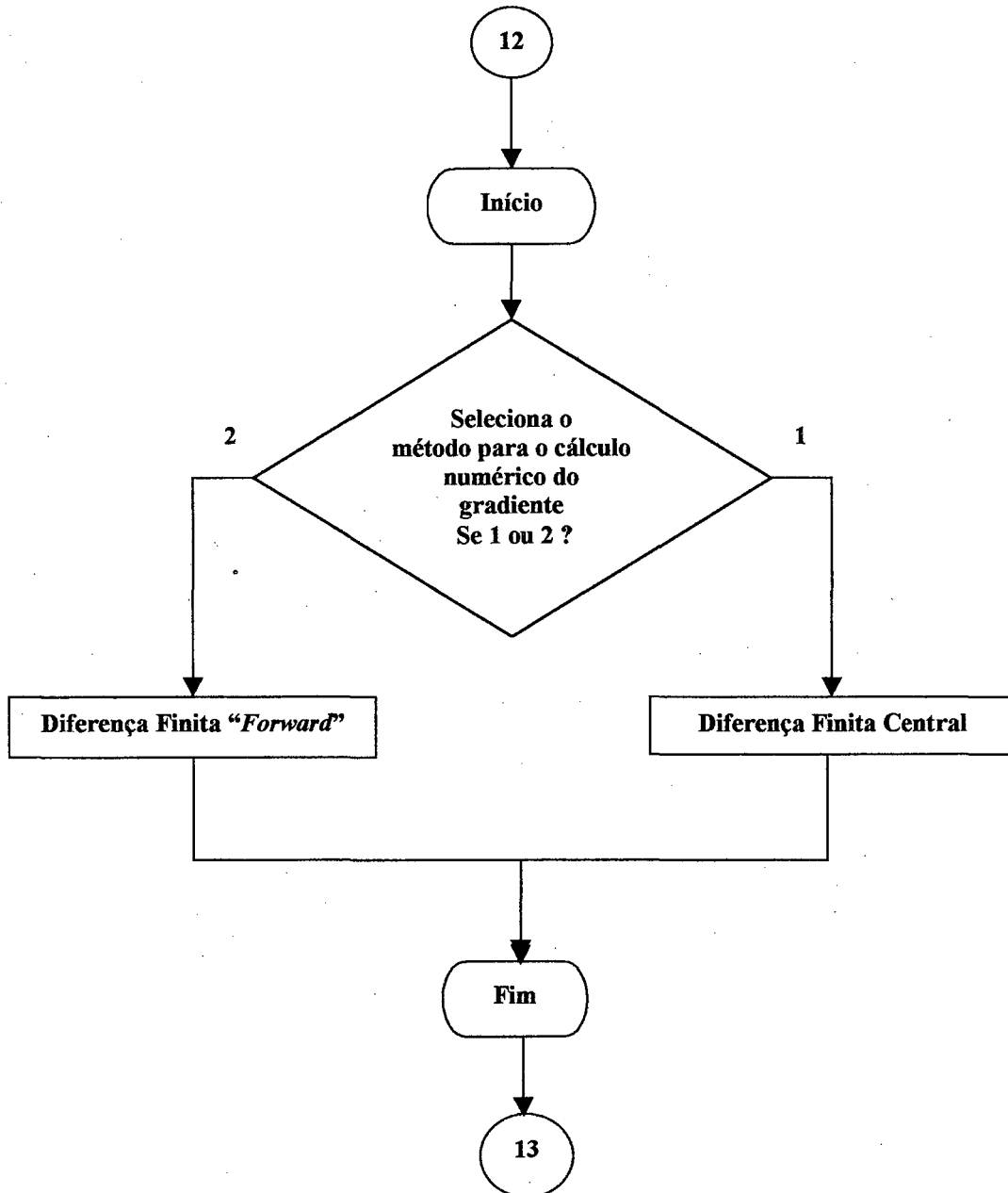


Figura 4.13 – Fluxograma da sub-rotina GradFunct_Num.

Função LineSearch

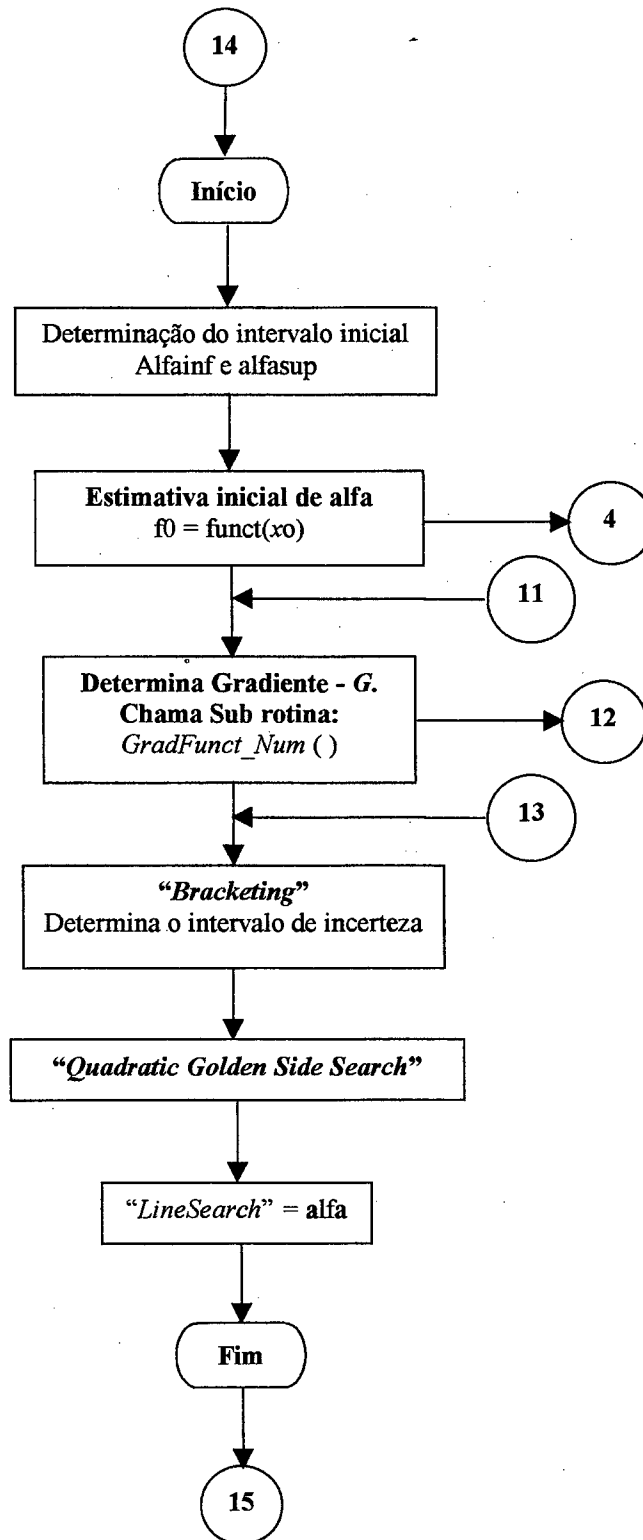


Figura 4.14 - Fluxograma da função LineSearch.

4.4.5 Procedimento para Programação

Após o entendimento do problema e ciente do que deve ser otimizado, quais as variáveis que podem ser manipuladas e as restrições às quais o sistema está sujeito, os seguintes passos devem ser feitos para programação do procedimento proposto, preparando-o para proceder à otimização de uma determinada estrutura ou componente:

1. Primeiramente, com os dados em mãos, formule o problema matematicamente, descrevendo a as variáveis de projeto, a função objetivo e funções restrições;
2. Proceda a construção do modelo. Para isto através de um editor de texto escreva o arquivo de entrada do programa comercial (ANSYS). Utilize o recurso do pré-processamento do programa ANSYS. Nesta fase é importante saber o tipo de resposta que se deseja para poder definir a função objetivo e função restrição. Portanto, através do pós-processamento com os comandos /OUTPUT, PRNSOL, /FORMAT, já descritos anteriormente, defina o arquivo de saída e sua formatação. Existem ainda outras três maneiras de se obter o arquivo de entrada para o programa ANSYS:

- O arquivo pode ser obtido a partir do arquivo .LOG do ANSYS. Este arquivo é obtido quando se entra pela primeira vez no programa no modo interativo. Neste caso, todo pré-processamento, solução e pós-processamento é realizado através de acesso ao programa ANSYS. O arquivo .LOG contém todos os comandos que foram usados durante a sessão, portanto é necessário que o mesmo seja editado para proceder à limpeza do mesmo, deixando somente os comandos pertinentes à descrição do problema, já que os outros não interferem no resultado final. A limpeza do arquivo .LOG deve ser feita por um usuário experiente para que não se cometa erros comprometendo o arquivo. Como segurança é interessante criar uma cópia de segurança;

- Outra maneira é o uso do arquivo .DB gerado pelo programa ANSYS, que pode ser convertido para formato ascii;

- Uma outra opção é através de um programa de CAD, como o PRO/E, PATRAN, ou outros que possuem o módulo para geração de modelo de elementos finitos;

3. Divida o arquivo .ANS ou .DAT ou .LOG (arquivo de entrada) de modo que se obtenha o arquivo com as variáveis de projeto disposto no formato ANSYS, permitindo que o mesmo seja mudado a cada iteração;
4. Na rotina **Main** proceda à definição do número de variáveis de projeto e restrições e inicialize o vetor variável de projeto;
5. Na função **F \mathbf{x} (\mathbf{x})**, crie o arquivo de variáveis de projeto correspondente, com comandos ANSYS, através da linguagem de programação *Fortran 90*. Neste ponto, quando o processo de otimização é iniciado a rotina faz, de modo automático, a leitura do valor da função objetivo do arquivo .OUT, se for a massa, ou de algum outro arquivo que o usuário determine caso tenha formulado um outro tipo de função objetivo.
6. Na sub-rotina **G \mathbf{x} (\mathbf{x})**, defina os limites laterais para as variáveis de projeto e os limites com relação as variáveis de estado (tensão e deslocamento). Nesta sub-rotina gere também o arquivo ANSYS com as variáveis de projeto como no item 4. Formule as restrições laterais, e as restrições de estado impostas para o problema, e com o comando OPEN (*Fortran 90*), se procede o acesso ao arquivo de resultados gerado pelo ANSYS, de onde é feito a leitura dos valores de interesse para computar a análise das restrições;
7. Na sub-rotina ANSY_RUN, para acesso ao programa ANSYS, estabeleça a integração utilizando a linha de comando que permite executar o programa comercial. Por exemplo, o comando para executar o programa ANSYS utilizado neste trabalho possui a seguinte sintaxe:

>ANSYS53 -b -i "c:\ESTUDO\shape\input_viga.dat" -o "c:\ESTUDO\shape\viga.out" ;
8. Após os acertos necessários em cada sub-rotina compile e linke as rotinas de otimização. Agora basta executar o programa e aguardar o resultado;
9. Para um outro tipo de programa proceda as alterações nas sub-rotinas, conforme os itens 1 a 8.

Capítulo 5

Resultados

5.1 Introdução

Neste capítulo serão apresentados alguns problemas objetivando demonstrar a aplicabilidade do procedimento de otimização proposto. Os resultados numéricos dos exemplos, são mostrados através de gráficos que permitem avaliar o comportamento da função a ser otimizada em cada iteração. São mostradas também, tabelas onde são comparados os resultados obtidos com alguns encontrados na bibliografia. Com esta exposição pode-se verificar que o uso deste procedimento de otimização, é viável em um ambiente de projeto.

Inicialmente, é apresentado um exemplo de viga engastada em três diferentes situações de problemas de otimização. Aproveitando este mesmo exemplo também foi utilizado para otimização da massa, dois programas distintos de elementos finitos, o ANSYS, e um programa acadêmico desenvolvido para solução de vigas. Desta forma verificou-se o aspecto genérico quanto à integração com programas de elementos finitos comerciais ou não.

Por tratar-se de um procedimento que utiliza um programa comercial de elementos finitos, é apresentado também alguns exemplos que comparam os resultados de otimização obtidos dos procedimentos de otimização do programa ANSYS com os resultados do procedimento proposto e de referências, quando disponível. Desta forma pode-se ter uma idéia, principalmente do grau de precisão de cada método.

A seguir é dada a descrição de cada problema e os resultados numéricos obtidos.

5.2 Exemplo 1 – Viga Engastada

5.2.1 Caso 1 – Otimização de volume

Inicia-se com um exemplo comparativo que utiliza o programa ANSYS e um programa de elementos finitos acadêmico, desenvolvido para modelos de viga. A estrutura 2D é uma viga engastada como mostrado na figura 5.1, e o problema objetiva determinar o volume ótimo da estrutura, respeitando as restrições de tensão e de deslocamento, além das restrições laterais imposta às variáveis de projeto.

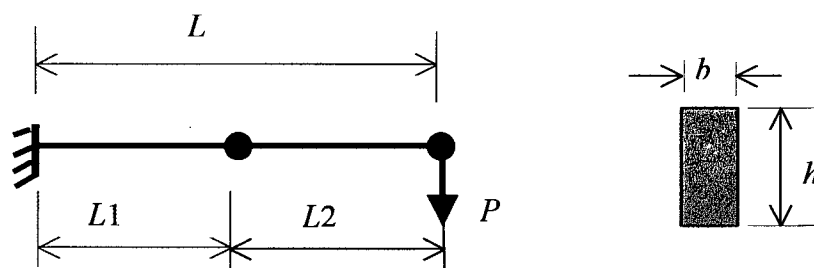


Figura 5.1 – Modelo da viga engastada.

Tabela 5.1 – Propriedades do modelo da viga engastada.

Propriedades materiais	$E = 2,1E5 \text{ MPa}$
	$\nu = 0,3$
	$\sigma_{adm.} = 400 \text{ MPa}$
Propriedades Geométricas	$b = 6 \text{ mm}$
	$L1 = 30 \text{ mm}$
	$L2 = 30 \text{ mm}$

Como variável de projeto é escolhido a altura da viga (h_i), sendo que a espessura (b) da mesma permanece inalterada ao longo de todo processo iterativo.

O modelo de elementos finitos é discretizado com dois elementos de viga e a condição de carregamento $P=5\text{ N}$ é colocada na extremidade oposta ao engaste na direção vertical, conforme indicado na figura 5.1. As características da estrutura estão indicadas na tabela 5.1.

O problema de otimização é matematicamente, descrito da seguinte forma:

Encontrar um $x = \{h_1, h_2\}$, vetor variável de projeto, que otimiza

$$f(x_i) = \text{Volume} = b \cdot x_i \cdot (L1 + L2) \tag{5.1}$$

Sujeito à

$$\text{restrições laterais: } x_{\text{inf}} \leq x_i \leq x_{\text{sup}} \tag{5.2}$$

$$\text{restrições de tensão: } -\sigma_{\text{inf}} \leq \sigma_i \leq \sigma_{\text{sup}} \tag{5.3}$$

$$\text{restrições de deslocamento: } u_{\text{inf}} \leq u_i \leq u_{\text{sup}} \tag{5.4}$$

Os limites para as restrições são indicados na tabela 5.2.

Tabela 5.2 - Limites para as restrições - exemplo 1.

Variável de projeto	$x_{\text{inf.}} = 1,0\text{ mm}$
	$x_{\text{sup.}} = 10,0\text{ mm}$
Limites de tensão	$\sigma_{\text{inf.}} = -400\text{ MPa}$
	$\sigma_{\text{sup.}} = 400\text{ MPa}$
Limites de desloc.	$u_{\text{inf.}} = 0.005\text{ mm}$
	$u_{\text{sup.}} = 0.005\text{ mm}$

A figura 5.2 mostra o modelo com as condições de contorno e os resultados de deslocamento e de tensões da configuração inicial.

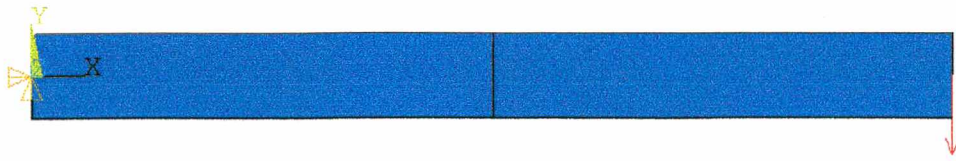


Figura 5.2 (a) - Modelo da viga engastada - configuração inicial.

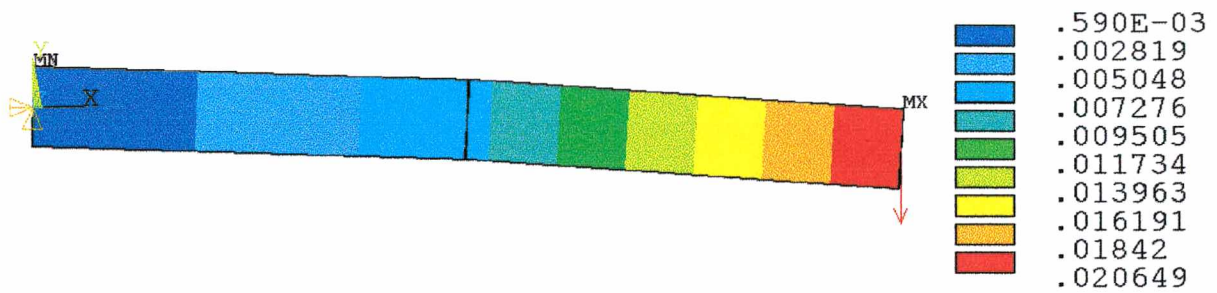


Figura 5.2 (b) - Deslocamento equivalente [mm] - configuração inicial.

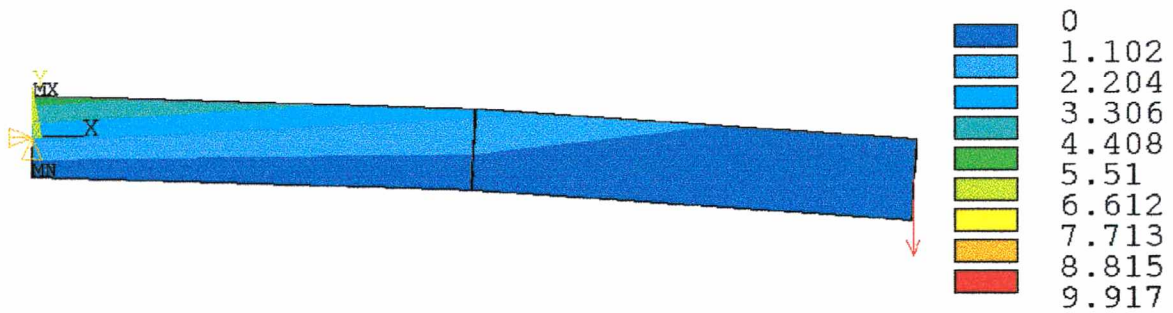


Figura 5.2 (c) - Distribuição da tensão principal - S1 [MPa] - configuração inicial.

Na figura 5.3 estão ilustrados os resultados de deslocamento e tensão para a configuração final.

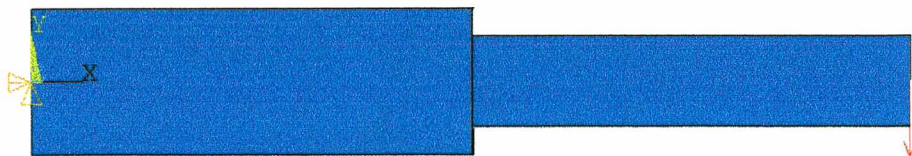


Figura 5.3 (a) - Modelo da viga engastada - configuração final.

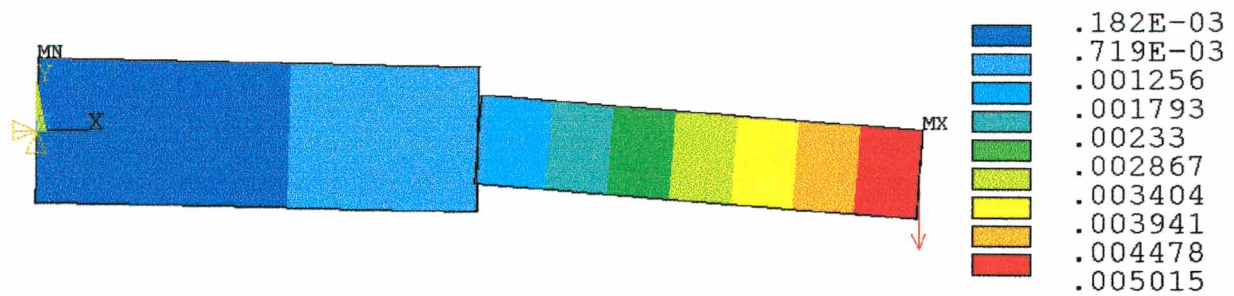


Figura 5.3 (b) - Deslocamento equivalente [mm] - configuração final.

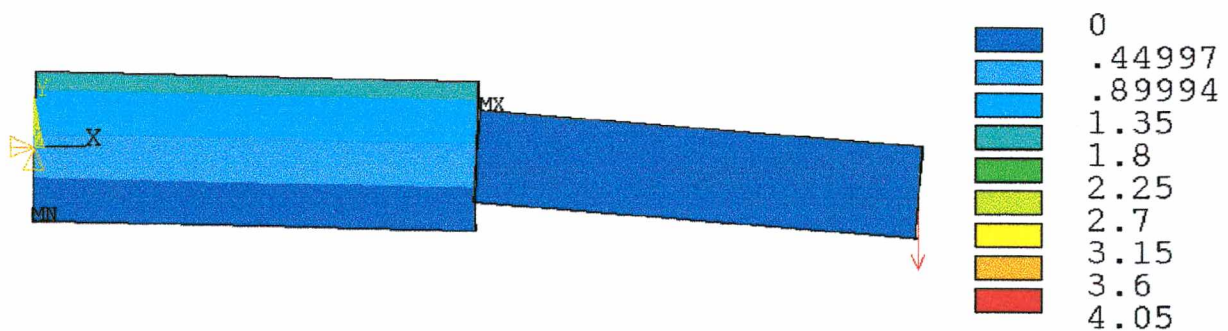


Figura 5.3(c) - Distribuição da tensão principal - S1 [MPa] - configuração final.

Na tabela 5.3 está indicado os resultados para a configuração inicial e o resultado do processo de otimização, utilizando o procedimento proposto com dois programas de elementos finitos (ANSYS e programa acadêmico), onde pode-se comparar os resultados.

Tabela 5.3 – Resultados: configuração inicial x final (ANSYS x Acadêmico).

	Valores iniciais	Valores finais ANSYS	Valores finais Prog. Acadêmico	Razão (P.Ac/ANSYS)
h_1 [mm]	5,500E+00	9,8927400530E+00	9,8943215269E+00	1,0001598621E+00
h_2 [mm]	5,500E+00	6,0860099468E+00	6,0842441157E+00	9,9970985405E-01
Volume [mm ³]	1,980E+03	2,8761386752E+03	2,8761420902E+03	1,0000011874E+00

Pode-se verificar que os valores finais para as variáveis de projeto em ambos os programas são bastante semelhantes. Neste sentido, as pequenas variações que possam ocorrer, quanto a precisão, está relacionado com a resposta que cada programa de elementos finitos oferece em termos de deslocamentos e tensões. Para verificar a convergência é esboçado um gráfico, mostrado na figura 5.4, onde se pode verificar o comportamento da função objetivo em cada iteração.

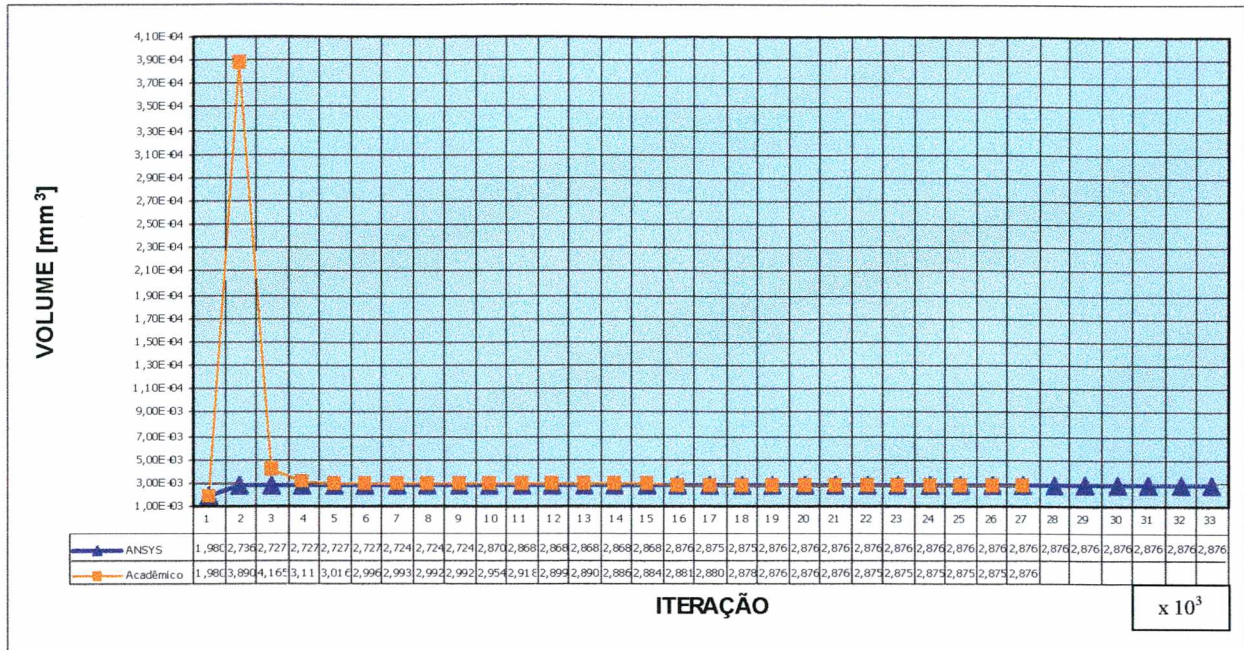


Figura 5.4 – Convergência da função objetivo (ANSYS x Programa Acadêmico).

Neste processo verifica-se somente uma discrepância inicial entre os dois programas de elementos finitos, mas após isto as funções passam a comportar-se de mesmo modo, assumindo o mesmo comportamento até a convergência.

5.2.2 Caso 2 – Determinação do módulo de elasticidade

É comum em engenharia encontrar situações onde se necessita a partir de testes, como por exemplo, teste de flexão, determinar a rigidez do material em questão. Sendo assim, foi elaborado este exemplo, que tem um importante papel prático em um ambiente de projeto ou num laboratório.

O problema de otimização proposto é encontrar a propriedade mecânica de rigidez, isto é o módulo de elasticidade, a partir do deslocamento determinado previamente por um teste de flexão ou calculado para viga engastada com carga $P=8$ N na extremidade, figura 5.1. Matematicamente, pode-se representar o problema da seguinte forma:

Encontrar E , para minimizar a função

$$f(x_i) = \left(1 - u(x_i) / u_o\right)^2 \quad (5.5)$$

sujeito à

$$\text{restrições laterais: } x_{\text{inf}} \leq x_i \leq x_{\text{sup}} \quad (5.6)$$

a variável de projeto x_i representa o módulo de elasticidade (E) do material, e os valores limites para a variável de projeto são:

$$x_{\text{inf}} = E_{\text{inf}} = 1.0\text{d}+04 \text{ MPa}; \quad x_{\text{sup}} = E_{\text{sup}} = 5.0\text{d}+06 \text{ MPa} \quad (5.7)$$

A seguir, na tabela 5.4, é indicado o resultado do processo iterativo, os valores de referência, os valores iniciais, e os valores finais do problema.

Tabela 5.4 – Resultados da determinação de E .

	Valores iniciais	Valores de referência	Valores Finais Proc. Proposto
E [MPa]	2,50050E+06	2,10000E+05	2,10000E+05
Função	8,39087E-01	0.0	0.0

A figura 5.5 dá uma idéia do comportamento da função objetivo e da variável de projeto ao longo de cada iteração.

Este exemplo é interessante pelo aspecto prático, pois mostra uma aplicação que pode ser utilizada em um ambiente de laboratório, onde muitas vezes necessita-se determinar as características mecânicas de um determinado material. Desta forma com uso de um dispositivo simples para teste de flexão pode-se determinar a deflexão e a seguir com o procedimento proposto fazer a análise computacional para determinar o módulo de elasticidade.

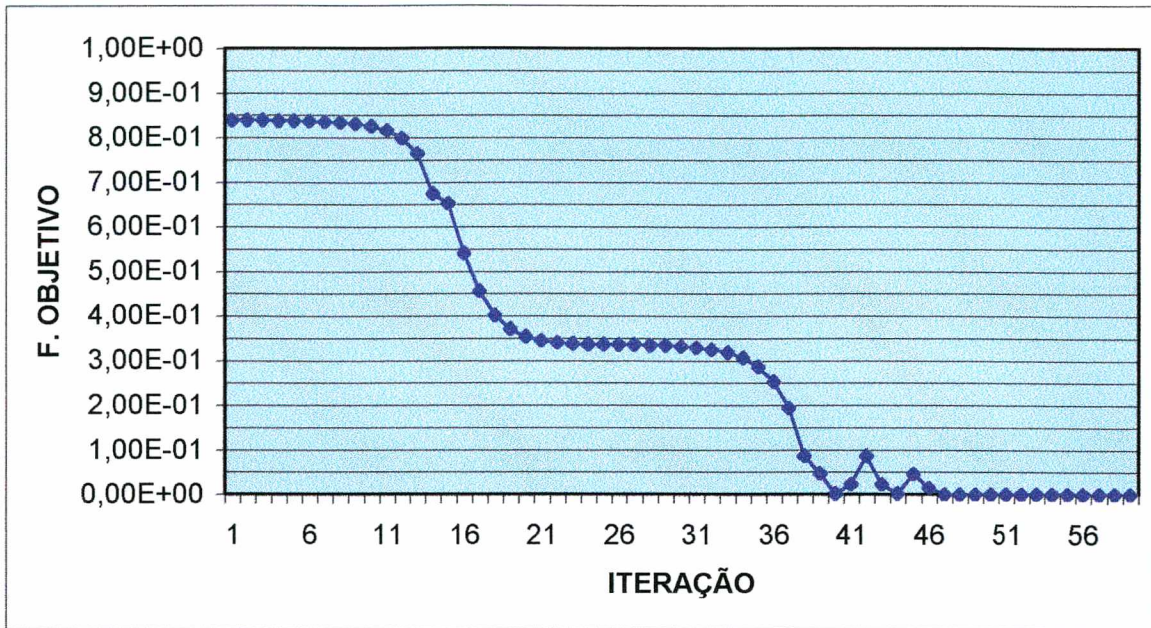


Figura 5.5 (a) – Convergência da função objetivo.

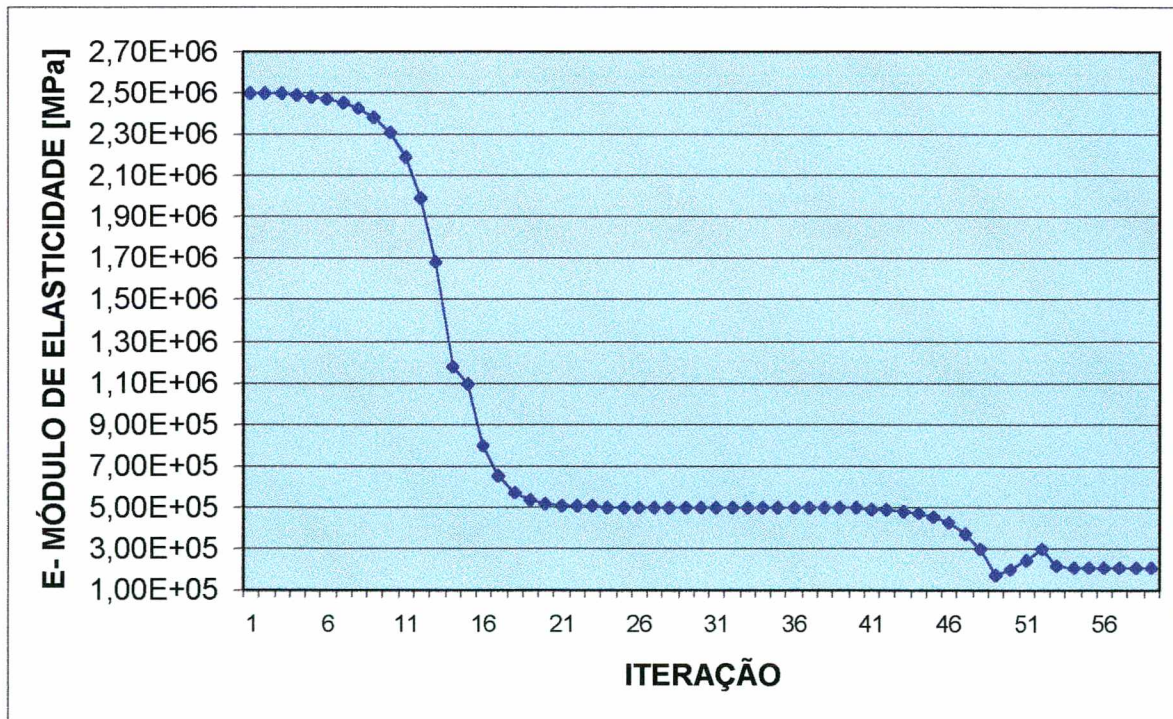


Figura 5.5 (b) – Convergência da variável de projeto.

5.2.3 Caso 3 - Otimização do trabalho para deflexão de uma viga engastada

Pelo princípio da conservação da energia verifica-se que

$$W_E = Q \quad (5.8)$$

onde, W_E é trabalho total realizado pelas forças externas aplicadas na estrutura durante o processo de carregamento e Q é a energia total de deformação armazenada no sistema. Das expressões para o cálculo de energia de deformação elástica de vigas e fazendo uso da lei de conservação de energia pode-se obter a deflexão de membros de uma estrutura. Para o caso de uma viga engastada, figura 5.1, com uma força $P=8$ N aplicada na extremidade, o trabalho devido a força externa é dado por

$$W_E = \frac{1}{2} \cdot P \cdot u(x_i) \quad (5.9)$$

onde, $u(x_i)$ é a deflexão na extremidade da viga, onde está aplicado a carga P . E a energia de deformação é dada por

$$Q = \frac{P^2 \cdot L^3}{6 \cdot E \cdot I} \quad (5.10)$$

sendo, L o comprimento da viga, E o módulo de elasticidade, e I o momento de inércia dado por:

$$I = \frac{b \cdot h^3}{12}, \text{ momento de inércia para uma seção retangular.} \quad (5.11)$$

Baseado na equação (5.8), determina-se a deflexão da viga, desconsiderando os efeitos de cisalhamento e a expressão toma a forma dada por:

$$u(x_i) = \frac{P \cdot L^3}{3 \cdot E \cdot I} \quad (5.12)$$

O objetivo deste problema é otimizar o trabalho calculado pela expressão W_E , sujeito a restrições laterais impostas sobre a variável de projeto x_i , que é a altura da seção transversal da viga. É imposto também, uma restrição quanto ao valor do volume, isto é, que o valor final do volume não seja superior a 70% do volume original da estrutura.

Matematicamente, o problema de otimização é descrito da forma:

Encontrar um vetor x_i , que satisfaça as restrições

de volume

$$V_f \leq 0,7.V_o, \quad (5.13)$$

e restrições laterais

$$x_{inf} \leq x_i \leq x_{sup} \quad (5.14)$$

para o otimizar o trabalho da força externa W_E dado por:

$$W_E = \frac{1}{2} \cdot P \cdot u(x_i) \quad (5.15)$$

O modelo de elementos finitos da viga está representado na figura 5.1, e os dados da tabela 5.1, mostram as propriedades e características geométricas do modelo considerado. Portanto, fazendo uso do programa de elementos finitos acadêmico desenvolvido para viga, obtém-se os deslocamentos e o volume da estrutura, e com o procedimento de otimização proposto inicia-se o processo para encontrar a solução do problema. Os valores limites estabelecidos para as restrições estão indicados na tabela 5.5.

Tabela 5.5 - Limites para as restrições - exemplo 1 - caso3.

Variável de projeto	$x_{inf} = 1,0 \text{ mm}$
	$x_{sup} = 10,0 \text{ mm}$
Volume	$V_{adm} = 1386 \text{ mm}^3$

A tabela 5.6 indica os resultados encontrados para a função objetivo, para as variáveis de projeto, o volume e o deslocamento na configuração final, bem como os valores iniciais para os respectivos itens analisados.

Tabela 5.6 – Resultados da otimização do trabalho.

	valores iniciais	valores finais
F. Objetivo	1,3189E-01	2,8595E-01
h1 [mm]	5,5000E+00	4,7684E+00
h2 [mm]	5,5000E+00	2,9316E+00
volume [mm ³]	1,9800E+03	1,3860E+03

O processo de convergência da função objetivo se mostra bem comportado ao longo das iterações, conforme indica o comportamento da curva na figura 5.6.

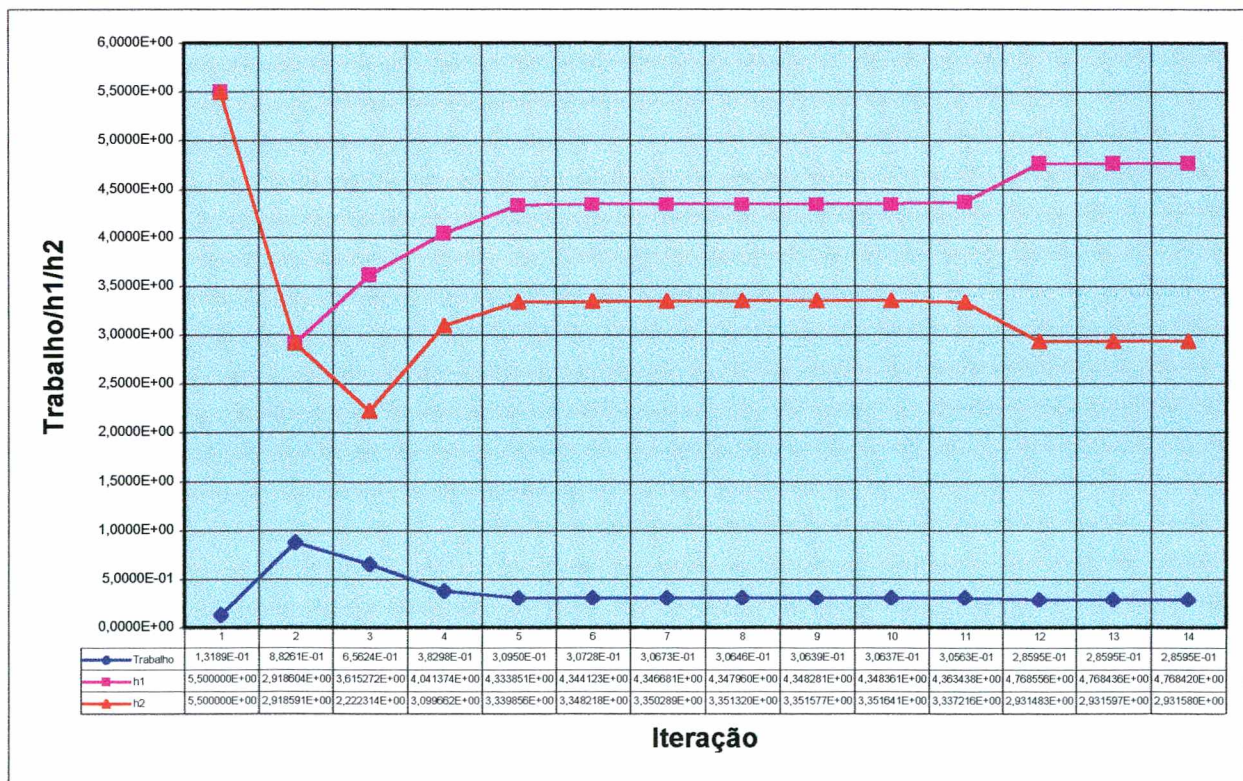


Figura 5.6 – Gráfico de convergência para função objetivo e variáveis de projeto.

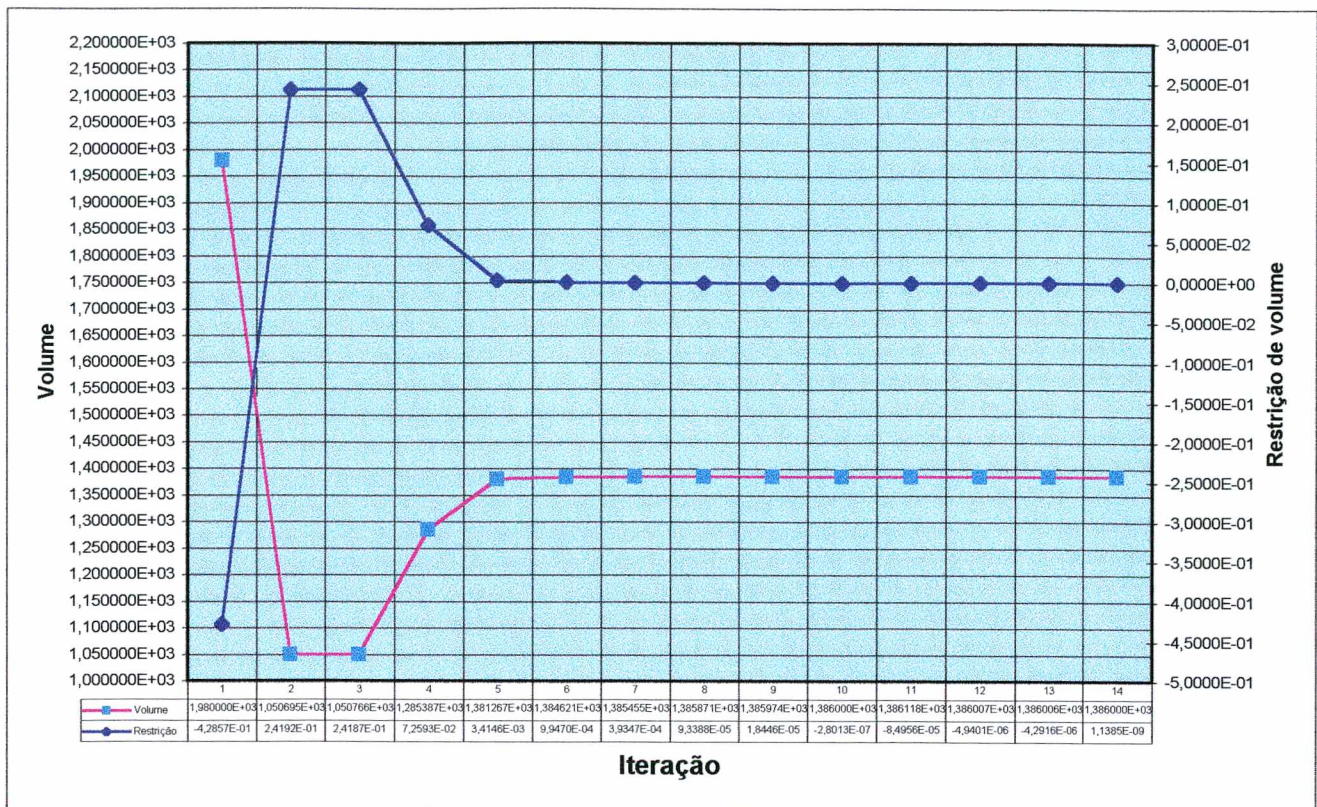


Figura 5.7 – Gráfico do comportamento do volume e restrição de volume.

O exemplo é interessante para mostrar uma outra aplicação para o procedimento proposto, e deixa claro até o momento a abrangência de problemas que se pode resolver com o uso deste procedimento. Com uso de diferenças finitas para o cálculo das derivadas e da determinação da direção de descida, a convergência se mantém bem comportada. Outro aspecto que pode ser observado na figura 5.7, é quanto ao comportamento do volume e a restrição imposta a este, mostrando os pontos onde a restrição imposta ao volume é violada, isto é onde a restrição tem o valor negativo. Pode verificado que ao final do processo iterativo o volume corresponde exatamente ao limite imposto, conforme indicado na tabela 5.5, o que assegura a convergência do processo.

5.3 Exemplo 2 – Pórtico Bi-dimensional

Neste exemplo é modelado um pórtico, cujo objetivo é minimizar o volume da estrutura que é formada de 4 elementos de viga, e sujeito à restrições de momentos fletores em cada extremidade dos elementos, com a seguinte forma [8]

$$K(d_j)^3 - |M_{ij}| \geq 0 \quad (5.16)$$

onde, M_{ij} é o momento fletor na extremidade i de cada elemento j , e d_j é a altura da seção transversal de cada elemento j . A área da seção dos membros da estrutura é retangular com espessura igual a metade da altura.

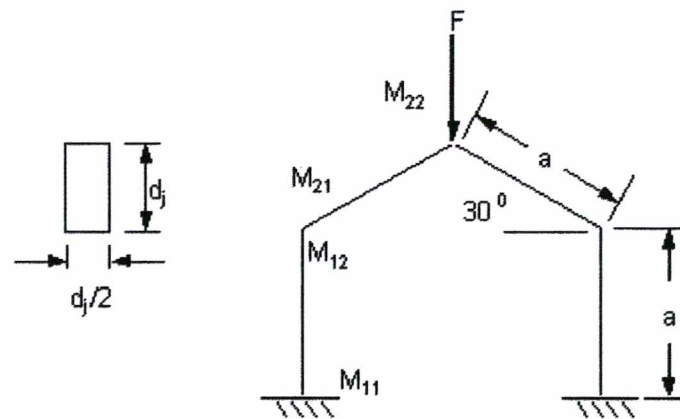


Figura 5.8 – Estrutura Pórtico bi-dimensional.

A figura 5.8, mostra o modelo geométrico da estrutura com a indicação das respectivas características geométricas e físicas.

O modelo de elementos finitos mostrado na figura 5.9 é construído utilizando a metade da estrutura, para isto é imposto condições de contorno de simetria sobre o nó que está na linha de centro do modelo. As áreas e momentos de inércia são expressos parametricamente a partir das variáveis de projeto como segue

$$A_j = \frac{1}{2} \cdot d_j^2 \quad (5.17)$$

$$I_j = \frac{1}{24} \cdot d_j^4 \tag{5.18}$$

onde A_j , é a área da seção transversal de cada elemento e, I_j , o momento de inércia.

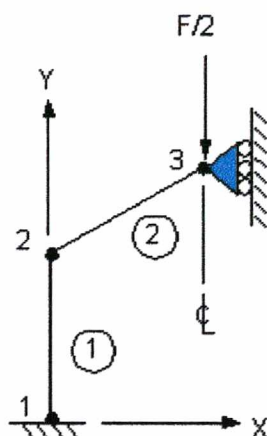


Figura 5.9 – Modelo de elementos finitos do pórtico.

O limite superior para a condição de restrição de estado imposta aos momentos em cada extremidade é 2000 N.m, e para restrições laterais os limites inferior e superior são, respectivamente, 0,05 m e 0,5 m . Na tabela 5.7, verifica-se as propriedades consideradas para este problema e o valor da carga F .

Tabela 5.7 - Propriedades: modelo pórtico 2D.

Propriedades materiais [Pa]	$E = 1 \times 10^{10}$
Propriedades geométricas [m]	$a = 2,5$
carga [N]	$F = 4000$

Na figura 5.10 é mostrado nas condições iniciais o resultado em termos de deslocamento da estrutura. Após o processo de otimização verifica-se os seguintes resultados indicados na figura 5.11.

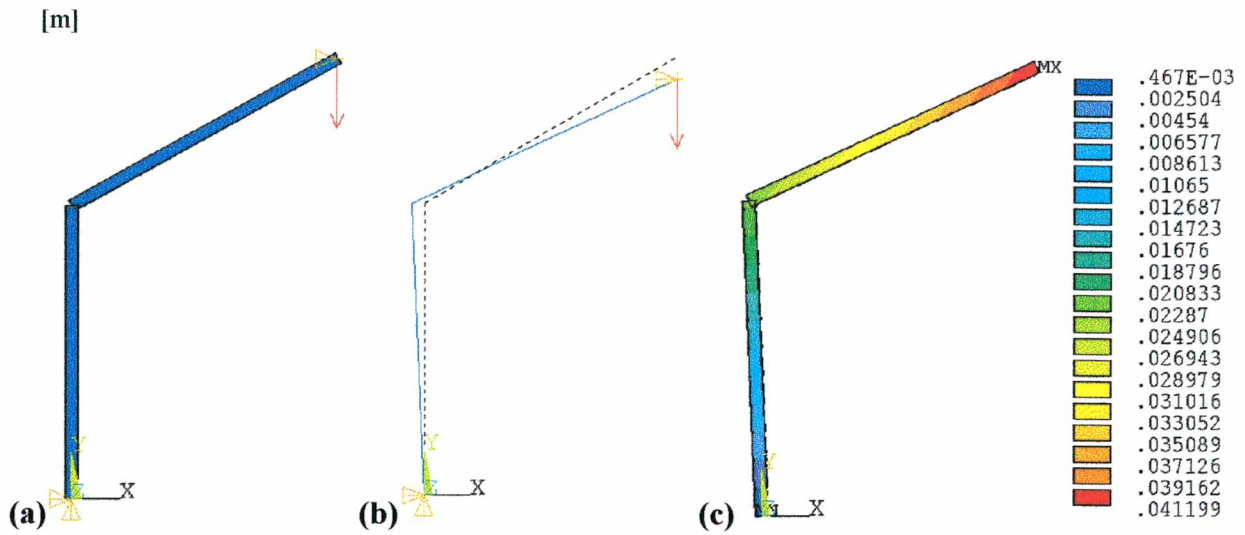


Figura 5.10 - Configuração inicial: (a) modelo; (b) deformada; (c) deslocamento equivalente.

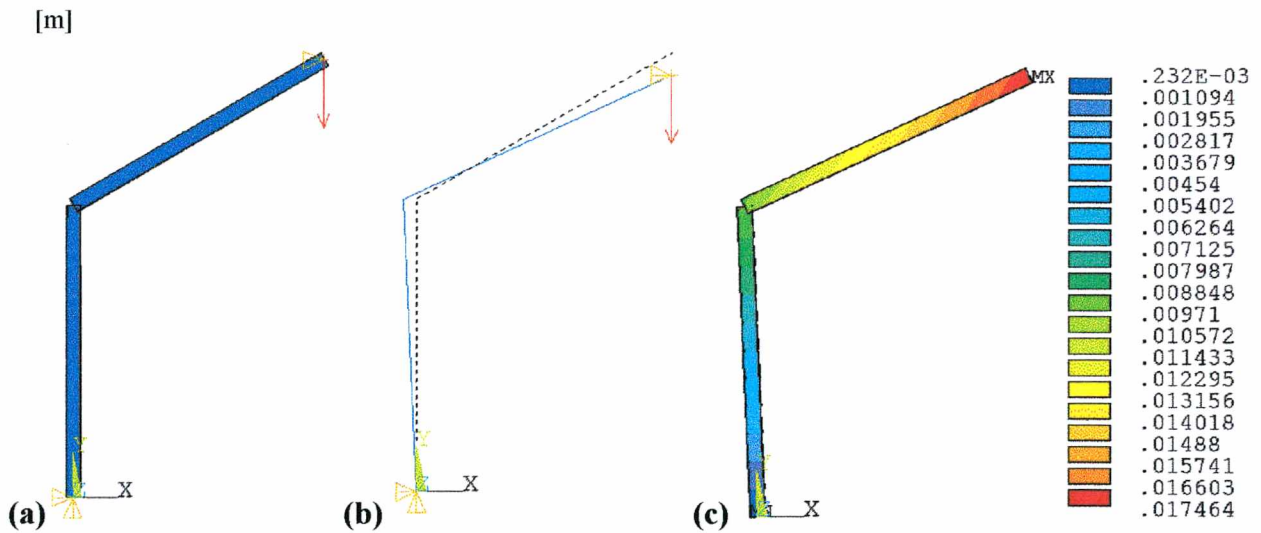


Figura 5.11 - Configuração final: (a) modelo; (b) deformada; (c) deslocamento equivalente.

Para uma análise comparativa entre o método proposto com o método utilizado pelo ANSYS está indicado na tabela 5.8.

Tabela 5.8 - Resultado comparativo entre os métodos de otimização – Pórtico 2D.

	Valores iniciais	Resultado referência [8]	M.S.A - ANSYS	Proc. Proposto
Volume [m³]	0,050000	0,076400	0,077100	0,076400
d1 [m]	0,100000	0,118000	0,119600	0,118140
d2 [m]	0,100000	0,129000	0,128800	0,129001
Momento (nó 1)	-1169,062736	-1058,116538	-1075,943930	-1059,508646
(nó 3)	1639,309596	1772,566682	1748,216611	1770,623059
Deflexão[m]	0,040730	0,017243	0,016884	0,017206
Razão	Método/Valor ref.	Volume	1,009162	1,000000
		d1	1,013559	1,001187
		d2	0,998450	1,000006

M. S. A . - Método do Subproblema Aproximado

Na figura 5.12 é indicado a evolução da função objetivo em relação a cada iteração do processo de otimização comparando o método interno do programa ANSYS com o procedimento proposto.

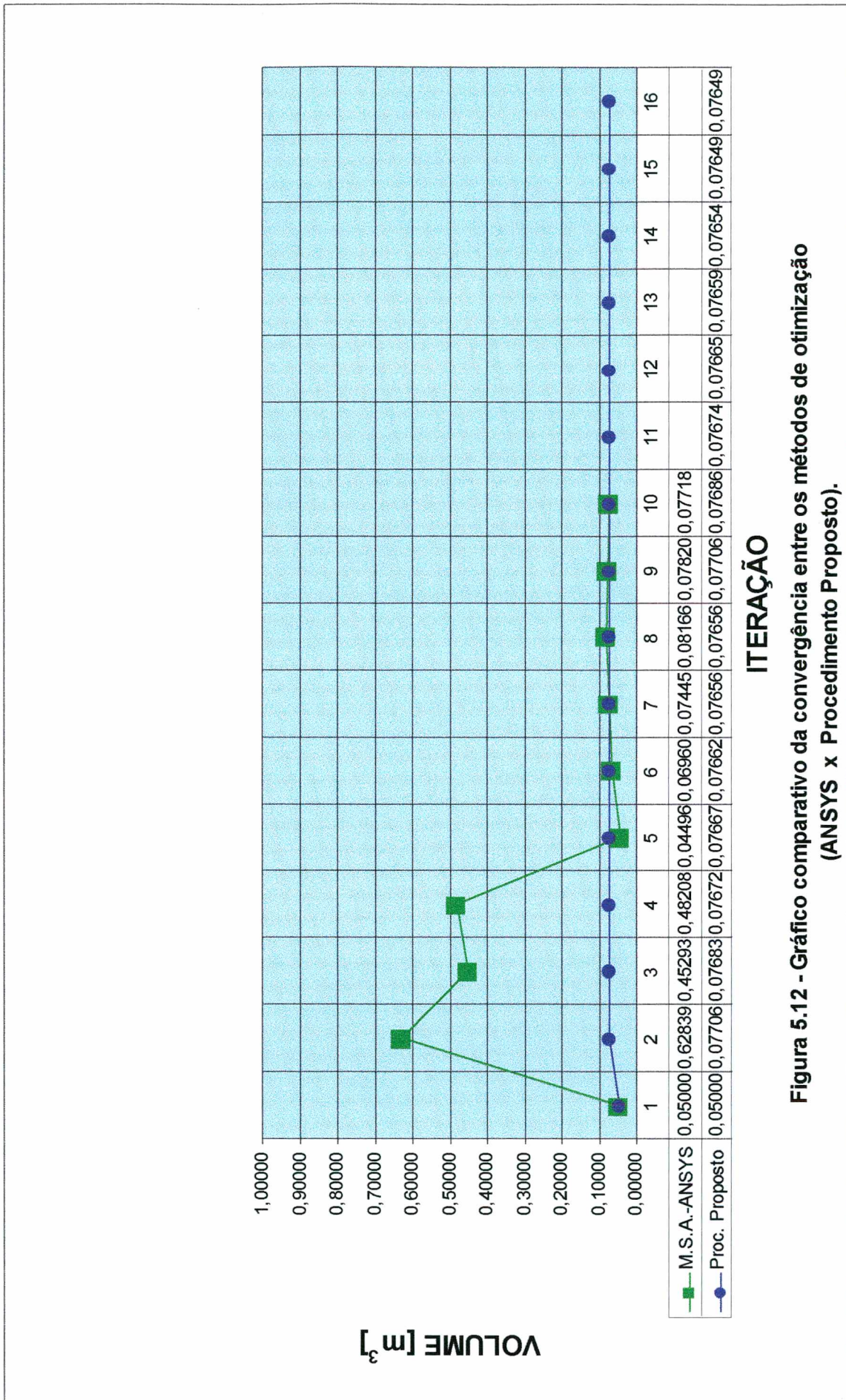


Figura 5.12 - Gráfico comparativo da convergência entre os métodos de otimização (ANSYS x Procedimento Proposto).

5.4 Exemplo 3 – Otimização de Forma

A finalidade deste exemplo é apresentar a otimização de forma de uma viga engastada, considerando estado plano de tensões (EPT), e comparar os resultados obtidos pelo procedimento proposto neste trabalho com os resultados apresentados pelo otimizador do programa ANSYS e o resultado da referência [8]. Sendo assim, pode-se validar a proposta de trabalho aqui exposta.

O objetivo na estrutura mostrada na figura 5.13 é otimizar o volume da viga, que está sujeita a um momento M . As restrições de estado imposta a estrutura são: a tensão máxima admissível de 30000 psi, e a deflexão máxima admissível, em módulo, de 0.5 in.



Figura 5.13 – Modelo geométrico da viga engastada (EPT).

Como variáveis de projeto foram escolhidos as espessuras ao longo do comprimento da viga, e sobre estas, são impostas restrições laterais que variam de um limite inferior de 0.15 in a um limite superior de 0.27 in. Considera-se a espessura na extremidade livre da viga permanece constante com valor de 0.15 in. A tabela 5.9, mostra algumas características da estrutura a ser otimizada.

O modelo de elementos finitos na configuração inicial é mostrado na figura 5.14(a). A estrutura é definida por k -pontos e a fibra superior da viga é formada pela conexão dos k -pontos através de uma “*spline*” cúbica. Pode-se notar que o modelo possui simetria, deste modo pode-se modelar somente metade da viga, impondo condições de simetria sobre os nós ou k -pontos que

estão na linha de centro da geometria. Portanto, no momento de computar o volume total da estrutura, deve-se multiplicar por dois o resultado obtido deste modelo.

Tabela 5.9 – Propriedades: viga engastada - E.P.T..

Propriedades materiais	$E = 10 \text{ E } +06 \text{ psi}$
	$\nu = 0.3$
Propriedades Geométricas	$L = 10 \text{ in}$
	$t = 0,3 \text{ in}$
	$b = 1 \text{ in}$
Carregamento	$M = 450 \text{ lbf. in}$

Matematicamente, o problema de otimização para esta estrutura pode ser assim descrito:

Minimizar

$$f(x) = \text{Volume} = b \cdot \sum_{i=1}^4 L_i \cdot TK_i \tag{5.19}$$

Sujeito à

<i>restrições de estado</i>	<i>restrições laterais</i>	
$\sigma \leq 30000$	$0.15 \leq TK_1 \leq 0.27$	
$\delta_{\max} \leq 0.5$	$0.15 \leq TK_2 \leq 0.27$	(5.20)
	$0.15 \leq TK_3 \leq 0.27$	
	$0.15 \leq TK_4 \leq 0.27$	

Os valores iniciais para as variáveis de projeto são

$$TK_1 = 0.25; TK_2 = 0.25; TK_3 = 0.25; TK_4 = 0.25 \text{ [in]} \tag{5.21}$$

A figura 5.14 (a), (b), (c), mostra o modelo de elementos finitos, os resultados de tensão e deslocamento máximo da viga na configuração inicial

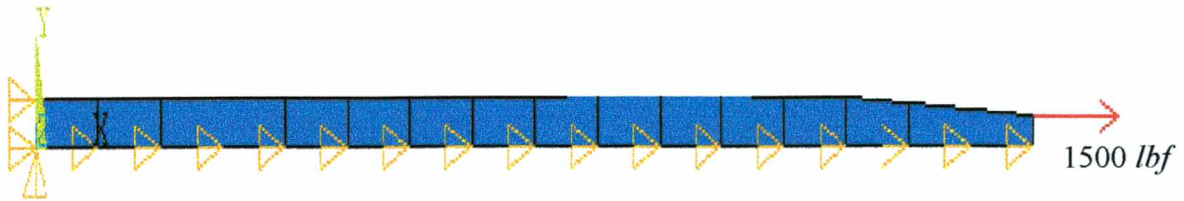


Figura 5.14 (a)- Modelo de elementos finitos da viga engastada (EPT) - configuração inicial.

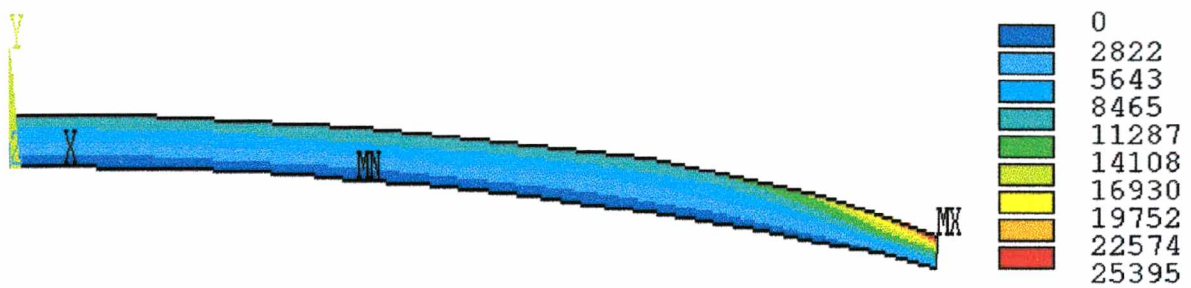


Figura 5.14 (b) – Tensões máxima [psi] - configuração inicial.

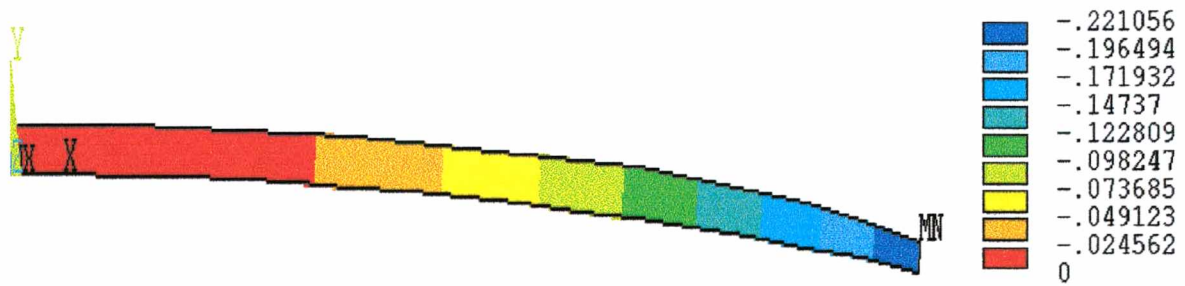


Figura 5.14 (c) – Deflexão máxima [in] - configuração inicial.

O modelo de elementos finitos da viga em sua configuração final é mostrada na figura 5.15.

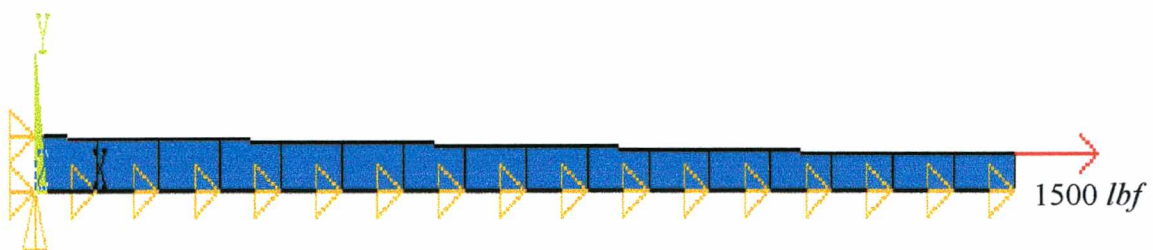


Figura 5.15 – Modelo de elementos finitos - configuração final.

Após o processo iterativo de otimização, pode-se mostrar os resultados da configuração otimizada da estrutura na figura 5.16 (a), (b)

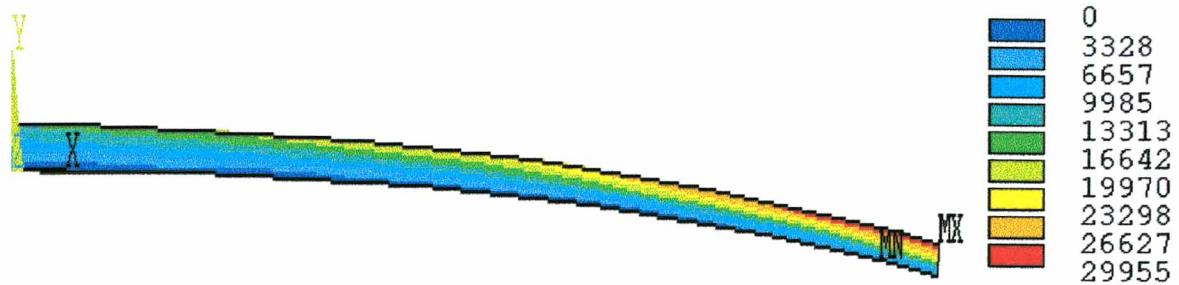


Figura 5.16 (a) – Tensão máxima [psi] - configuração final.

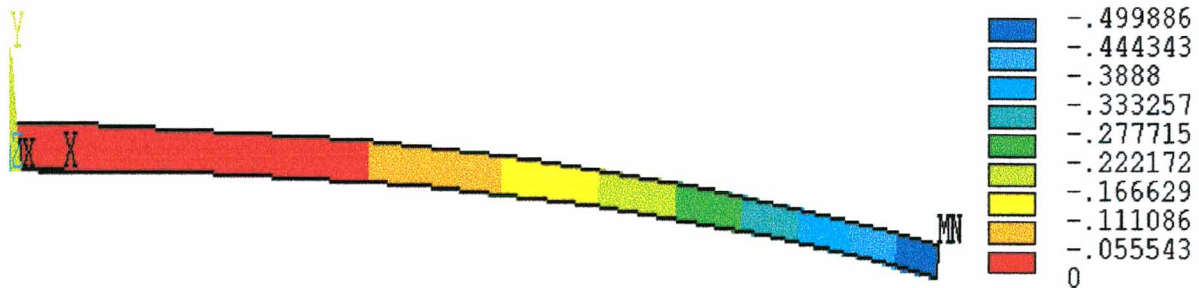


Figura 5.16 (b) – Deflexão máxima [in] – configuração final.

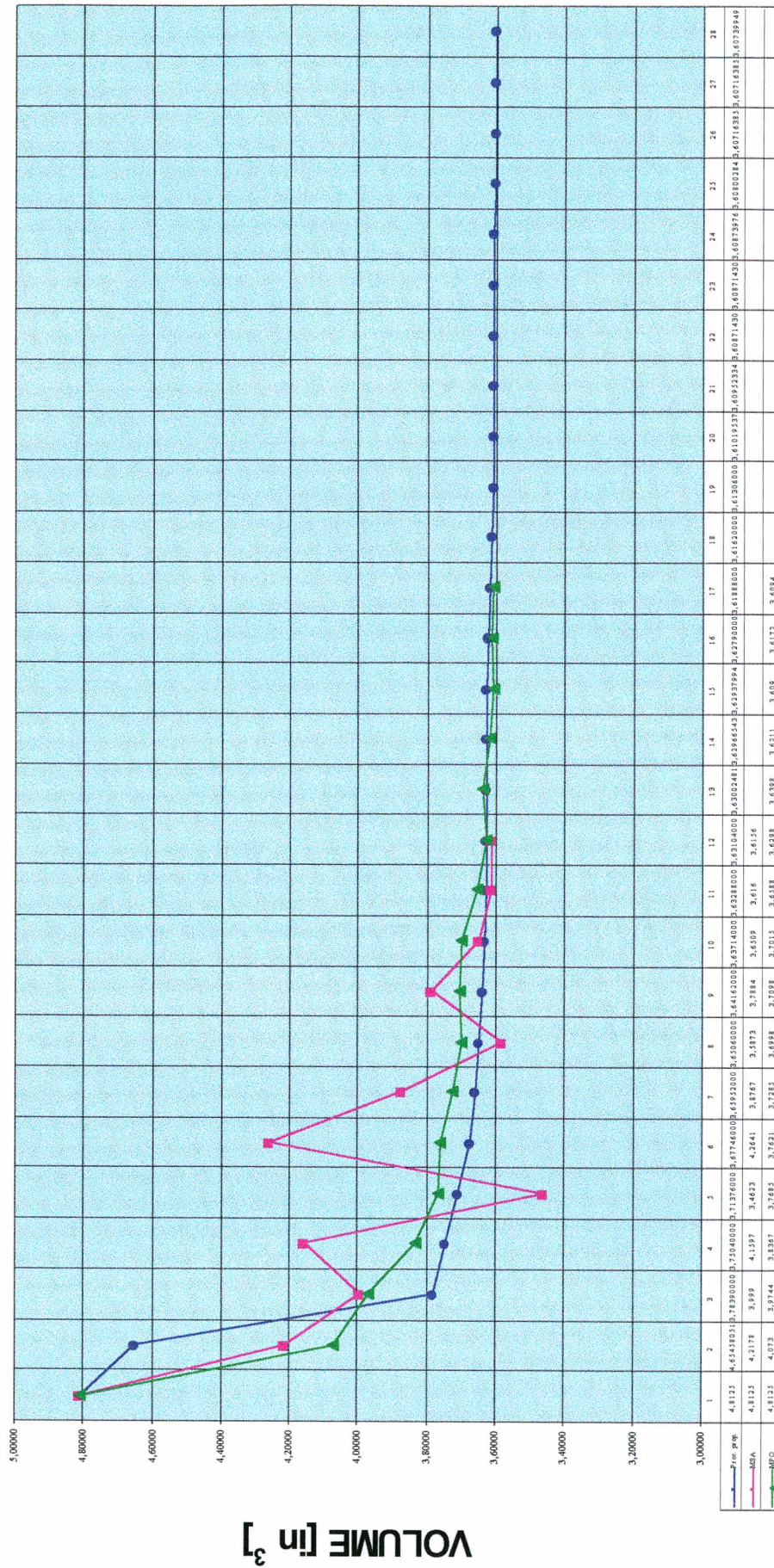
A validação do procedimento é comprovada através da comparação dos resultados obtidos com o resultado do otimizador do programa ANSYS e o resultado da referência [8], conforme indicado na tabela 5.10

Tabela 5.10 - Resultado comparativo entre os métodos de otimização - viga E.P.T..

	Resultado referência	M.S.A	M. P.O.	Proc. Proposto
Volume [in ³]	3,6	3,6156	3,609	3,6074
Deflexão máx. [in]	0,5	0,4992	0,5001	0,4998
Tensão máx. [psi]	30000	29740	29846	29955
Razão método/ ref.	VOLUME	1,0043	1,0025	1,0021
	Deflexão	0,9984	1,0002	0,9996
	Tensão	0,9913	0,9949	0,9985

M.P.O .- Método de Primeira Ordem - ANSYS;

M. S. A . - Método do Subproblema Aproximado - ANSYS.



ITERAÇÃO

Figura 5.17 - Gráfico comparativo da convergência entre os métodos de otimização (ANSYS x Procedimento proposto).

Na figura 5.17, é mostrado com mais detalhe a evolução da função objetivo ao longo do processo iterativo, comparando os dois métodos utilizados pelo programa ANSYS e o procedimento de otimização proposto.

5.5 Exemplo 4 – Caso Prático - Otimização de espessura de prateleira para refrigerador

Neste exemplo é verificado o uso do procedimento de otimização proposto para uma peça usada como prateleira plástica de um refrigerador. A peça fica apoiada sobre trilhos e o requisito de projeto é que suporte uma carga de 120 N, distribuída sobre a área útil da peça, figura 5.18, sendo que o critério de aprovação é que a flecha não ultrapasse 10 mm, salvo alguma restrição de projeto que seja observado. Esta peça possui uma limitação quanto a flecha, já que possui outro componente sob ela. Portanto, para que a mesma não tenha interferência com outro componente a sua deflexão máxima não deve ser superior a 7 mm.

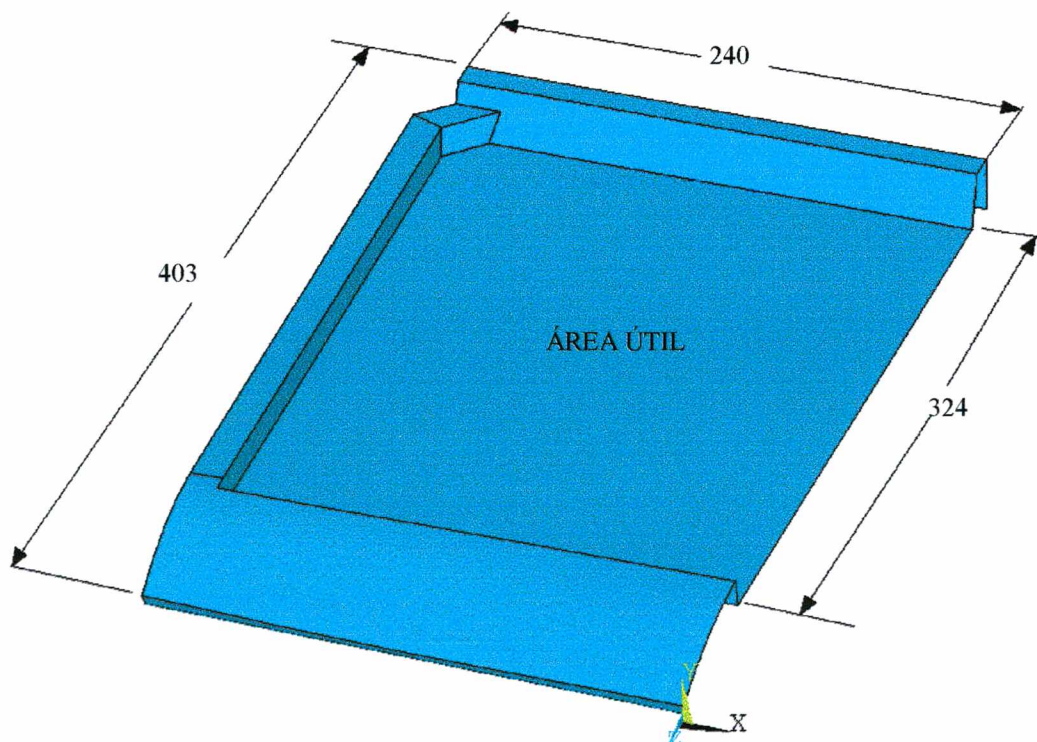


Figura 5.18 – Modelo geométrico da prateleira de um refrigerador [mm].

O modelo geométrico é mostrado na figura 5.18, com as características dimensionais básicas, nota-se que em função da simetria foi modelado somente metade da peça. Na tabela 5.11, estão as características do componente utilizadas para análise por elementos finitos.

Tabela 5.11 - Características do componente - caso prático.

Propriedades materiais	$E = 3500$ [MPa]
	$\nu = 0,35$
Propriedades geométricas	$x_1 = 4,0$ [mm]
	$x_2 = 4,0$ [mm]
	área útil = 155520 [mm ²]
carga	$F = 120$ N

Para o modelo de elementos finitos, ilustrado na figura 5.19, foi utilizado um elemento de casca triangular de 6 nós do programa ANSYS (SHELL93), e a geometria foi discretizada com 803 elementos, e 1684 nós.

No problema de otimização, foi tomado como vetor variável de projeto (x_i) as espessuras da área útil (x_1), onde é aplicado o carregamento, e a espessura da borda (x_2), no contorno da prateleira, obedecendo aos limites máximo e mínimo estabelecidos. O objetivo deste problema é reduzir o volume (massa) , obedecendo as restrições de projeto, tais como restrição de tensão e de deflexão.

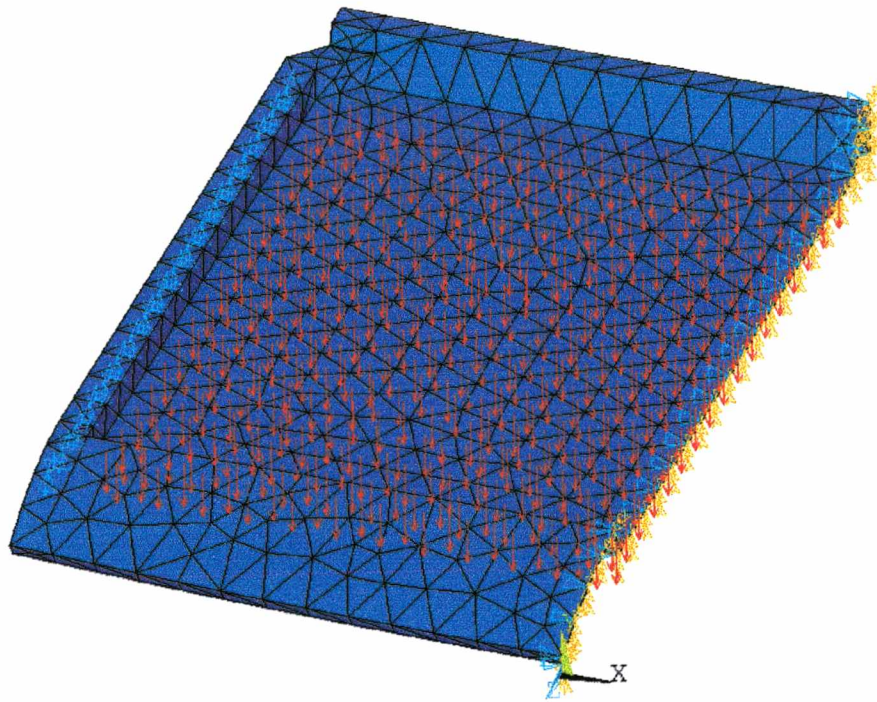


Figura 5.19 – Modelo de elementos finitos com as condições de contorno.

O problema de otimização é matematicamente descrito da seguinte forma:

Encontrar um x_i , vetor variável de projeto, que otimiza

$$Volume = \sum (Áreas.Espessuras)$$

Sujeito à

$$restrições\ laterais: x_{inf} \leq x_i \leq x_{sup} \quad (5.22)$$

$$restrições\ de\ tensão: -\sigma_{inf} \leq \sigma_i \leq \sigma_{sup} \quad (5.23)$$

$$restrições\ de\ deslocamento: u_{inf} \leq u_i \leq u_{sup} \quad (5.24)$$

Os limites para as restrições são indicados na tabela 5.12.

Tabela 5.12 - Limites para as restrições - caso prático.

Variável de projeto	$x_{inf} = 2,0$ mm
	$x_{sup} = 4,0$ mm
Limites de tensão	$\sigma_{inf.} = -35$ MPa
	$\sigma_{sup.} = 35$ MPa
Limites de desloc.	$u_{inf.} = 7,00$ mm
	$u_{sup.} = 7,00$ mm

As figuras 5.20 (a) e (b) mostram os resultados da análise nas condições iniciais do projeto. É importante salientar que a seleção da espessura inicial foi feita de modo intuitivo sem análise prévia.

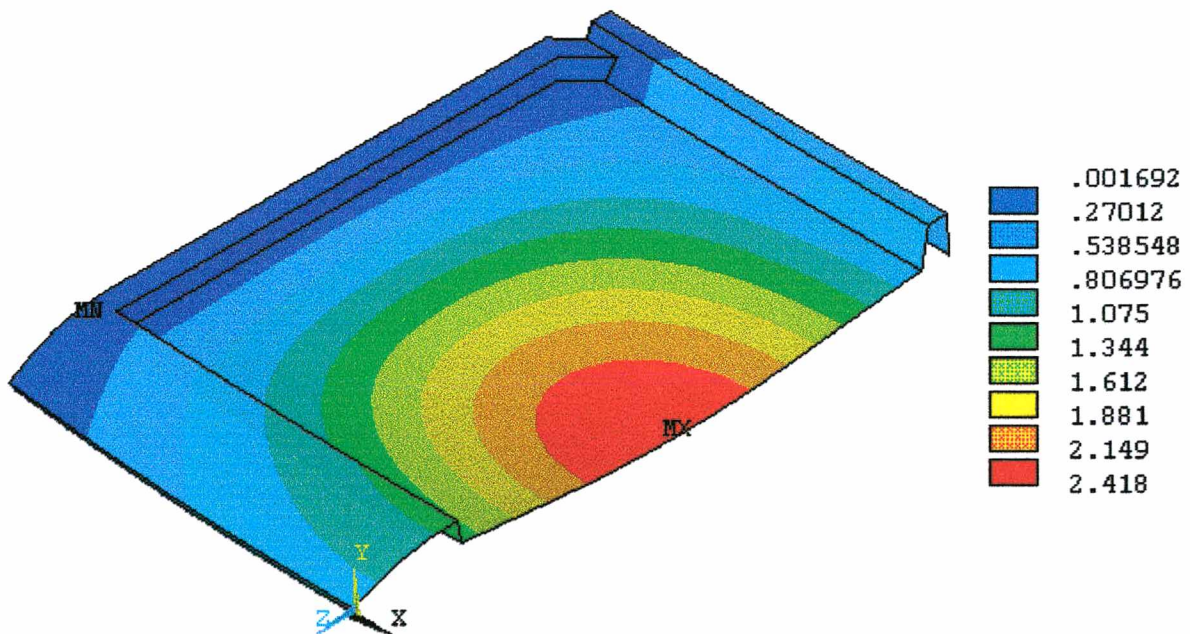


Figura 5.20 (a) – Resultado de deflexão [mm] - configuração inicial.

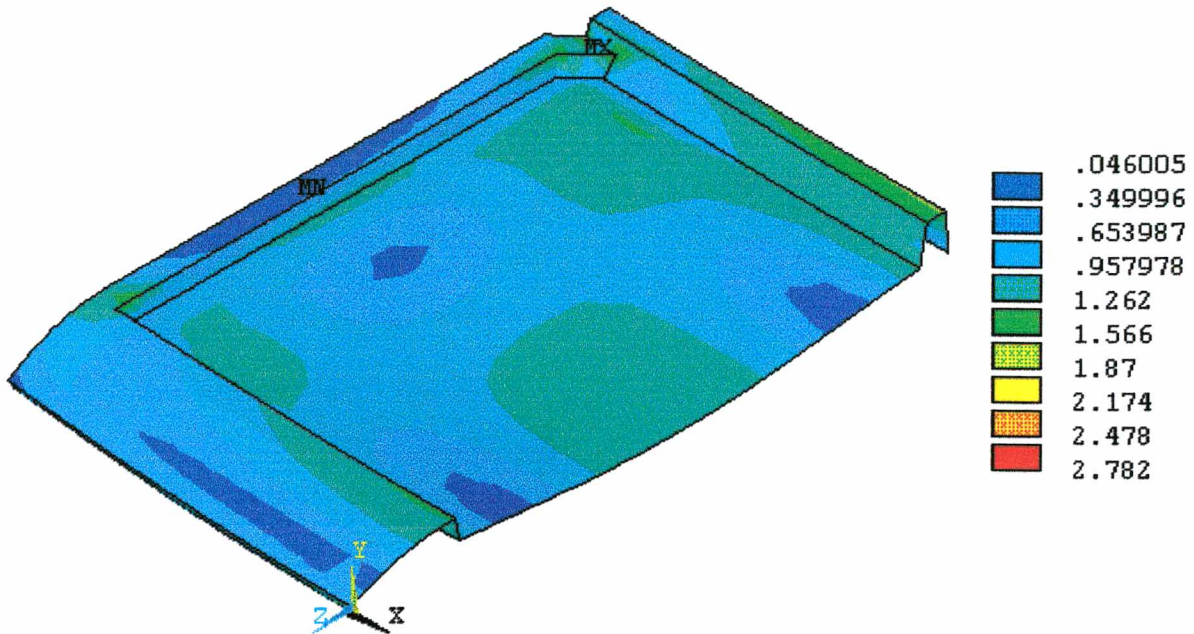


Figura 5.20 (b)– Resultado de tensões de von Mises [MPa], top - configuração inicial.

Submetendo a peça ao processo de otimização encontrou-se os resultados mostrados na tabela 5.13, e que atendem os requisitos do projeto e de manufatura da peça.

Tabela 5.13 - Resultado comparativo entre os métodos de otimização - caso prático.

	valores iniciais	Proc. Proposto	M.P.O	M.S.A
x_1 [mm]	4,0000E+00	2,6589E+00	2,6446E+00	2,6561E+00
x_2 [mm]	4,0000E+00	2,0000E+00	2,0000E+00	2,0041E+00
volume [mm ³]	1,073706E+06	6,3927E+05	6,3704E+05	6,3929E+05
massa [kg]	1,12202277	6,6804E-01	6,6571E+01	6,6806E+01
deflexão [mm]	2,4180E+00	6,9960E+00	7,0602E+00	6,9997E+00
Razão	(método/restricção)	9,9943E-01	1,0086E+00	9,9996E-01
	7,0000E+00			

M.P.O. - Método de Primeira Ordem - ANSYS

M.S.A. - Método Sub-problema Aproximado - ANSYS

Para ilustrar e comparar os resultados iniciais com o da peça otimizada, é mostrado nas figuras 5.21 (a) e (b) os resultados de deflexão e tensão da prateleira, respectivamente.

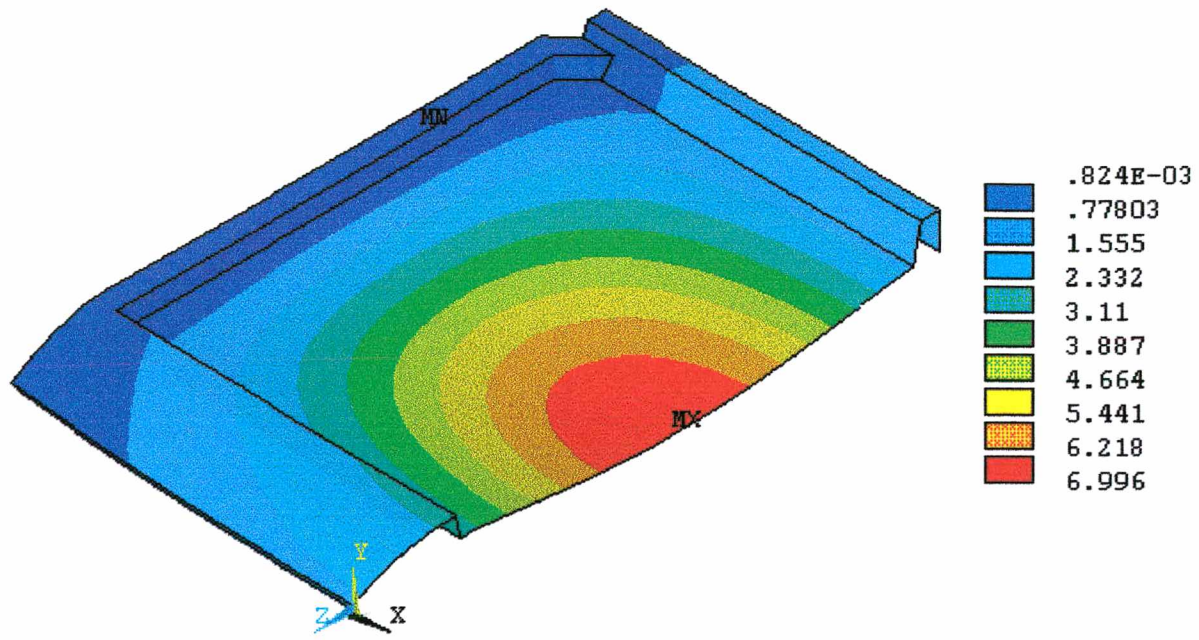


Figura 5.21 (a) – Resultado de deflexão [mm] - peça otimizada.

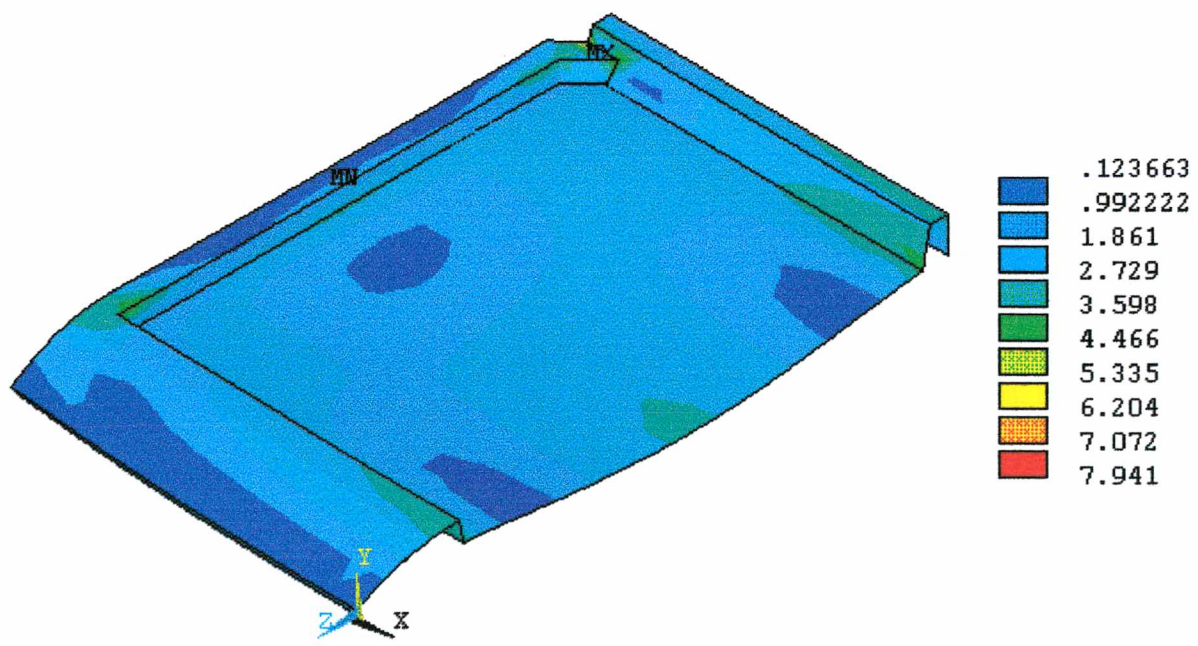


Figura 5.21 (b) – Resultado de tensão de von Mises [MPa], top - peça otimizada.

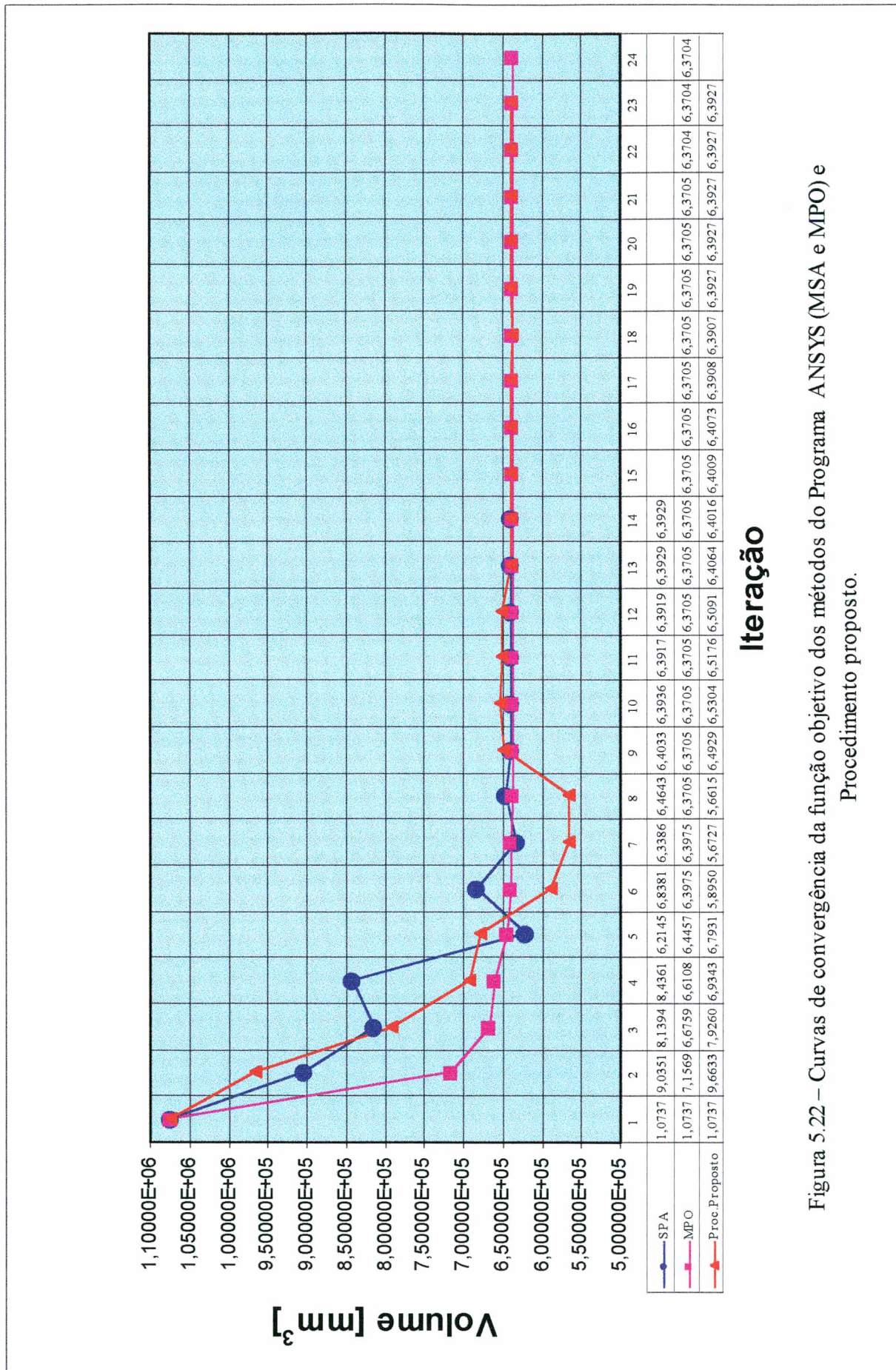


Figura 5.22 – Curvas de convergência da função objetivo dos métodos do Programa ANSYS (MSA e MPO) e Procedimento proposto.

Para análise dos resultados, verifica-se na tabela 5.13 os valores obtidos em cada método, onde se pode notar a estreita aproximação dos resultados. Se for comparado o valor da deflexão obtido em cada um dos métodos com o valor imposto como restrição de deslocamento, observa-se que o valor da razão quanto mais próximo estiver de um (1,0) melhor será a solução do problema. Portanto, o método M.S.A. do programa ANSYS apresenta valores bem precisos, assim como o Procedimento Proposto, que utiliza diferenças finitas para computar o gradiente da função, externamente ao programa comercial ANSYS. Quanto ao Método M.P.O. do ANSYS, embora, os resultados não tenham apresentado uma precisão tão boa quanto aos outros métodos, (M.S.A. e Procedimento Proposto), os mesmos também podem atender os aspectos do projeto, considerando a abordagem prática da indústria, embora exista a violação da restrição de deslocamento. Este exemplo viabiliza o uso do procedimento de otimização como ferramenta auxílio em um ambiente de projeto, visto as comparações feitas com os métodos do programa ANSYS. Na figura 5.22 nota-se a evolução da função objetivo através da curva de convergência de cada um dos métodos analisados. Quanto ao comportamento das curvas é importante salientar a queda da curva do Procedimento Proposto à um patamar abaixo da linha de convergência, caracterizando a violação da restrição de deslocamento, notado durante o processo iterativo.

Para dar uma ilustração do aspecto econômico, que é o foco atual no contexto industrial, esta redução de espessura na peça significa uma economia em matéria prima (Poliestireno Cristal) de 0,45398 kg, ou seja reduzir a massa de 1,12202 kg para 0,66804 kg, representando aproximadamente 40,46 % de redução em matéria prima. Se traduzir estes valores para unidade monetária tem-se para a peça com 4 mm de espessura (constante) um custo de R\$ 1,822, já que o custo por quilograma deste material é da ordem de R\$ 1,624 (07/2000). Na configuração otimizada o custo de matéria prima para a peça fica na ordem de R\$ 1,085, o que representa um ganho de R\$ 0,737/peça. Para contabilizar os ganhos reais no ano é interessante considerar o volume de produção anual dos produtos que utilizam este componente. Sendo assim, para ter-se um dado mais concreto utilizou-se valores de produção do ano de 1999 de refrigeradores, sendo contabilizado 1.015.000 produtos. Este volume representa então uma economia de matéria prima na ordem de 460.789,70 kg/ano e em dinheiro representa um ganho de R\$ 748.322,47/ano.

Capítulo 6

Conclusão e Sugestões

Este trabalho teve como objetivo desenvolver um procedimento para otimização estrutural integrando rotinas de otimização e gerenciamento de dados com um programa comercial de elementos finitos, gerando um procedimento genérico, flexível e de baixo custo. Com foi verificado, a idéia de se ter um procedimento deste tipo permite ao usuário resolver problemas de otimização de projeto, envolvendo minimização de forma, dimensional e identificação de propriedade de materiais. O procedimento desenvolvido permite também trabalhar com outras equações de estado, as quais podem ser solucionadas com o uso de diferentes métodos numéricos. Uma outra vantagem que se deve destacar é a flexibilidade em relação a programas comerciais. Esta flexibilidade permite trabalhar com uma gama bastante abrangente de problemas de engenharia, além de contar com a experiência do engenheiro ou técnico em relação ao programa comercial de elementos finitos e nos métodos de otimização disponíveis.

Um dos pontos chaves do sistema é a utilização de um programa comercial de análise, no qual a interação é feita por intermédio de arquivos de entrada (pré-processamento) e saída (pós-processamento). Este fato é importante salientar, já que o procedimento não se limita a um único programa comercial, pois através de simples alterações das rotinas de integração pode-se interagir com outros programas comerciais de elementos finitos. A integração com outros programas depende da possibilidade do programa comercial permitir um acesso através de uma linha de comando com parâmetros de entrada e saída. Isto pode ser possível, já que o sistema desenvolvido trabalha com o executável do programa comercial. O acesso ao programa de elementos finitos em modo "batch" ocorre através de comandos específicos do FORTRAN 90 que permitem utilizar comandos do sistema operacional.

Uma desvantagem que foi percebida nesta metodologia é com relação ao tempo. Como se sabe, o processo de otimização não é um procedimento simples. Os processos são iterativos e demandam um tempo maior para obtenção dos resultados. Além disso, existem uma série de controles e ajustes que são necessários fazer para obter-se uma solução ótima confiável e isto é inerente a qualquer sistema de otimização. Neste trabalho a queda de performance é resultante do tempo de acesso ao disco rígido para leitura, concatenação e geração de arquivos intermediários essenciais para a criação do arquivo de entrada para o sistema de comunicação entre as rotinas dos métodos de programação matemática e o programa comercial considerado. Outro ponto que contribui para este fato é o número muito grande de variáveis de projeto, pois requer um maior número de chamadas do programa comercial e conseqüentemente, a utilização intensa de acesso a disco rígido, principalmente, no cálculo numérico das derivadas com o uso do método das diferenças finitas central. Portanto, este método se torna ineficiente se o número de variáveis de projeto é muito grande. Quanto ao método das diferenças finitas observou-se nos exemplos que a convergência comportou-se de forma suave, obtendo-se resultados suficientemente precisos, quando comparados com o processo de otimização do ANSYS. Isto assegura que o procedimento funciona e pode ser usado como uma ferramenta a mais para auxiliar o engenheiro no ambiente de projeto.

Este trabalho deixa uma lacuna, que é a integração com tecnologias CAD. É comum se ver cada vez mais os programas de CAD com pré-processadores e pós-processadores, porém restritos a determinados programas de elementos finitos, dentre os mais conhecidos pode-se citar o Pro Engineer, IDEAS, Catia, etc. Normalmente, estas tecnologias CAD possuem módulos de programação para que o usuário possa desenvolver alguns aplicativos, customizando a tecnologia, o que permite adequar e aumentar a agilidade do seu processo de projeto. Portanto, fazendo uso deste recurso, pode-se desenvolver uma interface para que o processo de otimização proposto funcione integrado a tecnologias CAD e CAE dentro de um mesmo ambiente. Um exemplo, que pode ser citado é o que já ocorre hoje entre Pro/E e ANSYS, Pro/E e Mechanical, Patran e Nastran. No entanto, com o procedimento proposto integrado com C.A.D. e C.A.E., teria-se maior flexibilidade com relação a escolha das rotinas de otimização a serem usadas.

Outra alternativa ainda a ser explorada é o da integração das rotinas de otimização dentro do próprio programa comercial, se este permitir. No caso do programa ANSYS isto é perfeitamente viável, mediante a criação de macros. Também como sugestão, agora com vista

para uma integração mais completa, é integrar a análise estrutural e a simulação do processo de injeção de peças plásticas dentro do mesmo procedimento. Isto poderia ser feito utilizando um programa comercial para simulação de injeção de termoplástico, por exemplo, o MoldFlow. Desta forma ter-se-ia a possibilidade de obter-se uma peça otimizada levando em consideração ambos os aspectos, o de projeto e o de manufatura.

Uma outra idéia, também bastante interessante, é conceber uma forma de disponibilizar a tecnologia para usuários menos experientes, não só em otimização, mas também no uso de programas comerciais de elementos finitos. Para isto pode ser feita uma interface mais amigável entre o usuário e o sistema proposto, tornando este procedimento semi-especialista, através do qual o usuário pode ser orientado para formular o problema de otimização e criar os arquivos necessários para a utilização do programa comercial de elementos finitos e o programa de otimização.

Outro ponto interessante é com relação a manutenção deste procedimento, pois se sabe que no meio científico as mutações são constantes. A todo instante novas técnicas e métodos matemáticos estão sendo implementados e aperfeiçoados. Neste sentido nada impede que se aproveite estes novos conceitos dentro deste procedimento, tornando-o muito mais robusto e flexível, e melhorando sua performance de modo geral.

Referência Bibliográfica

01. Adeli, H. - **Advances in Design Optimization**, Chapman & Hall 1994.
02. Adelman, H. M., Haftka, R. T. - **Sensitivity Analysis for Discrete Structural Systems**, AIAA J., vol. 24, pp. 823-831, 1986.
03. Alves, Marcelo K. - **Otimização**, Notas de aula 1997.
04. Ansys Manual - **Design Optimization Seminar**, ANSYS Inc., 1997.
05. Ansys Manual - **Programmer's Manual Rev. 5.3**, ANSYS Inc., 1995.
06. Ansys Manual - **Theoretical Manual Rev. 5.5**, ANSYS Inc., 1998.
07. Ansys Manual - **User's Manual Rev. 5.3**, ANSYS Inc., 1995.
08. Ansys Manual - **Verification Manual Rev. 5.5**, ANSYS Inc., 1998.
09. Ansys Manual - **Ansys/ProFEA User's Guide Rev. 5.5**, ANSYS Inc., 1998.
10. Ansys Manual - **APDL Programmer's Guide Rev. 5.4**, ANSYS Inc., 1997.
11. Arora, J. S. - **Introduction To Optimum Design**, McGraw-Hill, 1989.
12. Atatekin I. S. - **An Introduction to Parametric CAD**, Journal of Engineering Computing and Applications, vol. 4 nr -2, 1989; p. 53 - 58.
13. Atatekin I. S. - **A Review of Parametric CAD Software**, Journal of Engineering Computing and Applications, vol. 4 nr -2, 1989; p. 59 - 68.
14. Babuska, I. & Szabó, B. - **Finite Element Analysis**, John Wiley & Sons, Inc. 1991.

15. Bazaraa, M. S., Sherali, H. D., Shetty, C. M., - **Nonlinear Programming: Theory and Algorithms**, John Wiley & Sons, Inc. 1993.
16. Beck, A. T. e Rosa, E. da - **ACE-Pro - Manual para Utilização e Programação**. Grante – Grupo de Análise e Projeto Mecânico 1999, Curso de Pós Graduação em Engenharia Mecânica, UFSC.
17. Beck, A. T. - **Um Método para Análise de Confiabilidade para Elementos Finitos**. Dissertação de Mestrado 1999, Curso de Pós Graduação em Engenharia Mecânica, UFSC.
18. Bertsekas, D. P. - **Constrained Optimization and Lagrange Multiplier Methods**, Academic Press, San Diego, 1982.
19. Blakely, K. D. - **Getting the Best Design with FEA**, MACHINE DESIGN vol. 15 - 20 Julho 1989; p. 89 - 93.
20. Botikin, M. E., Lust, R. V., Yang, R. J., Katnik, R. B., Song, J. O. - **Structural Sizing Optimization Using an External Finite Element Program**, AIAA paper No. 87-0883, AIAA, New York, USA, 1987.
21. Brush, D.O. & Almroth, B.O. - **Buckling of Bars, Plates and Shells**, McGraw-Hill, 1975.
22. Calixto, A. K. – **Análise e Otimização de Cascas de Materiais Compostos Laminados**. Dissertação de Mestrado 1998, Curso de Pós Graduação em Engenharia Mecânica, UFSC.
23. Chandrupatla, T. R., Belegundu, A. D. - **Introduction to Finite Elements in Engineering**, Prentice Hall, Inc. 1991.
24. Chen, J. L., Ho, J. S. - **Direct Variational Method for Sizing Design Sensitivity Analysis of Beam and Frame Structures**, Computers & Structures, Vol. 42, pp. 503 - 509, 1992.
25. Cook, Robert D., Malkus, David S., Plesha, Michael E. – **Concepts and Applications of Finite Element Analysis**, John Wiley & Sons, Inc. 1989.

26. Dhatt, G., Touzot, G., - **The Finite Element Method Displayed**, John Wiley & Sons. 1984.
27. Feijóo, R. A., Taroco, E. - **Introducción al Analisis de Sensibilidad**, Laboratório Nacional de Computação Científica, Rio de Janeiro, 1993.
28. Fleury, C. - **Optimization Of Elastic Structures By Finite Elements**, Aerospace Laboratory Of The University Of Liege.1981.
29. Fleury, C., Braibant, V. - **Structural Optimization Involving Discrete Design Variables**, Aerospace Laboratory Of The University Of Liege.1982.
30. Gill, P. E., Murray, W., Wright, M. H. - **Practical Optimization**, Academic Press, New York 1981
31. Haftka, R.T., Kamat, M. P. - **Elements of Structural Optimization**, Martinus Nijhoff Publishers 1985.
32. Han, C. S., Wang, B. P. - **Optimum Design of Retangular Panels with a Hole**. Advances in Engineering Software, vol.16, pp. 127 - 134, 1993.
33. Haug, E. J., Choi, K.K., Komkov, V. - **Design Sensitivity Analysis of Structural Systems**, Academic Press, Orlando, USA, 1984.
34. Hugues, T.J.R. - **The Finite Element Method**, Prentice-Hall Inc.,1987.
35. Jahau Lewis Chen e Jan Sheiw Ho. - **Developing a Computer aided Optimum Structural Design System by Using Commercial Finite Element Programs**, Advances in Engineering Software vol.17, pp. 79 - 86, 1993.
36. Kumar, V., Lee, S. J., German, M. D. - **Finite Element Design Sensitivity Analysis and Integration with Numerical Optimization Techniques for Structural Design**, Computers & Structures, Vol. 32, No. , pp. 883 - 897, 1989.

37. Luemberger, D. G. - **Linear and Nonlinear Programming**, Addison -Wesley Publishing Company, 1992.
38. Lund, Erik - **Finite Element Based Design Sensitivity Analysis and Optimization**, Ph.D. dissertation, Institute of Mechanical Engineering, Aalborg University 1994.
39. Malvern, Lawrence E. - **Introduction to the Mechanics of a Continuous Medium**, Prentice Hall, Inc. 1969.
40. McCormick, Garth P. - **Nonlinear Programming: Theory, Algorithms, and Applications**, John Wiley & Sons, 1983.
41. Mahmoud, K. G., Engl, H. W. e Holzleitner, L. - **Optimum Structural Design Using MSC/NASTRAN and Sequential Quadratic Programming**. Computers & Structures vol. 52, No. 3, pp. 437-447, 1994.
42. Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. - **Numerical Recipes in Fortran - The Art of Scientific Computing**, second edition, Cambridge University Press – 1992.
43. Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. - **Numerical Recipes in Fortran 90 - The Art of Parallel Scientific Computing**, second edition, Vol. 2, Cambridge University Press – 1992.
44. Rao, Singiresu S. - **Engineering Optimization: Theory and Practice**, third edition John Wiley & Sons 1996.
45. Rosa, E da - **Modelamento de Sistemas Físicos**, Grante, Curso de Engenharia Mecânica - UFSC.
46. Rouse N. E. - **Design Optimization Goes Commercial**. MACHINE DESIGN vol. 25 - 23 Outubro 1986; p. 56 - 62.
47. Stoecker, W. F. - **Design of Thermal Systems**, third edition McGrawHill 1989.

48. Vanderplaats, G. N. - **Numerical Optimization Techniques For Engineering Design With Applications**, McGraw-Hill, 1984.