



Universidade Federal da Paraíba  
Centro de Informática  
Programa de Pós-graduação em Modelagem Matemática e Computacional

OTIMIZAÇÃO DE ESTRUTURAS ATRAVÉS DE UMA TÉCNICA DE  
PROGRAMAÇÃO SEMIDEFINIDA DE GRANDE PORTE

André Francisco Coêlho Castro

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional, PPGMMC, da Universidade Federal da Paraíba, como parte dos requisitos necessários à obtenção do título de Mestre em Modelagem Matemática e Computacional.

Orientador: José Miguel Aroztegui Massera

João Pessoa  
Fevereiro de 2018

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

C355o Castro, André Francisco Coêlho.

Otimização de estruturas através de uma técnica de programação semidefinida de grande porte / André Francisco Coêlho Castro. - João Pessoa, 2018.  
82 f. : il.

Orientação: José Miguel Aroztegui Massera.  
Dissertação (Mestrado) - UFPB/CI.

1. Modelagem Matemática e computacional. 2. Programação semidefinida. 3. Algoritmo de pontos interiores. 4. Otimização estrutural. I. Massera, José Miguel Aroztegui. II. Título.

UFPB/BC

OTIMIZAÇÃO DE ESTRUTURAS ATRAVÉS DE UMA TÉCNICA DE  
PROGRAMAÇÃO SEMIDEFINIDA DE GRANDE PORTE

André Francisco Coêlho Castro

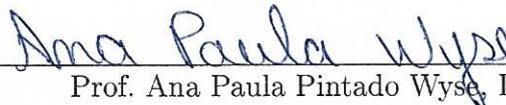
DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE  
PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL  
(PPGMMC) DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL  
DA PARAÍBA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A  
OBTENÇÃO DO GRAU DE MESTRE EM MODELAGEM MATEMÁTICA E  
COMPUTACIONAL.

Examinada por:



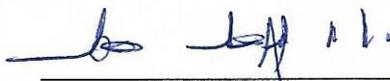
---

Prof. José Miguel Aroztegui Massera, D.Sc.



---

Prof. Ana Paula Pintado Wyse, D.Sc.



---

Prof. José Jefferson da Silva Nascimento, D.Sc.

JOÃO PESSOA, PB – BRASIL  
FEVEREIRO DE 2018

*À minha família, pessoas que  
muito amo.*

# Agradecimentos

Primeiramente, agradeço a Deus pela força, sabedoria e luz concedida em todos os momentos que preciso.

Aos meus pais, Francisco de Assis Castro e Maria de Jesus Coêlho Castro, e aos meus irmãos, Ancelmo e Aline, por estarem, incondicionalmente, ao meu lado em todos os momentos da minha vida e sempre torcendo pelas minhas conquistas.

A minha noiva, Mirlla, pelo apoio, atenção e pelos cuidados dedicados a mim do início ao fim desta jornada.

De forma especial, agradeço ao professor Józse Miguel Aroztegui Massera pelas orientações recebidas, disciplinas ministradas, conhecimentos transmitidos e por todo apoio, atenção e paciência ao longo da realização deste trabalho.

Ao professor Sérgio de Carvalho por ter, durante esses dois anos, dedicado seu tempo e experiência quase que de forma exclusiva para a turma 2016.1.

Ao Creyton pelos ensinamentos, vivências compartilhadas e por também ter aceitado esse desafio que em nenhum momento foi fácil, mas conseguimos!

A todos os amigos do PPGMMC que estiveram comigo e colaboraram com palavras de apoio e incentivo durante esta caminhada.

Aos professores José Jefferson e Ana Wyse por terem aceitado, prontamente, participar da banca examinadora na certeza de colaborar com o estudo.

A todos que, de forma direta ou indireta, me ajudaram e contribuíram com o estudo, meus sinceros agradecimentos.

Resumo da Dissertação apresentada ao PPGMMC/CI/UFPB como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## OTIMIZAÇÃO DE ESTRUTURAS ATRAVÉS DE UMA TÉCNICA DE PROGRAMAÇÃO SEMIDEFINIDA DE GRANDE PORTE

André Francisco Coêlho Castro

Fevereiro/2018

Orientador: José Miguel Aroztegui Massera

Programa: Modelagem Matemática e Computacional

As técnicas de programação semidefinida, que permitem lidar com problemas de otimização sujeitos a restrições matriciais, são muito eficientes quando se trata de aplicações em otimização estrutural. Utilizando-se destas técnicas, o presente trabalho apresenta um novo algoritmo numérico, o FDIPA-GSDP(3), pertencente a família FDIPA-SDP-NL, capaz de resolver problemas estruturais de grande porte. Seu diferencial está na formulação de um novo sistema de Newton, cuja função é encontrar uma direção que seja ao mesmo tempo de descida e viável, com dimensões bem reduzidas em relação ao das versões anteriores, o que facilita o seu armazenamento em memória e torna possível sua aplicação em problemas que requerem um grande número de elementos na discretização da estrutura. Com a finalidade de mostrar o desempenho deste algoritmo, apresentam-se resultados numéricos de aplicações das técnicas desenvolvidas em um problema clássico de otimização estrutural: a maximização da frequência natural de estruturas sujeito a restrições de volume e complacência.

Palavras-chave: programação semidefinida; algoritmo de pontos interiores por direções viáveis; maximização da frequência natural de estruturas mecânicas.

Resumo da Dissertação apresentada ao PPGMMC/CI/UFPB como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## OTIMIZAÇÃO DE ESTRUTURAS ATRAVÉS DE UMA TÉCNICA DE PROGRAMAÇÃO SEMIDEFINIDA DE GRANDE PORTE

André Francisco Coêlho Castro

Fevereiro/2018

Orientador: José Miguel Aroztegui Massera

Programa: Modelagem Matemática e Computacional

The semidefinite programming techniques, which allow to work with optimization problems subject to matrix constraints, are very efficient in structural optimization applications. Using these techniques, the present dissertation presents a new numerical algorithm, the FDIPA-GSDP(3), belonging to the family FDIPA-SDP-NL, capable to solve large optimization problems. Its differential is in the formulation of a new Newton system, whose function is to find a direction that is at the same time descent and viable, with dimensions quite reduced compared to previous versions, which facilitates their storage in memory and makes possible its application in problems that require a great number of elements in the discretization of the structure. In order to show the performance of this algorithm, numerical results of the applications developed in a classical problem of structural optimization are presented: maximization of the natural frequency of structures subject to volume and compliance constraints.

Keywords: semidefinite programming; feasible direction interior point algorithm; maximization of the natural frequency of mechanical structures.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Lista de Símbolos</b>	<b>xii</b>
<b>Lista de Abreviaturas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	3
1.2 Organização do Trabalho . . . . .	4
<b>2 Notações e Definições</b>	<b>5</b>
2.1 Funções . . . . .	5
2.2 Vetores e Matrizes . . . . .	7
2.3 Programação Matemática . . . . .	13
2.3.1 Problema de Otimização . . . . .	13
2.3.2 Problema de Programação Semidefinida Não Linear . . . . .	15
2.4 Discretização de um Sistema Estrutural Contínuo . . . . .	20
2.4.1 Elementos Finitos . . . . .	21
2.5 Formulação do Problema de Otimização Estrutural . . . . .	21
2.5.1 Formulação Clássica do Problema de Máxima Frequência . . . . .	22
2.5.2 Formulação SDP . . . . .	26
<b>3 Análise da Família de Algoritmos FDIPA</b>	<b>28</b>
3.1 A Matriz do Sistema de Newton nos Algoritmos FDIPA . . . . .	28
3.1.1 O Algoritmo FDIPA-SDP . . . . .	29
3.1.2 O Algoritmo FDIPA-GSDP(1) . . . . .	29
3.1.3 O Algoritmo FDIPA-GSDP(2) . . . . .	33
<b>4 FDIPA-GSDP(3): uma Nova Técnica de Programação Semidefinida para Problemas de Grande Porte</b>	<b>36</b>
4.1 Aperfeiçoamento do Algoritmo . . . . .	36

4.2	Testes Preliminares . . . . .	38
4.3	O Novo Algoritmo . . . . .	39
<b>5</b>	<b>Testes Numéricos</b>	<b>41</b>
5.1	Detalhes do Problema Semidefinido . . . . .	41
5.2	Dados de Controle do Algoritmo . . . . .	43
5.3	Escolha do Código . . . . .	45
5.4	Resultados dos Testes . . . . .	45
5.4.1	Exemplo 1 - Treliça Bidimensional de 26 Barras . . . . .	46
5.4.2	Exemplo 2 - Treliça Bidimensional de 534 Barras . . . . .	50
5.4.3	Exemplo 3 - Treliça Bidimensional de 1580 Barras . . . . .	52
<b>6</b>	<b>Considerações Finais</b>	<b>55</b>
6.1	Conclusão . . . . .	55
6.2	Sugestões para Trabalhos Futuros . . . . .	56
	<b>Referências Bibliográficas</b>	<b>57</b>
	<b>Apêndice A Informações Gerais sobre as Iterações Obtidas pelo</b>	
	<b>FDIPA-GSDP(3)</b>	<b>59</b>
A.1	Tabela de Iterações - Treliça de 26 Barras . . . . .	59
A.2	Tabela de Iterações - Treliça de 534 Barras . . . . .	60
A.3	Tabela de Iterações - Treliça de 1580 Barras . . . . .	63

# Lista de Figuras

2.1	Função Convexa . . . . .	6
2.2	Função Semi-Contínua Superiormente . . . . .	6
2.3	Treliça Bidimensional. . . . .	22
5.1	Geometria da treliça de 26 barras. . . . .	46
5.2	Convergência do FDIPA-GSDP(3) para a treliça de 26 barras. . . . .	48
5.3	Configurações inicial e final para a treliça de 26 barras. . . . .	49
5.4	Geometria da treliça de 534 barras. . . . .	50
5.5	Convergência do FDIPA-GSDP(3) para a treliça de 534 barras. . . . .	51
5.6	Configurações inicial e final para a treliça de 534 barras. . . . .	52
5.7	Geometria da treliça de 1580 barras. . . . .	52
5.8	Convergência do FDIPA-GSDP(3) para a treliça de 1580 barras. . . . .	53
5.9	Configurações inicial e final para a treliça de 1580 barras. . . . .	54

# Lista de Tabelas

5.1	Notação empregada para definir a dimensão do problema . . . . .	43
5.2	Notação empregada para definir o esforço computacional . . . . .	43
5.3	Notação empregada para definir parâmetros das estruturas mecânicas em todos os exemplos. . . . .	46
5.4	Dimensão do problema para a treliça de 26 barras. . . . .	47
5.5	Exemplo 1. Execução com o mesmo ponto inicial viável. . . . .	47
5.6	Exemplo 1. Execução com cada algoritmo buscando seu próprio ponto inicial viável. . . . .	47
5.7	Exemplo 1. Comprovação da sensibilidade do código aos parâmetros $\eta$ e $\xi$ . . . . .	47
5.8	Estrutura ótima obtida para a treliça de 26 barras. . . . .	49
5.9	Dimensão do problema para a treliça de 534 barras. . . . .	50
5.10	Exemplo 2. Execução com o mesmo ponto inicial viável. . . . .	50
5.11	Estrutura ótima obtida para a treliça de 534 barras . . . . .	51
5.12	Dimensão do problema para a treliça de 1580 barras. . . . .	53
5.13	Exemplo 3. Execução com o mesmo ponto inicial viável. . . . .	53
5.14	Estrutura ótima obtida para a treliça de 1580 barras . . . . .	54
A.1	Notação empregada para definir detalhes da execução do algoritmo. . . . .	59
A.2	Exemplo 1. Redução da função objetivo, histórico das condições de complementariedade e critérios de parada obtidos pelo FDIPA-GSDP(3). . . . .	60
A.3	Exemplo 2. Redução da função objetivo, histórico das condições de complementariedade e critérios de parada obtidos pelo FDIPA-GSDP(3). . . . .	63
A.4	Exemplo 3. Redução da função objetivo, histórico das condições de complementariedade e critérios de parada obtidos pelo FDIPA-GSDP(3). . . . .	69

# Lista de Símbolos

$A$	Matriz da restrição semidefinida, p. 15
$G$	Matriz diagonal cujas entradas são as restrições de desigualdade, $g_i$ , p. 30
$K$	Matriz de rigidez da estrutura, p. 22
$M$	Matriz de massa da estrutura, p. 22
$\Gamma$	Matriz diagonal cujas entradas são os multiplicadores de Lagrange das restrições de desigualdade, $\gamma$ , p. 31
$\Lambda$	Multiplicador de Lagrange associado à restrição matricial, p. 18
$\gamma$	Multiplicador de Lagrange associado às restrições de desigualdade, p. 30
$f$	Função objetivo, p. 13
$g$	Vetor das restrições de desigualdade, p. 13
$h$	Vetor das restrições de igualdade, p. 13
$m$	Número de restrições de desigualdade, p. 14
$n$	Número de variáveis de projeto, p. 14
$p$	Vetor de forças nodais, p. 22
$q$	Número de linhas ou colunas da matriz de restrições semidefinidas $A$ , p. 15
$u$	Vetor de deslocamentos, p. 22
$x$	Vetor das variáveis de projeto, p. 13

# Lista de Abreviaturas

- FDIPA-GSDP(3)      Feasible Direction Interior Point Algorithm for General Nonlinear Semidefinite Programming - Version Three, em português, Algoritmo de Pontos Interiores por Direções Viáveis para Programação Semidefinida Generalizada - Versão 3, p. 36
- FDIPA-SDP-NL      Feasible Direction Interior Point Algorithm for Nonlinear Semidefinite Programming, em português, Algoritmo de Pontos Interiores por Direções Viáveis para Programação Semidefinida Não Linear, p. 3

# Capítulo 1

## Introdução

Há tempos o homem vive uma incessante busca pelo ótimo, seja em atividades de natureza pessoal ou profissional. Durante a realização de tarefas cotidianas, até mesmo de maneira intuitiva, é fácil perceber a sua intenção em executá-las da melhor maneira possível, como por exemplo: ao traçar o caminho mais curto para um determinado local visando ganhar tempo; ao decidir a melhor disposição dos livros em uma prateleira, a fim de facilitar uma futura consulta; ao tirar os aparelhos elétricos da tomada quando não estão em uso, para economizar energia; ao procurar o melhor arranjo físico para os móveis em uma sala, com o objetivo de sobrar o maior espaço livre possível; entre outras.

O fato de ser dotado de inteligência permite ao ser humano avaliar ações desenvolvidas ou em desenvolvimento e, a partir disso, definir qual a mais acertada em meio às demais opções viáveis alternativas.

Encontrar a mais adequada entre as diversas soluções possíveis constitui um problema de otimização. Para resolvê-lo, é imprescindível a adoção de um critério de avaliação, objetivo ou subjetivo, através do qual seja possível afirmar que uma determinada solução é melhor que as outras. Tal critério é conhecido como função objetivo, que se pretende otimizar, minimizar ou maximizar, a depender do caso em questão. Por outro lado, a presunção de que nem toda solução é viável indica a existência de restrições a serem observadas. Esses dois elementos caracterizam um problema de otimização: otimizar uma função objetivo satisfazendo restrições existentes.

Devido ao grau de maturidade atingido nos últimos anos [1], atribuído a inúmeras pesquisas e descobertas neste campo de estudo, os problemas de otimização são aplicados nas mais diversas áreas, tais como: industrial (indústrias aeroespaciais, automotivas, químicas, alimentícias, elétricas, de construção e de fabricação), transporte, agrícola, planejamento da produção, empresarial e de aplicações financeiras.

E na engenharia não é diferente, a excelência também é o alvo. Nesta área, até bem pouco tempo, os projetos eram elaborados com uma metodologia convencional

que dependia essencialmente da intuição, experiência e habilidade do projetista, culminando em um processo de tentativa e erro, onde a melhoria e viabilidade do produto eram obtidas a um alto custo [2], fator este que vai de encontro à meta geral: desenvolver sistemas melhores com gastos menores.

Em uma era de grande concorrência, já não basta apenas projetar um sistema que desempenhe sua função corretamente, é essencial que se projete o melhor sistema. Aqui, o significado de *melhor* diz respeito a um sistema eficiente, versátil, distinto e com uma boa relação custo-benefício.

Nesse sentido, a demanda cada vez maior para reduzir os custos de produção, visando suportar a concorrência global, levou os engenheiros a buscarem métodos rigorosos de tomada de decisão para projetar e desenvolver produtos e sistemas de forma econômica e eficiente [3]. Esta procura por melhores soluções é um dos grandes objetivos da engenharia, que tem, progressivamente, investido nos algoritmos de otimização para que seja possível aplicar esta ferramenta em problemas sempre maiores e mais complexos.

Graças à recente revolução tecnológica hoje, os computadores podem executar algoritmos robustos, realizando cálculos complexos e processando grandes quantidades de dados rapidamente, permitindo que os processos de concepção e otimização em engenharia se beneficiassem grandemente deste enorme progresso.

As técnicas de otimização, também conhecidas como técnicas de Programação Matemática, aliadas ao grande poder de processamento dos computadores permitem que problemas de grande porte, muitas vezes intratáveis pela grande quantidade de variáveis e restrições envolvidas, pudessem ser projetados e implementados de forma mais eficaz. Isso é altamente desejável, pois os sistemas melhor projetados custam menos, possuem mais capacidade e são fáceis de manter e operar.

As primeiras utilizações das técnicas de Programação Matemática, no ambiente de Projeto Ótimo em Engenharia, empregaram algoritmos de Programação Linear (tanto a função objetivo quanto as restrições que a delimitam têm natureza linear), embora sempre tenha sido considerado que para tais aplicações os problemas são de Programação Não-Linear (pelo menos um dos dois elementos - função objetivo ou restrições - tem caráter não linear) [4].

Apesar de existirem diversas técnicas de Programação Não-Linear para variadas aplicações em Projeto Ótimo, não há consenso entre pesquisadores em adotar uma delas como sendo a melhor, pois além de cada algoritmo possuir suas vantagens e desvantagens e de não haver uma base unificada para fins de comparação, as pesquisas nesse campo da ciência avançam de forma contínua [5].

A escolha do algoritmo mais apropriado para um determinado problema requer conhecimento e experiência. Uma parte significativa dos problemas reais da engenharia, principalmente no que tange a busca pelo ótimo em projetos estruturais -

aqueles cuja meta é encontrar uma estrutura viável que minimize uma função custo (o peso total da estrutura, por exemplo) e verifique um conjunto de restrições mecânicas impostas, sejam elas estáticas ou dinâmicas -, são de grande porte, já que envolvem um vasto número de variáveis e restrições de projeto.

Uma técnica que tem se mostrado bastante eficiente na resolução deste tipo de problema é a Programação Semidefinida [6], que pode ser entendida, de forma sucinta, como sendo a matemática utilizada para modelar e estudar as propriedades de uma classe de problemas de otimização [7].

O fato de alguns métodos de pontos interiores terem sido generalizados para programação semidefinida despertou o interesse dos estudiosos da época pelo tema e, desde então, novos algoritmos de programação matemática têm sido propostos, principalmente a partir de Direções ou Arcos Viáveis. Algumas implementações destes algoritmos já estão disponíveis e seu uso encontra-se em constante aperfeiçoamento [8, 9]. Entretanto, novas aplicações exigem do algoritmo melhorias em termos de robustez, eficiência e confiabilidade.

Neste contexto, o presente trabalho explora técnicas numéricas de grande porte que auxiliem algoritmos de otimização a resolver problemas de engenharia mecânica. Em particular, propõe-se a implementação de um Algoritmo de Pontos Interiores por Direções Viáveis para Programação Semidefinida Não Linear (FDIPA-SDP-NL).

A proposta inicial deste algoritmo foi idealizada por Herskovits [10] e, mais tarde, aprimorada e generalizada em [9] para trabalhar com programação semidefinida. Pertencendo a uma família de métodos de pontos interiores que geram configurações sempre viáveis, esse tipo de algoritmo é muito apropriado para Projeto Ótimo em Engenharia onde essa característica é muitas vezes imprescindível.

Visando criar um ambiente de validação para o algoritmo implementado, deseja-se aplicá-lo na resolução de um problema de otimização clássico amplamente conhecido na literatura da engenharia mecânica: a maximização da frequência natural de estruturas, cujo cerne é a construção de corpos que, frente ações externas periódicas, não venham a ter oscilações com grandes amplitudes, evitando que sofram deformações que comprometam sua integridade devido ao fenômeno da ressonância mecânica.

## 1.1 Objetivos

Os objetivos deste trabalho são:

- Construir o modelo de otimização para o problema de maximização da frequência natural de estruturas lineares elásticas por meio de restrições semidefinidas nas matrizes de rigidez e de massa;

- Dissertar sobre algoritmos de pontos interiores para programação potencialmente não convexa, não linear e semidefinida, com particular ênfase na técnica FDIPA-SDP-NL;
- Programar o código de elementos finitos e as técnicas numéricas associadas ao FDIPA-SDP-NL;
- Obter os resultados numéricos para estruturas discretizadas com grande número de elementos.

## 1.2 Organização do Trabalho

Os próximos capítulos desta dissertação estão organizados da seguinte forma:

**Capítulo 2:** apresenta algumas notações e definições matemáticas fundamentais para modelar o problema da maximização da frequência natural de estruturas, através de técnicas de Programação Semidefinida Não Linear, bem como as noções gerais necessárias de alguns métodos essenciais para o desenvolvimento do algoritmo central desta pesquisa, tais como: método dos elementos finitos.

**Capítulo 3:** foca em uma análise sobre a matriz do sistema de Newton na família de algoritmos FDIPA, vista muitas vezes como o grande empecilho para a resolução de problemas estruturais de grande porte, pois, devido ao seu tamanho, dificulta ou até mesmo torna inviável seu armazenamento em memória.

**Capítulo 4:** descreve o algoritmo proposto nesta dissertação, abordando as modificações realizadas no FDIPA para contornar o problema do armazenamento da matriz de Newton do sistema e, assim, facilitar sua aplicação em problemas com uma grande quantidade de elementos finitos (seja em estruturas discretas ou discretizadas).

**Capítulo 5:** expõe os resultados dos testes numéricos, aplicados ao problema de otimização estrutural desenvolvido no Capítulo 2, para avaliar o comportamento do Novo Algoritmo.

**Capítulo 6:** finalmente, apresenta as conclusões e as sugestões para trabalhos futuros.

# Capítulo 2

## Notações e Definições

A formulação de um problema de otimização depende essencialmente de conceitos e técnicas matemáticas, principalmente na área de Álgebra Linear. Visando facilitar o entendimento dos capítulos posteriores desta dissertação, apresenta-se aqui algumas notações e definições que serão de grande valia.

### 2.1 Funções

**Definição 1.** (*Conjunto Convexo*). Seja  $\mathcal{C}$  um subconjunto de  $\mathbb{R}^n$ .  $\mathcal{C}$  é um conjunto convexo se, para qualquer par de pontos  $x, y \in \mathcal{C}$  e para qualquer  $\lambda \in [0, 1]$ , o ponto  $\lambda x + (1 - \lambda)y$  também pertence a  $\mathcal{C}$ .

**Definição 2.** (*Função Convexa*). Uma função  $f : \mathcal{C} \rightarrow \mathbb{R}$ , definida em um conjunto convexo  $\mathcal{C}$ , é convexa se, e somente se, para todo  $x, y \in \mathcal{C}$  e para  $\lambda \in [0, 1]$ , verifica-se que:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (2.1)$$

Se para todo  $\lambda \in (0, 1)$  e  $x \neq y$  é válido que:

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y), \quad (2.2)$$

diz-se que  $f$  é estritamente convexa.

A Figura 2.1 elucida este conceito exibindo o gráfico de uma função convexa.

**Definição 3.** (*Função Quase-Convexa*). Uma função  $f$  é quase-convexa se, e somente se, para qualquer  $x$  e  $y$  e para  $\lambda \in [0, 1]$ , a seguinte desigualdade é satisfeita:

$$f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\}. \quad (2.3)$$

**Definição 4.** (*Função Semi-Contínua Superiormente*). Uma função  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é semi-continua superiormente no ponto  $a \in \mathbb{R}^n$  quando, dado um  $c > f(a)$ , existe  $\delta > 0$  tal que  $x \in \mathbb{R}^n$ ,  $\|x - a\| < \delta$  implica  $f(x) < c$ .

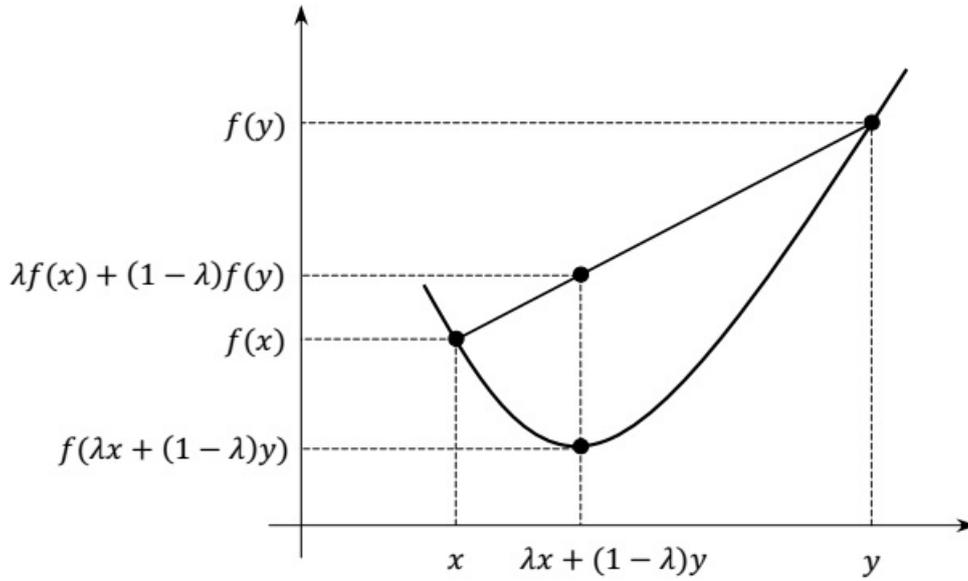


Figura 2.1: Função Convexa

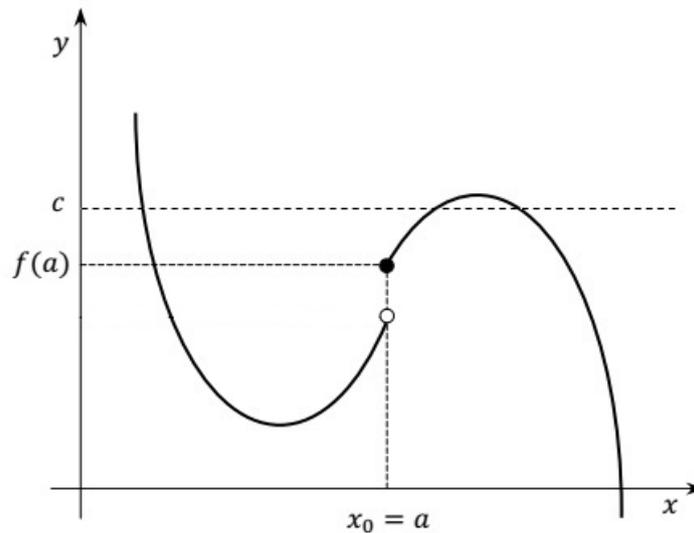


Figura 2.2: Função Semi-Contínua Superiormente

A Figura 2.2 elucida este conceito exibindo o gráfico de uma função semi-contínua superiormente.

**Definição 5.** (*Função Diferenciável*). Uma função  $f : \Phi \rightarrow \mathbb{R}$ , onde  $\Phi \subset \mathbb{R}^n$ , é diferenciável em  $x \in \Phi$  se, e somente se, para todo  $d \in \mathbb{R}^n$ :

$$f(x + d) = f(x) + \nabla f(x)^\top d + o(x, d), \quad (2.4)$$

onde:

$$\lim_{d \rightarrow 0} \frac{o(x, d)}{\|d\|} = 0. \quad (2.5)$$

O vetor  $\nabla f(x) \in \mathbb{R}^n$  é o gradiente de  $f(x)$ , vetor de derivadas parciais de  $f$  em

$x$ . A aproximação linear de  $f$  em  $x$ , denotada por  $AL$ , é:

$$AL(d) = f(x) + \nabla f(x)^\top d. \quad (2.6)$$

Similarmente, a função  $f$  é diferenciável em  $x$  até segunda ordem quando, para todo  $d$ :

$$f(x + d) = f(x) + \nabla f(x)^\top d + \frac{1}{2} d^\top \nabla^2 f(x) d + o_2(x, d), \quad (2.7)$$

onde:

$$\lim_{d \rightarrow 0} \frac{o_2(x, d)}{\|d\|^2} = 0. \quad (2.8)$$

e  $\nabla^2 f(x)$  é denominada Matriz Hessiana, sendo seus componentes correspondentes às derivadas parciais de segunda ordem de  $f$ . A derivada direcional de  $f(x)$  em torno de  $x$  na direção  $d \in \mathbb{R}^n$  é:

$$f'(x, d) = \lim_{\alpha \rightarrow 0} \frac{f(x + \alpha d) - f(x)}{\alpha}, \quad (2.9)$$

se o limite existir.

## 2.2 Vetores e Matrizes

**Definição 6.** (*Matriz Identidade*). Uma matriz identidade é uma matriz diagonal onde todos os seus elementos não nulos são iguais a um, isto é:

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \quad (2.10)$$

**Definição 7.** (*Matriz Inversa*). Se  $A$  for uma matriz quadrada e pudermos encontrar uma matriz  $B$  de mesmo tamanho tal que  $AB = BA = I$ , então  $A$  é inversível (ou não singular) e  $B$  é a inversa de  $A$ , denotada por  $A^{-1}$ . Se esta matriz  $B$  não puder ser encontrada diz-se que  $A$  não é inversível ou singular.

**Definição 8.** (*Matriz Simétrica*). Uma matriz quadrada  $A \in \mathbb{R}^{n \times n}$  é chamada simétrica se ela e sua transposta são iguais, isto é:

$$A^\top = A. \quad (2.11)$$

O espaço das matrizes simétricas com coeficientes reais e de tamanho  $n \times n$  é representado por:  $\mathbb{S}^n$ .

**Definição 9.** (*Matriz Esparsa*). Uma matriz é esparsa quando o número de elementos nulos é muito maior que o número de elementos não nulos.

**Definição 10.** (*Matriz Ortogonal*). Uma matriz ortogonal é uma matriz quadrada de componentes reais onde suas linha e colunas são vetores ortonormais (perpendiculares e de norma igual a um), o que equivale a  $A^\top = A^{-1}$  e leva a inferir que  $A^\top A = AA^\top = I$ .

**Definição 11.** (*Operadores vec e mat*). Para organizar em um vetor os componentes de uma matriz quadrada  $A$ , de ordem  $n$ , utiliza-se o operador  $\text{vec}(A) \in \mathbb{R}^{n^2}$ , que é definido como:

$$\text{vec}(A) = [a_{11}, \dots, a_{n1}, \dots, a_{12}, \dots, a_{n2}, \dots, a_{1n}, \dots, a_{nn}]^\top. \quad (2.12)$$

Denota-se por  $\text{mat}$  a operação inversa ao  $\text{vec}$ :

$$\text{mat}(\text{vec}(A)) = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & a_{22} & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix}. \quad (2.13)$$

Exemplo: Seja a matriz

$$A = \begin{bmatrix} 1 & -1 & 2 \\ -1 & 2 & 4 \\ 2 & 4 & 3 \end{bmatrix},$$

logo

$$\text{vec}(A) = [1, -1, 2, -1, 2, 4, 2, 4, 3]^\top.$$

**Definição 12.** (*Traço de uma Matriz*). O traço de uma matriz quadrada  $A$  de ordem  $n$  é definido pela soma dos elementos da sua diagonal principal:

$$\text{tr}(A) = \sum_{i=1}^n a_{ii}. \quad (2.14)$$

**Teorema 2.2.1.** (*Propriedades do traço de uma matriz*). Sejam  $A \in \mathbb{R}^{n \times n}$  e  $B \in \mathbb{R}^{n \times n}$ , os seguintes enunciados são equivalentes:

1.  $\text{tr}(A) = \sum_{i=1}^n \lambda_i(A)$ , onde  $\lambda_i(A)$  é o  $i$ -ésimo autovalor de  $A$ ;
2.  $\text{tr}(A) = \text{tr}(A^\top)$ ;
3.  $\text{tr}(AB) = \text{tr}(BA)$ ;
4.  $\text{tr}(AB^\top) = \text{tr}(A^\top B) = \text{vec}(A)^\top \text{vec}(B) = \sum_{i,j=1}^n a_{ij}b_{ij}$ .

*Demonstração: ver [11].*

**Definição 13.** (*Produto Interno de Matrizes*). Define-se o produto interno entre matrizes  $\mathbb{R}^{n \times n}$  a partir das propriedades do traço, de tal forma que:

$$A \cdot B = \langle A, B \rangle = \text{tr}(A^T B) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{ij}. \quad (2.15)$$

**Definição 14.** (*Norma de uma Matriz*). Define-se a norma Frobenius de uma matriz  $A \in \mathbb{R}^{m \times n}$ , denotada por  $\|A\|_F$ , como:

$$\|A\|_F = \left( \sum_{i=1}^m \sum_{j=1}^n (a_{ij})^2 \right)^{1/2}. \quad (2.16)$$

A norma Frobenius de uma matriz quadrada  $A \in \mathbb{R}^{n \times n}$  também pode ser definida através do produto interno:

$$\|A\|_F = \sqrt{\langle A, A \rangle} = \sqrt{\text{tr}(AA^T)}. \quad (2.17)$$

Por conveniência, a partir desta seção do trabalho, adotar-se-á a notação simplificada  $\|A\|$  para representar a norma de Frobenius de uma determinada matriz  $A$ .

**Teorema 2.2.2.** (*Propriedades da Norma de uma Matriz*). Sejam  $A$  e  $B \in \mathbb{R}^{n \times n}$ , a norma de uma matriz satisfaz as seguintes condições:

1.  $\|A\| > 0$  se  $A \neq O$ , onde  $\|O\| = 0$  ( $O$  é uma matriz com todos os seus elementos nulos);
2.  $\|cA\| = |c| \|A\|$ , para todo  $c \in \mathbb{R}$ ;
3.  $\|A + B\| \leq \|A\| + \|B\|$ ;
4.  $\|AB\| \leq \|A\| \|B\|$ .

*Demonstração: ver [12]*

**Definição 15.** (*Autovalores e Autovetores*). Um problema de autovalor-autovetor pode ser definido como: encontrar o par  $(\lambda, y)$  tal que

$$Ay = \lambda y, \quad (2.18)$$

onde  $\lambda$  é um autovalor e  $y \in \mathbb{R}^n$ , com  $y \neq 0$ , é um autovetor de  $A \in \mathbb{R}^{n \times n}$ , respectivamente.

Deseja-se encontrar uma solução não trivial, ou seja, o par  $(\lambda, y)$ , com  $y$  não nulo, que satisfaça a equação (2.18), que pode ser reescrita como:

$$(A - \lambda I)y = 0. \quad (2.19)$$

Para que isso seja possível, é necessário que  $(A - \lambda I)$  seja uma matriz singular, ou seja:

$$\det(A - \lambda I) = 0. \quad (2.20)$$

A equação (2.20) é conhecida como Equação Característica.

Se a matriz  $A$  é simétrica, o Teorema Espectral (enunciado abaixo) assegura a existência de  $n$  autovalores e  $n$  autovetores reais. Assim sendo:

- O menor autovalor de  $A$  denota-se por  $\lambda_{\min}(A)$ ,  $\lambda_1(A)$  ou  $\lambda_1$ ;
- O maior autovalor de  $A$  denota-se por  $\lambda_{\max}(A)$ ,  $\lambda_n(A)$  ou  $\lambda_n$ .

Para cada autovalor  $\lambda$ , seu autovetor associado,  $y$ , pode ser obtido pela equação (2.19).

**Definição 16.** (*Matriz Definida Positiva*). Uma matriz  $A \in \mathbb{R}^{n \times n}$  é definida positiva quando:

$$v^T A v > 0 \quad (2.21)$$

para qualquer  $v \neq 0 \in \mathbb{R}^n$ .

**Teorema 2.2.3.** (*Propriedades de uma Matriz Definida Positiva*). Seja  $A \in \mathbb{S}^n$ , os seguintes enunciados são equivalentes:

1.  $A \in \mathbb{S}_{++}^n$  ou  $A \succ 0$  ( $A$  é definida positiva);
2.  $\lambda_{\min}(A) > 0$ , onde  $\lambda_{\min}$  é o menor autovalor da matriz  $A$ ;

*Demonstração:* ver [13, 14].

**Definição 17.** (*Matriz Semidefinida Positiva*). Uma matriz  $A \in \mathbb{R}^{n \times n}$  é semidefinida positiva quando:

$$v^T A v \geq 0, \quad \forall v \in \mathbb{R}^n. \quad (2.22)$$

**Teorema 2.2.4.** (*Propriedades de uma Matriz Semidefinida Positiva*). Seja  $A \in \mathbb{S}^n$ , os seguintes enunciados são equivalentes:

1.  $A \in \mathbb{S}_+^n$  ou  $A \succeq 0$  ( $A$  é semidefinida positiva);
2.  $\lambda_{\min}(A) \geq 0$ , onde  $\lambda_{\min}$  é o menor autovalor da matriz  $A$ ;

*Demonstração:* ver [13, 14].

**Teorema 2.2.5.** (*Inversa de uma Matriz Definida Positiva*). Qualquer matriz definida positiva é inversível (seu determinante é diferente de zero e positivo) e sua inversa é definida positiva.

*Demonstração:* ver [13, 14].

**Teorema 2.2.6.** (*Multiplicação de uma Matriz de Posto  $k$  por uma Matriz Definida Positiva*). Se  $A \in \mathbb{R}^{n \times n}$  é positiva definida (não necessariamente simétrica) e  $X \in \mathbb{R}^{n \times k}$  tem posto  $k$ , então  $B = X^T A X \in \mathbb{R}^{k \times k}$  também é positiva definida.

*Demonstração:* ver [13, 14].

Por conveniência, adotar-se-á também neste trabalho, a partir desta seção, as seguintes notações:

- Matriz simétrica definida positiva:  $A \in \mathbb{S}_{++}^n$  ou  $A \succ 0$ .
- Matriz simétrica definida negativa:  $A \in \mathbb{S}_{--}^n$  ou  $A \prec 0$ , equivalente a  $-A \succ 0$ .
- Matriz simétrica semidefinida positiva:  $A \in \mathbb{S}_+^n$  ou  $A \succeq 0$ .
- Matriz simétrica semidefinida negativa:  $A \in \mathbb{S}_-^n$  ou  $A \preceq 0$ , equivalente a  $-A \succeq 0$ .

**Teorema 2.2.7.** (*Teorema Espectral para Matrizes Simétricas*). Seja  $A$  uma matriz real e simétrica. Então todos os autovalores de  $A$  são reais e existe uma base ortonormal de autovetores reais de  $A$ .

*Demonstração:* ver [14]

**Definição 18.** (*Produto de Kronecker*). Define-se o produto de Kronecker de duas matrizes  $A \in \mathbb{R}^{m \times n}$  e  $B \in \mathbb{R}^{r \times s}$  como sendo a matriz bloco de ordem  $mr \times ns$ :

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mr \times ns}, \quad (2.23)$$

onde cada bloco é formado pela multiplicação de um elemento de  $A$  pela matriz  $B$ , ou seja,  $[a_{ij}B]$ , com  $i = 1, \dots, m$  e  $j = 1, \dots, n$ .

Exemplo: Seja  $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  e  $B = \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix}$ , então:

$$A \otimes B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \times \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2 \times \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3 \times \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4 \times \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

**Teorema 2.2.8.** (*Propriedades do Produto de Kronecker*). Sejam  $A, B, C$  e  $D$  matrizes reais:

1.  $(C \otimes A)\text{vec}(D) = \text{vec}(ADC^\top)$ ;
2.  $(C \otimes A)^\top = C^\top \otimes A^\top$ ;
3.  $(C \otimes A)^{-1} = C^{-1} \otimes A^{-1}$ ;
4.  $(C \otimes A)(D \otimes B) = (CD) \otimes (AB)$ ;
5. se  $A \succ 0$  e  $B \succ 0$ , então  $(A \otimes B) \succ 0$ ;
6. se  $A \succ 0$  e  $B \prec 0$ , então  $(A \otimes B) \prec 0$ ;
7. se  $A \succeq 0$  e  $B \preceq 0$ , então  $(A \otimes B) \preceq 0$ .

*Demonstração:* ver [13]

**Definição 19.** (*Complemento de Schur*). Seja a matriz:

$$M = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}, \quad (2.24)$$

onde  $A$  é uma matriz definida positiva e  $C$  é uma matriz simétrica, então a matriz  $D = C - B^\top A^{-1}B$  é o complemento de Schur de  $A$  em  $M$ .

**Teorema 2.2.9.** (*Propriedades do Complemento de Schur*). A partir da definição do Complemento de Schur, os seguintes enunciados são equivalentes:

1.  $M$  é definida (ou semidefinida) positiva;
2.  $D$  é definida (ou semidefinida) positiva.

*Demonstração:* ver [15]

**Definição 20.** (*Condicionamento de Sistemas Lineares*). Diz-se que um sistema de equações lineares  $Ax = b$  é mal condicionado se pequenas perturbações em alguns de seus coeficientes produzem bruscas alterações em sua solução. Para detectar o mal condicionamento de um sistema linear, deve-se calcular o número de condição da matriz do sistema, definido por:

$$k(A) = \|A\| \|A^{-1}\|, \quad (2.25)$$

se  $k(A)$  for próximo de 1, o sistema é bem condicionado, caso contrário, é mal condicionado.

**Definição 21.** (*Transformação Linear*). Uma transformação (ou mapeamento)  $T$  é uma aplicação que satisfaz:

1.  $T(u + v) = T(u) + T(v)$ , para todo  $u, v$  no domínio de  $T$ ;
2.  $T(cu) = cT(u)$ , para todo  $u$  e todo escalar  $c$ .

## 2.3 Programação Matemática

### 2.3.1 Problema de Otimização

Para descrever conceitos e métodos de otimização, precisa-se de uma declaração matemática geral. Esse modelo matemático é definido como a minimização/maximização de uma função objetivo enquanto certas restrições são satisfeitas.

A ideia de otimizar (ou aperfeiçoar) um determinado objeto, implicitamente, pressupõe-se alguma liberdade em alterá-lo e as ferramentas para tais mudanças são os elementos que compõem um problema de otimização:

1. **variáveis de projeto:** um conjunto de valores que varia na busca pelo ótimo, ou seja, variáveis que poderão ser alteradas arbitrariamente ao longo do processo de otimização;
2. **função objetivo:** também chamada de função custo, representa quantitativamente a qualidade do objeto material a ser projetado e é escrito em função das variáveis de projeto (são usadas para medir o desempenho de um projeto);
3. **restrições:** conjunto de exigências que devem ser respeitadas e são definidas por propriedades físicas ou limitações de outra natureza.

Um valor particular das variáveis de projeto define uma configuração. Uma configuração onde todas as restrições são satisfeitas denomina-se configuração viável. O projeto é ótimo quando a função objetivo é a menor/maior (dependendo do caso) entre todas as configurações viáveis possíveis.

Com o intuito de exibir uma primeira notação para um problema de otimização, define-se  $x$  como sendo o vetor das variáveis de projeto,  $f(x)$  a função objetivo,  $g(x)$  e  $h(x)$  os vetores das restrições e  $\Omega$  o conjunto viável correspondente.

Assim, um problema geral de otimização restrito pode ser definido como:

$$\begin{cases} \text{minimizar} & f(x) \\ \text{sujeito a:} & x \in \Omega. \end{cases} \quad (2.26)$$

Em outros termos, pretende-se encontrar o valor da variável de projeto,  $x$ , que minimize a função objetivo,  $f(x)$ , sujeito a condição de que  $x$  pertença ao conjunto

viável  $\Omega$  que, em grande parte dos problemas de Projeto Ótimo em Engenharia, tem seguinte forma:

$$\Omega = \{x \in \mathbb{R}^n \mid g(x) \leq 0 \text{ e } h(x) = 0\}, \quad (2.27)$$

onde  $g(x)$  e  $h(x)$  são as restrições de desigualdade e de igualdade, respectivamente.

Sendo  $f(x)$ ,  $g(x)$  e  $h(x)$  funções não necessariamente lineares, (2.26) é um problema de Programação Não Linear e pode ser reescrito com mais detalhes da seguinte forma:

$$\begin{cases} \underset{x \in \mathbb{R}^n}{\text{minimizar}} & f(x) \\ \text{sujeito a:} & g(x) \leq 0 \\ & \text{e } h(x) = 0 \end{cases} \quad (2.28)$$

onde:

- $x \in \mathbb{R}^n$  é o vetor das variáveis de projeto;
- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função objetivo;
- $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  é a função das restrições de desigualdade, em que

$$g(x) \leq 0 \iff g_1(x) \leq 0, \dots, g_m(x) \leq 0$$

ou, em uma notação mais compacta

$$g_i(x) \leq 0, \text{ para } i = 1, \dots, m;$$

- $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$  é a função das restrições de igualdade, em que

$$h(x) = 0 \iff h_1(x) = 0, \dots, h_p(x) = 0$$

ou, em uma notação mais compacta

$$h_j(x) = 0, \text{ para } j = 1, \dots, p;$$

- $f$ ,  $g$  e  $h$  são funções diferenciáveis (de classe  $C^1$ ) e pelo menos uma delas é não linear.

Assim, deseja-se encontrar o valor de  $x$  que minimize a função  $f(x)$  respeitando as restrições de desigualdade,  $g(x) \leq 0$ , e de igualdade,  $h(x) = 0$ , que formam o conjunto viável  $\Omega$ .

Realizada esta visão geral preliminar sobre as notações de um problema de otimização, foca-se agora na apresentação do Problema de Programação Semidefinida

Não Linear, que será central neste trabalho e, por esta razão, exibido com mais detalhes.

### 2.3.2 Problema de Programação Semidefinida Não Linear

A programação semidefinida é uma área de estudo que integra vários problemas padrões de otimização e sua relevância prática é comprovada por ter seus conceitos aplicados, dentre outros, em problemas de engenharia (otimização estrutural e teoria do controle) e análise combinatória [9].

A criação de famílias de algoritmos para programação semidefinida tornou mais eficiente a busca por soluções ótimas em problemas de otimização estrutural de grande porte, desta maneira, é possível modelar um objeto (estrutura) de forma mais fiel à sua realidade e, conseqüentemente, a aproximação da solução encontrada também será melhor.

Um problema de Programação Semidefinida pode ser formulado da seguinte forma:

$$\begin{cases} \underset{x \in \mathbb{R}^n}{\text{minimizar}} & f(x) \\ \text{sujeito a:} & A(x) \preceq 0 \end{cases} \quad (2.29)$$

onde  $x \in \mathbb{R}^n$  é o vetor das variáveis de projeto,  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função objetivo,  $A : \mathbb{R}^n \rightarrow \mathbb{S}^q$  é a função das restrições matriciais e  $\Omega = \{x \in \mathbb{R}^n \mid A(x) \preceq 0\}$  é o conjunto das soluções viáveis (conjunto viável). Importante ressaltar que  $f$  e  $A$  são suaves e não necessariamente lineares e, em problemas de grande porte,  $n$  e  $q$  são valores muito maiores que um ( $n \gg 1$  e  $q \gg 1$ ).

Para resolver o problema (2.29), é necessário a realização de uma análise dos elementos que o compõe (função objetivo, variáveis de projeto e restrições) e esta tarefa envolve uma série de novos conceitos e notações, que serão apresentados a seguir.

Seja  $n$  o número de variáveis de projeto, representado por  $x = [x_1, \dots, x_n]^\top$ ,  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função objetivo e  $A : \mathbb{R}^n \rightarrow \mathbb{S}^q$  a função de restrição semidefinida, onde  $A(x) \in \mathbb{S}^q$ .

**Definição 22.** (*Gradientes*). *Apresenta-se aqui a notação do gradiente de funções empregadas neste trabalho:*

- Uma função com valores escalares  $f = f(x_1, x_2, \dots, x_n)$ , com  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , tem derivada conhecida como o gradiente de  $f$ :

$$\nabla f(x) = \left[ \frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right]^\top \in \mathbb{R}^n. \quad (2.30)$$

- Uma função de variável real com valores vetoriais,  $f(x) =$

$[f_1(x), f_2(x), \dots, f_m(x)]$ , com  $f : \mathbb{R} \rightarrow \mathbb{R}^m$ , tem derivada:

$$\frac{\partial f(x)}{\partial x} = \left[ \frac{\partial f_1(x)}{\partial x}, \dots, \frac{\partial f_m(x)}{\partial x} \right]^\top \in \mathbb{R}^m. \quad (2.31)$$

- Uma função vetorial de variável vetorial,  $f(x) = [f_1(x_1, x_2, \dots, x_n), \dots, f_m(x_1, x_2, \dots, x_n)]$ , com  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , tem derivada conhecida como o Jacobiano de  $f$ :

$$\frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \dots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}. \quad (2.32)$$

Em outras palavras, a função  $f$  é definida por um vetor de  $m$  componentes, sendo cada uma das funções  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ , com  $i = 1, \dots, m$ . As derivadas parciais dessas funções podem ser organizadas em uma matriz  $m \times n$ , que é a Matriz Jacobiana.

Outra representação bem usual nos algoritmos de otimização trabalha com a forma transposta, considerando  $x \in \mathbb{R}^n$  e a função  $f = [f_1(x), f_2(x), \dots, f_m(x)]^\top$ , ou seja,  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , as derivadas parciais de  $f_i$  em relação a  $x_i$  podem ser, também, agrupadas na matriz  $\nabla f(x) \in \mathbb{R}^{n \times m}$ , escrita como:

$$\nabla f(x) = [\nabla f_1(x), \dots, \nabla f_m(x)],$$

assim,  $\nabla f(x)^\top = \frac{\partial f(x)}{\partial x}$ .

- A segunda derivada de  $f$  em relação a  $x$ , também conhecida como a Hessiana de  $f$  em  $x$ , é denotada por:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (2.33)$$

onde as componentes  $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ , com  $i, j = 1, \dots, n$ , são as derivadas parciais de segunda ordem da função  $f$ .

- Uma função matricial  $A(x) \in \mathbb{S}^q$  tem suas derivadas parciais com respeito a

$x_i$ , com  $i = 1, \dots, n$ , denotada por  $A_i(x) = \frac{\partial A(x)}{\partial x_i} \in \mathbb{R}^{q \times q}$ :

$$A_i(x) = \frac{\partial A(x)}{\partial x_i} = \begin{bmatrix} \frac{\partial a_{11}(x)}{\partial x_i} & \dots & \frac{\partial a_{1q}(x)}{\partial x_i} \\ \vdots & \ddots & \vdots \\ \frac{\partial a_{q1}(x)}{\partial x_i} & \dots & \frac{\partial a_{qq}(x)}{\partial x_i} \end{bmatrix}. \quad (2.34)$$

O gradiente do mapeamento  $A$  em  $x$  denota-se por  $\nabla A(x) \in \mathbb{R}^{n \times q^2}$  e define-se como:

$$\nabla A(x) = \frac{\partial A(x)}{\partial x} = \begin{bmatrix} \text{vec}(A_1(x))^\top \\ \vdots \\ \text{vec}(A_n(x))^\top \end{bmatrix}. \quad (2.35)$$

**Definição 23.** (*Região Viável*). Define-se o conjunto dos pontos viáveis,  $\Omega$ , do problema (2.29) como:

$$\Omega = \{x \in \mathbb{R}^n \mid A(x) \preceq 0\}. \quad (2.36)$$

**Definição 24.** (*Ponto Interior*). Um ponto  $x$  é dito interior ao conjunto  $\mathbb{X}$ , se existe  $\kappa > 0$  tal que  $B(x, \kappa) \subset \mathbb{X}$ . Onde  $B(x, \kappa)$  denota uma bola aberta de centro em  $x$  e raio  $\kappa$ .

O conjunto formado por todos os pontos interiores de  $\mathbb{X}$  denomina-se interior de  $\mathbb{X}$  e denota-se por  $\text{int}(\mathbb{X})$ . Em programação semidefinida, o interior de  $\Omega$ , definido em (2.36), se assume, por hipótese, caracterizado por

$$\text{int}(\Omega) = \{x \in \mathbb{R}^n \mid A(x) \prec 0\}. \quad (2.37)$$

**Definição 25.** (*Mínimo Local*). O ponto  $x^* \in \Omega$  é um mínimo local de  $f$  se existe um  $\epsilon > 0$  tal que  $f(x^*) \leq f(x)$  para todo  $x \in \Omega \cap B(x^*, \epsilon)$ . Onde  $B(x^*, \epsilon)$  é uma bola aberta de centro  $x^*$  e raio  $\epsilon$  [16].

**Definição 26.** (*Mínimo Local Estrito*). O ponto  $x^* \in \Omega$  é um mínimo local estrito de  $f$  se existe um  $\epsilon > 0$  tal que  $f(x^*) < f(x)$  para todo  $x \in \Omega \cap B(x^*, \epsilon)$  [16].

**Definição 27.** (*Mínimo Global*). O ponto  $x^* \in \Omega$  é um mínimo global de  $f$  em  $\Omega$  se  $f(x) \geq f(x^*)$  para todo  $x \in \Omega$  [16].

Observa-se que um mínimo global também é um mínimo local e que os mínimos globais somente podem ser caracterizados em casos particulares, como em problemas de otimização onde as funções envolvidas são convexas (Programação Convexa), por exemplo.

**Definição 28.** (*Algoritmo Localmente Convergente*). Um algoritmo iterativo é localmente convergente se existe um  $\zeta$  positivo tal que para qualquer ponto inicial

$x^0 \in \Omega$  que verifica  $\|x^0 - x^*\| \leq \zeta$ , gera uma sequência de pontos que convergem à solução do problema,  $x^*$ .

**Definição 29.** (*Algoritmo Globalmente Convergente*). Um algoritmo iterativo é globalmente convergente se para qualquer ponto inicial  $x^0 \in \Omega$  gera uma sequência de pontos que convergem à solução do problema,  $x^*$ .

A preocupação principal das modernas técnicas de Programação Matemática está centrada na procura de métodos robustos e com convergência global. Na prática, os algoritmos com convergência local não são muito úteis uma vez que a vizinhança de convergência raramente é conhecida com antecedência [4].

**Definição 30.** (*Direção Viável*). Um vetor  $d \in \mathbb{R}^n$  é uma direção viável em  $x \in \Omega$  se existe um  $\tau \geq 0$  tal que  $(x + td) \in \Omega$  para todo  $t \in [0, \tau)$  [16].

Importante mencionar que a palavra direção, em programação matemática, é sinônimo de vetor. Em geral, os algoritmos globalmente convergentes definem a cada ponto uma direção de busca e procuram por um novo ponto nesta direção.

**Definição 31.** (*Direção de Descida*). Um vetor  $d \in \mathbb{R}^n$  é uma direção de descida para uma função  $f$  em  $x \in \mathbb{R}^n$  se existe um  $\psi > 0$  tal que  $f(x + td) < f(x)$  para todo  $t \in (0, \psi]$  [16].

**Definição 32.** (*Lagrangeano*). O Lagrangeano das funções envolvidas no problema (2.29), denotado por  $L : \mathbb{R}^n \times \mathbb{S}^q \rightarrow \mathbb{R}$ , é:

$$L(x, \Lambda) = f(x) + \langle \Lambda, A(x) \rangle, \quad (2.38)$$

onde  $\langle \cdot, \cdot \rangle$  denota o produto interno de matrizes e  $\Lambda \in \mathbb{S}^q$  é uma variável auxiliar, associada à restrição matricial, chamada de multiplicador de Lagrange.

Em otimização restrita, o Lagrangeano, ou função Lagrangeana, desempenha um papel equivalente à função objetivo em otimização sem restrições.

Usando o operador  $vec$ , mostrado na Definição 11, a notação  $a(x) = vec(A(x))$  e  $\lambda = vec(\Lambda)$ , ajuda a redefinir a função Lagrangeana como:

$$L(x, \lambda) = f(x) + \langle a^\top(x), \lambda \rangle. \quad (2.39)$$

O gradiente da função Lagrangeana em relação a  $x$ , associada ao problema com restrição matricial (2.29), é denotado como:

$$\nabla_x L(x, \lambda) = \nabla f(x) + \nabla A(x)\lambda. \quad (2.40)$$

**Definição 33.** (*Ponto Regular*). Seja  $\{b_1(x), \dots, b_r(x)\} \subset \mathbb{R}^{q^2}$  uma base ortonormal do  $\ker(A(x) \otimes I)$ . Um ponto  $x$  é um ponto regular do problema (2.29) se os vetores:

$$\nabla A(x)b_i(x) : i = 1, \dots, r \quad (2.41)$$

são linearmente independentes.

**Definição 34.** (*Condição de Mangasarian-Fromovitz*). A condição de Mangasarian-Fromovitz se verifica no ponto  $x$  se existe uma direção  $d \in \mathbb{R}^n$  tal que:

$$A(x) + \sum_{i=1}^n d_i A_i(x) \prec 0. \quad (2.42)$$

**Teorema 2.3.1.** (*Condições de Otimalidade de Primeira Ordem - KKT*). Se  $f$  e  $A$  são de classe  $C^1$ ,  $x$  é um mínimo local de (2.29) e verifica a condição de Mangasarian-Fromovitz, então existe uma matriz de multiplicadores de Lagrange  $\Lambda \in \mathbb{S}^q$  tal que:

$$\begin{aligned} \nabla f(x) + \nabla A(x)\lambda &= 0 \\ \Lambda A(x) &= 0 \\ A(x) &\preceq 0 \\ \Lambda &\succeq 0 \end{aligned} \quad (2.43)$$

onde  $\lambda = \text{vec}(\Lambda)$ .

As equações presentes em (2.43) são denominadas Condições de Otimalidade de Primeira Ordem de *Karush-Kuhn-Tucker* - KKT. A igualdade  $\Lambda A(x) = 0$  é chamada de condição de complementariedade.

Quando as hipótese do Teorema 2.3.1 e a condição de regularidade definida em (33) são verdadeiras, a matriz  $\Lambda$  (multiplicador de Lagrange) é única [17].

**Definição 35.** (*Ponto Estacionário*). Um ponto regular  $x$  é um ponto estacionário do problema (2.29), se existe um  $\Lambda \in \mathbb{S}^q$  tal que as seguintes condições são verificadas simultaneamente:

$$\begin{aligned} \nabla_x L(x, \Lambda) &= 0 \\ \Lambda A(x) &= 0 \\ A(x) &\preceq 0 \end{aligned} \quad (2.44)$$

onde  $\nabla_x L(x, \Lambda) = \nabla f(x) + \nabla A(x)\lambda$ , com  $\lambda = \text{vec}(\Lambda)$ .

**Definição 36.** (*Ponto de KKT*). Um ponto de KKT do problema (2.29) é um ponto estacionário tal que  $\Lambda \succeq 0$ .

## 2.4 Discretização de um Sistema Estrutural Contínuo

Define-se sistema estrutural como sendo uma associação de elementos estruturais formando um conjunto resistente, adequado a resistir a solicitações. Entende-se como esses elementos os corpos sólidos elástico-deformáveis com capacidade de receber e transmitir esforços de uma forma geral.

Um meio contínuo resulta da distribuição homogênea da matéria no volume. Fisicamente, a hipótese do contínuo consiste em abstrair-se da composição molecular da matéria que compõe o meio e da sua conseqüente descontinuidade.

Nesse contexto, pode-se dizer que um sistema estrutural é contínuo desde que o menor elemento estrutural retirado do sistema possua as mesmas propriedades físicas específicas e não haja descontinuidade no conjunto.

A dificuldade decorrente da admissão de tal hipótese reside na formulação das equações consistentes que governam o sistema estrutural, uma vez que essas são equações diferenciais ou integrais e devem satisfazer as suas condições de contorno. Cabe ressaltar que, em função desta continuidade, os sistemas estruturais contínuos possuem um número infinito de graus de liberdade (número de movimentos rígidos possíveis e independentes que um corpo pode executar).

Conforme [18], para contornar os problemas advindos da continuidade, empregam-se métodos que transformam um sistema estrutural contínuo em um sistema estrutural discreto, através da divisão do modelo geométrico dos elementos estruturais em pequenas regiões chamadas de elementos finitos, sendo estas interconectadas entre si para formar o conjunto estrutural.

Esse processo de discretização é a base do conceito dos métodos finitos, que permitem expressar a solução exata dos sistemas contínuos de uma forma aproximada, através da utilização de sistemas discretos que contêm um número finito de graus de liberdade. Em problemas estáticos, este procedimento resulta em um sistema de equações algébricas que pode ser facilmente resolvido com a ajuda de técnicas computacionais. Já em problemas dinâmicos, a discretização resulta em um sistema de equações diferenciais ordinárias do movimento, que requerem o uso de técnica computacional apropriada e métodos numéricos adequados de integração no tempo, além da discretização do tempo contínuo.

Com a obtenção desse sistema de equações, resolve-se o problema já citado, uma vez que tais sistemas facilitam a imposição das condições de contorno da estrutura. Por outro lado, a solução obtida passa a ser aproximada e para se chegar a uma resposta satisfatória, usualmente, deve-se escolher o tipo adequado de elemento finito, a quantidade e sua disposição geométrica.

### 2.4.1 Elementos Finitos

A definição do tipo apropriado de elemento finito é dada em função do sistema estrutural que se deseja analisar. Para isso, os elementos finitos são classificados em:

- elementos lineares: também conhecidos como elementos de barra, são elementos em que uma dimensão é muito maior do que as outras duas. Estes são subdivididos em elementos de pórtico planos e espaciais, de treliças planas e espaciais, de vigas e de grelhas;
- elementos laminares: são elementos em que uma dimensão é muito menor do que as outras duas e são subdivididos em elementos de placa, de chapa, de membrana e de casca;
- elementos sólidos: são elementos em que as três dimensões tem a mesma ordem de grandeza e permitem obter uma distribuição qualquer de tensões na estrutura.

Para que estes elementos finitos possam se relacionar uns com os outros, os mesmos devem ser interconectados por meio de um número discreto de pontos comuns distribuídos no contorno. Esses pontos comuns são denominados pontos nodais ou, simplesmente, nós (ao conjunto de elementos finitos e pontos nodais, dá-se, usualmente o nome de malha de elementos finitos). Com isso, o comportamento estrutural de cada elemento pode ser definido somente em função de variáveis nodais, tais como: coordenadas, deslocamentos e carregamentos, permitindo-se a montagem dos sistemas de equações já mencionadas anteriormente.

Uma vez definido o tipo e a disposição dos elementos finitos, deve-se estabelecer a quantidade a ser utilizada na discretização da estrutura. Essa quantidade vai afetar a precisão da solução e o tempo de processamento numérico, visto que o tamanho dos sistemas de equações a serem resolvidos depende do número de nós existentes. Cabe ressaltar que a quantidade de nós define o número de graus de liberdade da estrutura e, conseqüentemente, os seus modos de vibração em problemas dinâmicos.

## 2.5 Formulação do Problema de Otimização Estrutural

A engenharia, nas últimas décadas, vem levando em consideração vários problemas nos quais são analisadas as vibrações em estruturas quando submetidas a ação de carregamentos dinâmicos (ações de cargas que variam ao longo do tempo), principalmente para evitar que o corpo sofra deformações maiores que o esperado ou até

mesmo chegue a se romper, culminando no processo conhecido como ressonância mecânica.

Nestes casos, os impactos podem ser determinados através de uma análise das propriedades que tratam dos modos de vibração, associados às frequências naturais destas estruturas, e dos fatores de amortecimento.

Devido aos seus inúmeros desafios, abordar-se-á, a seguir, um problema de otimização clássico amplamente conhecido na literatura da engenharia mecânica [15, 19]: a maximização da frequência natural de estruturas com restrições de volume, equilíbrio e energia de deformação (complacência).

### 2.5.1 Formulação Clássica do Problema de Máxima Frequência

Problemas de geometria e carregamentos complexos, como frequentemente encontrados na indústria, são de difícil solução analítica, devido à impossibilidade, em geral, de formulação por expressões algébricas. Por esta razão, as equações de equilíbrio de problemas estruturais de grande porte são usualmente resolvidas de forma discreta. E, em casos como este, o Método dos Elementos Finitos é uma ferramenta numérica amplamente utilizada, dispondo dos conceitos de discretização do contínuo e de matriz de interpolação para fornecer os deslocamentos em um determinado ponto, no interior do elemento, em função de seus deslocamentos nodais.

Para a formulação do problema que será resolvido nesta dissertação, considere-se uma estrutura de treliças - conjunto de barras interconectadas nas extremidades - elástica, linear e de dimensão finita. Neste tipo de estrutura a discretização do domínio é imediata, pois cada barra representa um elemento finito, que, por sua vez, é composto por dois nós (extremos) e cada nó possui dois graus de liberdade (treliça 2D, caso abordado neste trabalho), como mostra a Figura (2.3).

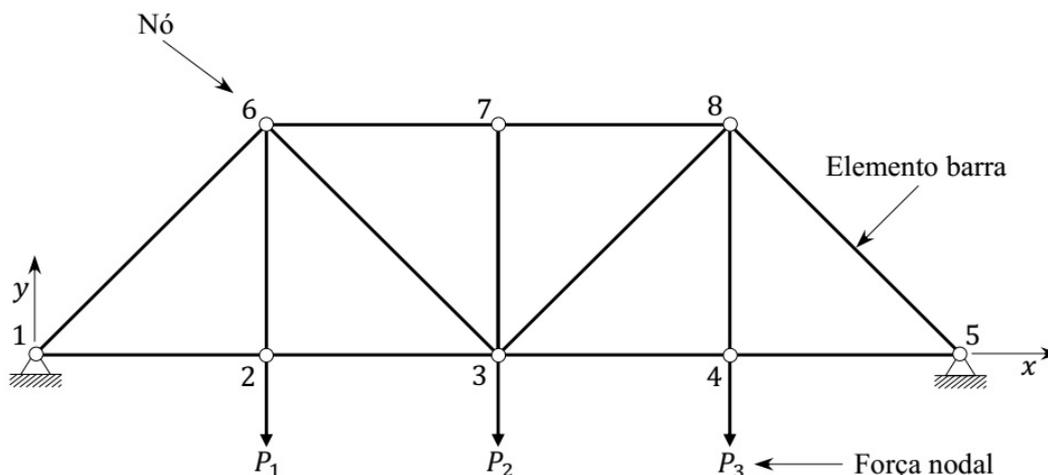


Figura 2.3: Treliça Bidimensional.

Determinada a estrutura a ser utilizada, o objetivo é buscar uma configuração para esta treliça onde se obtenha um desempenho ótimo segundo alguns parâmetros pré-definidos. Assim, ao formular um problema de otimização estrutural é indispensável determinar quais são as variáveis de projeto, a função objetivo e as restrições.

Para tanto, denota-se por  $n_{elem}$  o número de elementos da discretização (barras da treliça),  $n_{gdl}$  o número de graus de liberdade (gdl) da estrutura, correspondente a soma dos graus de liberdade de cada nó,  $x \in \mathbb{R}^{n_{elem}}$  o vetor das variáveis de projeto,  $K(x) \in \mathbb{S}^{n_{gdl}}$  a matriz de rigidez,  $M(x) \in \mathbb{S}^{n_{gdl}}$  a matriz de massa,  $p \in \mathbb{R}^{n_{gdl}}$  o vetor de forças nodais e  $u \in \mathbb{R}^{n_{gdl}}$  o vetor de deslocamentos nodais.

Define-se equação de equilíbrio como sendo o seguinte sistema linear:

$$K(x)u = p \quad (2.45)$$

Quando algum nó é submetido a uma força, o conjunto de barras é deformado (tração ou compressão) e os nós são deslocados. Assim, o vetor de deslocamento  $u$  pode ser obtido a partir da resolução do sistema linear dado pela equação (2.45).

Assume-se que as matrizes de rigidez e massa são funções lineares das variáveis de projeto. Desta forma, pode-se defini-las como:

$$K(x) = \sum_{i=1}^{n_{elem}} x_i K_i; \quad (2.46)$$

$$M(x) = \sum_{i=1}^{n_{elem}} x_i M_i, \quad (2.47)$$

onde  $K_i \in \mathbb{S}^{n_{gdl}}$  e  $M_i \in \mathbb{S}^{n_{gdl}}$  são matrizes constantes semidefinidas positivas, sendo  $x_i K_i$  e  $x_i M_i$  as matrizes de rigidez e massa dos elementos. Assume-se que  $x_i \geq 0$ , para  $i = 1, \dots, n_{elem}$ , o que implica que  $K(x) \succeq 0$  e  $M(x) \succeq 0$ .

Como as matrizes de rigidez e massa reduzidas  $K_i$  e  $M_i$  são constantes, então pode-se calcular suas derivadas em relação a  $x_i$  como:

$$\frac{\partial K(x)}{\partial x_i} = K_i; \quad (2.48)$$

$$\frac{\partial M(x)}{\partial x_i} = M_i. \quad (2.49)$$

### Frequência Natural de uma Estrutura: um Problema de Autovalores

A equação diferencial do movimento de uma estrutura para vibrações livres é:

$$M(x)\ddot{u} + K(x)u = 0. \quad (2.50)$$

A equação (2.50) pode significar, por exemplo, o movimento dos nós de uma

estrutura após uma “pancada”, ou seja, uma força  $p$  que se aplica por um instante de tempo e cessa imediatamente. Neste movimento não se consideram forças de atrito.

Assume-se que a solução da equação (2.50) tem a forma:

$$u = \Phi e^{j\omega t}. \quad (2.51)$$

Substituindo a equação (2.51) em (2.50), tem-se:

$$-w^2 M(x) \Phi e^{j\omega t} + K(x) \Phi e^{j\omega t} = 0. \quad (2.52)$$

A equação (2.52) pode ser escrita como:

$$[-w^2 M(x) \Phi + K(x) \Phi] e^{j\omega t} = 0. \quad (2.53)$$

Como  $e^{j\omega t} \neq 0$ , então tem-se o seguinte problema de autovalores:

$$K(x) \Phi = \lambda M(x) \Phi, \quad (2.54)$$

onde os autovalores  $\lambda$  são o quadrado das frequências naturais do sistema, isto é,  $\lambda = w^2$  e os autovetores associados,  $\Phi$ , representam os modos de vibração da estrutura.

Para garantir a existência de uma solução não trivial para a equação (2.54), verifica-se que:

$$\det[K(x) - \lambda M(x)] = 0. \quad (2.55)$$

O desejável em uma estrutura sob carregamentos dinâmicos é que todas as suas frequências naturais sejam bem maiores do que a máxima frequência possível dos carregamentos. Para garantir isto, precisa-se impor apenas que a menor das frequências naturais da estrutura seja maior do que um valor previamente conhecido (a maior frequência possível dos carregamentos multiplicada por um coeficiente de segurança, por exemplo). Com isto, impede-se que a estrutura falhe sob carregamentos dinâmicos.

Por exemplo, se o motor de um carro vibra a uma frequência  $\lambda_f$  e deseja-se projetar uma estrutura (com distribuição de volume  $x$ ) que sustente o motor, a aplicação da restrição de frequências naturais abaixo é bastante desejável:

$$\lambda_{min}(x) \geq \bar{\lambda}. \quad (2.56)$$

onde a constante  $\bar{\lambda}$  é escolhida de modo que  $\bar{\lambda} = \lambda_f \times fs$ , sendo  $fs$  um determinado fator de segurança. Este tipo de restrições previne que a estrutura falhe, pois se

a frequência fundamental da estrutura for próxima a  $\lambda_f$ , a estrutura pode entrar em ressonância: os deslocamentos podem aumentar até que a estrutura sofra uma fratura (quebra).

### Definição e Propriedades da Função $\lambda_{min}(x)$

Define-se a função  $\lambda_{min}(x)$  como o menor autovalor do problema (2.54), representado da seguinte forma:

$$\lambda_{min}(x) = \min\{\lambda : \exists \Phi \neq 0 \in \mathbb{R}^{n_{gdl}} : K(x)\Phi = \lambda M(x)\Phi\} \quad (2.57)$$

Seja  $\chi = \{x \in \mathbb{R}^{n_{elem}} \mid x \geq 0 \text{ e } x \neq 0\}$ , define-se as seguintes propriedades de  $\lambda_{min}(x)$  [15]:

1.  $\lambda_{min}(\cdot)$  é finito e não negativo em  $\chi$ ;
2. para todo  $x \in \chi$ ,

$$\lambda_{min}(x) = \sup\{\lambda \mid K(x) - \lambda M(x) \succeq 0\}; \quad (2.58)$$

3. para todo  $x \in \chi$ ,

$$\lambda_{min}(x) = \inf_{u: M(x)u \neq 0} \frac{u^\top K(x)u}{u^\top M(x)u}; \quad (2.59)$$

4.  $\lambda_{min}(\cdot)$  é semi-contínua superiormente em  $\chi$ ;
5.  $-\lambda_{min}(\cdot)$  é quase convexa em  $\chi$ .

Levando-se em consideração as afirmações descritas anteriormente, o problema da maximização da frequência natural com restrições de complacência, equilíbrio e volume é definido como:

$$\left\{ \begin{array}{l} \text{maximizar} \\ x \in \mathbb{R}^{n_{elem}}, u \in \mathbb{R}^{n_{gdl}} \end{array} \right. \lambda_{min}(x) \\ \left\{ \begin{array}{l} \text{sujeito a:} \\ \left( \sum_{i=1}^{n_{elem}} x_i K_i \right) u = p \\ p^\top u \leq \bar{\gamma} \\ \sum_{i=1}^{n_{elem}} x_i \leq \bar{V} \\ x_i \geq 0, \quad i = 1, \dots, n_{elem}. \end{array} \right. \quad (2.60)$$

onde  $\bar{\gamma}$  e  $\bar{V}$  são constantes positivas dadas e representam, respectivamente, os limites superiores da complacência e do volume. As variáveis de projeto são o deslocamento  $u \in \mathbb{R}^{n_{gdl}}$  e o volume dos elementos que compõem a estrutura  $x \in \mathbb{R}^{n_{elem}}$ . Isto representa um total de  $n_{elem} + n_{gdl}$  variáveis.

Em outras palavras, o problema pretende encontrar uma estrutura com distribuição de volumes  $x$ , dentre todas as estruturas que verificam determinado nível de deformação ( $\bar{\gamma}$ ) e tenham seu volume limitado a uma quantidade máxima ( $\bar{V}$ ). Este é um dos grandes desafios da engenharia mecânica: encontrar a estrutura  $x$  com a maior frequência possível limitando-se apenas a quantidade de material ( $\bar{V}$ ) e deformação ( $\bar{\gamma}$ ).

O problema (2.60) é denominado clássico por apresentar restrições inicialmente abordadas na literatura da programação não linear (restrições de igualdade e de desigualdade). No entanto, esta formulação apresenta uma série de dificuldades técnicas e grande parte delas gira em torno da função  $\lambda_{min}(x)$ , como apresentado abaixo [19]:

1. O número de variáveis de projeto pode ser muito grande, pois depende não somente da quantidade de elementos em que a estrutura foi discretizada ( $n_{elem}$ ), mas também do número de graus de liberdade ( $n_{gd}$ );
2. A função  $\lambda_{min}(x)$  pode ser descontínua quando certos componentes do vetor  $x$  são nulos;
3. A função  $\lambda_{min}(x)$  só será uma função diferenciável quando a multiplicidade de menor autovalor de (2.54) for um;
4. O cálculo de  $\lambda_{min}(x)$  apresenta-se como um desafio, pois, apesar da existência de procedimentos numéricos bem estabelecidos, ainda requer um tempo computacional razoável para que o resultado tenha uma precisão apropriada.

Para superar essas dificuldades, na seção seguinte apresenta-se um problema de programação semidefinida equivalente (*Semidefinite Programming*, SDP).

## 2.5.2 Formulação SDP

A variável de projeto  $u \in \mathbb{R}^{n_{gd}}$  pode ser eliminada do problema (2.60) tendo em vista a seguinte proposição, extraída de [15]:

**Proposição 2.5.1.** (*Conversão do Problema Clássico para o Problema SDP*). *Sejam  $x \in \mathbb{R}^{n_{elem}}$  e  $x \geq 0$  sendo fixo  $\bar{\gamma} \in \mathbb{R}$ . Então existe  $u \in \mathbb{R}^{n_{gd}}$  que satisfaz:*

$$\begin{aligned} \left( \sum_{i=1}^{n_{elem}} x_i K_i \right) u &= p \\ p^\top u &\leq \bar{\gamma} \end{aligned} \iff \begin{pmatrix} \bar{\gamma} & -p^\top \\ -p & K(x) \end{pmatrix} \succeq 0. \quad (2.61)$$

*Demonstração: ver [15].*

Desta forma, a restrição de igualdade  $(\sum_{i=1}^{n_{elem}} x_i K_i) u = p$  e a de desigualdade  $p^\top u \leq \bar{\gamma}$  são substituídas por uma restrição semidefinida, onde não aparece a variável de projeto  $u \in \mathbb{R}^{n_{gd}}$ .

Por outro lado, como visto em [20], para eliminar  $\lambda_{min}(x)$  da função objetivo de (2.60) basta seguir o raciocínio, baseando-se em (2.58): maximizar  $\lambda_{min}(x)$  equivale a maximizar  $\lambda$  onde  $\lambda \leq \lambda_{min}(x)$ . Isto equivale a minimizar  $-\lambda$  onde  $\lambda \leq \lambda_{min}(x)$ . Portanto,  $K(x) - \lambda M(x) \succeq 0$ .

Desta forma, o problema (2.60) equivale a:

$$\left\{ \begin{array}{l} \text{minimizar} \\ x \in \mathbb{R}^{n_{elem}}, \lambda \in \mathbb{R} \end{array} \quad -\lambda \right. \\ \left. \begin{array}{l} \text{sujeito a:} \\ \begin{pmatrix} \bar{\gamma} & -p^\top \\ -p & K(x) \end{pmatrix} \succeq 0 \\ \sum_{i=1}^{n_{elem}} x_i \leq \bar{V} \\ K(x) - \lambda M(x) \succeq 0 \\ x_i \geq 0, \quad i = 1, \dots, n_{elem}, \end{array} \right. \quad (2.62)$$

onde, nesta nova formulação, as variáveis de projeto são  $x$  e  $\lambda$ .

Ainda em conformidade com [20], (2.62) é um problema de programação semi-definida com uma restrição BMI (*Bilinear Matrix Inequality*), ou seja, possui uma desigualdade matricial bilinear representada por:

$$K(x) - \lambda M(x) \succeq 0, \quad (2.63)$$

portanto, este problema não é convexo e não pode ser resolvido por técnicas numéricas para problemas convexos. Em razão disso, ele será resolvido, nas seções seguintes, pela técnica proposta nesta dissertação.

# Capítulo 3

## Análise da Família de Algoritmos FDIPA

### 3.1 A Matriz do Sistema de Newton nos Algoritmos FDIPA

Nesta seção discute-se as principais vantagens e desvantagens de uma família de algoritmos de pontos interiores para Programação Semidefinida Não Linear. Mister mencionar que o real objetivo não é comentar o algoritmo como um todo, mas sim tratar os detalhes sobre a matriz do sistema de Newton nessas implementações, considerada muitas vezes como o grande gargalo para a resolução de problemas estruturais de grande porte.

Diferentes algoritmos podem ser obtidos usando-se as técnicas tipo Newton para a solução iterativa das condições de otimalidade de primeira ordem de Karush-Kuhn-Tucker. Todavia, por se constituir como um dos primeiros métodos que utiliza o conceito de pontos interiores para resolver problemas de programação semidefinida, todos os algoritmos apresentados aqui têm com base o FDIPA (Feasible Direction Interior Point Algorithm), idealizado por Herskovits [10].

A ideia por trás do método é resolver dois sistemas de equações lineares, ambos com a mesma matriz de coeficientes (que contém, implicitamente, as restrições do problema e suas derivadas), e realizar uma busca linear ou curvilínea restrita. Para isso, a partir de um ponto inicial no interior da região viável  $\Omega$  gera uma sequência de pontos interiores e calcula uma direção de descida e viável, de maneira que o valor da função objetivo vai decrescendo em cada iteração. Logo após, realiza-se uma atualização de parâmetros para iniciar uma nova iteração.

Na sequência apresenta-se três versões do FDIPA, sendo possível realizar uma análise comparativa entre eles e visualizar a necessidade de aprimorá-los para que possam ser utilizados no caso em questão.

### 3.1.1 O Algoritmo FDIPA-SDP

Inicialmente pensado para trabalhar apenas com restrições de igualdade e desigualdade, o algoritmo FDIPA foi generalizado para resolver problemas de programação semidefinida [9]. Esta versão, denominada FDIPA-SDP, permite trabalhar apenas com restrições matriciais, deste modo, seu estudo gira em torno de problemas como o formulado em (2.29).

Em trabalhos como o apresentado em [7], utiliza-se o FDIPA-SDP para resolver problemas de otimização de material, onde as restrições vetoriais (ou restrições de desigualdade) são diagonalizadas junto com as restrições matriciais, formando uma única matriz. Porém de acordo com análises realizadas em [21], esta formulação só permite trabalhar com estruturas onde  $n$  e  $q$  são pequenos, devido a impossibilidade no armazenamento, em memória, da matriz completa do gradiente do sistema de equações de Newton, requisitos essenciais para esta família de algoritmos.

Ainda de acordo com [21], se a matriz de restrição for armazenada por completo, o armazenamento da derivada precisa de muita memória, na ordem dos Gigabytes (GB) e o número máximo de elementos finitos que esta formulação permite trabalhar para problemas de otimização de material é 16.

Fazendo uma alteração no armazenamento da matriz de restrições e aproveitando sua condição de ser simétrica e esparsa, como descrito em [9], pode-se aumentar o número de elementos finitos para a resolução do problema, chegando a 100 elementos. Entretanto, o algoritmo não mantém a estrutura da matriz de restrições original, sendo visto como uma desvantagens quando se trabalha com o método dos elementos finitos.

### 3.1.2 O Algoritmo FDIPA-GSDP(1)

Para resolver problemas com  $n$  e  $q$  maiores, utiliza-se o FDIPA-GSDP, um algoritmo geral que permite trabalhar com restrições vetoriais separadas das restrições matriciais.

Assim, o novo problema pode ser formulado como:

$$\left\{ \begin{array}{l} \underset{x \in \mathbb{R}^n}{\text{minimizar}} \quad f(x) \\ \text{sujeito a:} \quad g(x) \leq 0 \\ \quad \quad \quad \text{e} \quad A(x) \preceq 0 \end{array} \right. \quad (3.1)$$

onde  $f(x) \in \mathbb{R}$ ,  $g(x) \in \mathbb{R}^m$ ,  $A(x) \in \mathbb{S}^q$  e  $\Omega = \{x \in \mathbb{R}^n \mid g(x) \leq 0 \text{ e } A(x) \preceq 0\}$ .

Como o algoritmo descrito a seguir é o primeiro entres outros dois, que serão comentados posteriormente, a resolver problemas a partir desta formulação, ele será denotado por FDIPA-GSDP(1).

Para o problema (3.1) as condições de otimalidade de primeira ordem de KKT podem ser escritas como:

$$\nabla f(x) + \nabla g(x)\gamma + \nabla A(x)\lambda = 0 \quad (3.2)$$

$$G(x)\gamma = 0 \quad (3.3)$$

$$\Lambda A(x) = 0 \quad (3.4)$$

$$g(x) \leq 0 \quad (3.5)$$

$$A(x) \preceq 0 \quad (3.6)$$

$$\gamma \geq 0 \quad (3.7)$$

$$\Lambda \succeq 0 \quad (3.8)$$

onde  $\gamma \in \mathbb{R}^m$  e  $\Lambda \in \mathbb{S}^q$  são os multiplicadores de Lagrange,  $G(x)$  é uma matriz diagonal tal que  $G_{ii} = g_i(x)$  e  $\lambda = \text{vec}(\Lambda)$ .

Para resolver este problema, o algoritmo define uma função vetorial  $\phi : \mathbb{R}^{n+m+q^2} \rightarrow \mathbb{R}^{n+m+q^2}$ , tal que:

$$\phi(x, \gamma, \lambda) = \begin{bmatrix} \phi_l(x, \gamma, \lambda) \\ \phi_v(x, \gamma, \lambda) \\ \phi_m(x, \gamma, \lambda) \end{bmatrix}, \quad (3.9)$$

onde:

$$\phi_l(x, \gamma, \lambda) = \nabla f(x) + \nabla g(x)\gamma + \nabla A(x)\lambda \quad (3.10)$$

$$\phi_v(x, \gamma, \lambda) = G(x)\gamma \quad (3.11)$$

$$\phi_m(x, \gamma, \lambda) = \text{vec}(\Lambda A(x)) \quad (3.12)$$

Através das propriedades do produto de Kronecker, definidas em (2.2.8), o produto  $\Lambda A$  é vetorizado, conseqüentemente, o termo  $\phi_m(x, \gamma, \lambda)$  da função vetorial  $\phi$  pode ser redefinido de duas maneiras:

$$\phi_m(x, \gamma, \lambda) = \text{vec}(\Lambda A(x)I) = (I \otimes \Lambda)\text{vec}(A(x)) \quad (3.13)$$

ou

$$\phi_m(x, \gamma, \lambda) = \text{vec}(I\Lambda A(x)) = (A(x) \otimes I)\text{vec}(\Lambda). \quad (3.14)$$

Para calcular as raízes da função pelo método de Newton, calcula-se a matriz Jacobiana de  $\phi$ , que, de acordo com a Definição 22, é equivalente à forma  $\nabla \phi^\top$ , o que implica a derivação de  $\phi$  em relação às variáveis  $x, \gamma$  e  $\lambda$ , chegando-se à seguinte

forma:

$$\nabla\phi(x, \gamma, \lambda)^\top = \begin{bmatrix} \nabla_x\phi_l(x, \gamma, \lambda) & \nabla_\gamma\phi_l(x, \gamma, \lambda) & \nabla_\lambda\phi_l(x, \gamma, \lambda) \\ \nabla_x\phi_v(x, \gamma, \lambda) & \nabla_\gamma\phi_v(x, \gamma, \lambda) & \nabla_\lambda\phi_v(x, \gamma, \lambda) \\ \nabla_x\phi_m(x, \gamma, \lambda) & \nabla_\gamma\phi_m(x, \gamma, \lambda) & \nabla_\lambda\phi_m(x, \gamma, \lambda) \end{bmatrix}. \quad (3.15)$$

Usando a equação (3.10), tem-se que:

$$\nabla_x\phi_l(x, \gamma, \lambda) = \nabla_x(\nabla f(x) + \nabla g(x)\gamma + \nabla A(x)\lambda) = H(x, \gamma, \lambda). \quad (3.16)$$

$$\nabla_\gamma\phi_l(x, \gamma, \lambda) = \nabla g. \quad (3.17)$$

$$\nabla_\lambda\phi_l(x, \gamma, \lambda) = \nabla A. \quad (3.18)$$

Pela equação (3.11), define-se:

$$\nabla_x\phi_v(x, \gamma, \lambda) = \Gamma\nabla g^\top. \quad (3.19)$$

$$\nabla_\gamma\phi_v(x, \gamma, \lambda) = G. \quad (3.20)$$

$$\nabla_\lambda\phi_v(x, \gamma, \lambda) = 0. \quad (3.21)$$

onde  $\Gamma_{ii} = \gamma_i$ .

E, pela equação (3.12), mostra-se que:

$$\nabla_x\phi_m(x, \gamma, \lambda) = (I \otimes \Lambda)\nabla A^\top. \quad (3.22)$$

$$\nabla_\gamma\phi_m(x, \gamma, \lambda) = 0. \quad (3.23)$$

$$\nabla_\lambda\phi_m(x, \gamma, \lambda) = (A(x) \otimes I). \quad (3.24)$$

Todas essas operações resultam na seguinte matriz Jacobiana:

$$\nabla\phi(x, \gamma, \lambda)^\top = \begin{bmatrix} H(x, \gamma, \lambda) & \nabla g(x) & \nabla A(x) \\ \Gamma\nabla g(x)^\top & G(x) & 0 \\ (I \otimes \Lambda)\nabla A(x)^\top & 0 & (A(x) \otimes I) \end{bmatrix}. \quad (3.25)$$

Substituindo a Hessiana  $H(x, \gamma, \lambda)$  por uma matriz  $B \in \mathbb{S}_{++}^n$ , que pode ser uma matriz quase-Newton, uma matriz identidade ou qualquer outra matriz definida positiva de tamanho  $n \times n$ , finalmente tem-se a matriz do sistema de Newton:

$$N(x, \gamma, \lambda) = \begin{bmatrix} B & \nabla g(x) & \nabla A(x) \\ \Gamma\nabla g(x)^\top & G(x) & 0 \\ (I \otimes \Lambda)\nabla A(x)^\top & 0 & (A(x) \otimes I) \end{bmatrix} \in \mathbb{R}^{(n+m+q^2) \times (n+m+q^2)}. \quad (3.26)$$

Uma iteração de Newton para a solução do sistema de equações (3.2) até (3.8)

em  $(x, \gamma, \lambda)$  é indicado como:

$$\begin{bmatrix} B^k & \nabla g(x^k) & \nabla A(x^k) \\ \Gamma^k \nabla g(x^k)^\top & G(x^k) & 0 \\ (I \otimes \Lambda^k) \nabla A(x^k)^\top & 0 & (A(x^k) \otimes I) \end{bmatrix} \begin{bmatrix} x^{k+1} - x^k \\ \gamma^{k+1} - \gamma^k \\ \lambda^{k+1} - \lambda^k \end{bmatrix} = - \begin{bmatrix} \nabla f(x^k) + \nabla g(x^k) \gamma^k + \nabla A(x^k) \lambda^k \\ G(x^k) \gamma^k \\ \text{vec}(\Lambda^k A(x^k)) \end{bmatrix}, \quad (3.27)$$

onde  $(x^k, \gamma^k, \lambda^k)$  é o ponto inicial da iteração e  $(x^{k+1}, \gamma^{k+1}, \lambda^{k+1})$  é a nova estimativa.

Em cada iteração é calculada uma direção de busca  $d^k$ , a qual tem que ser uma direção de descida da função objetivo no ponto  $x^k$  e uma direção viável em  $\Omega$ . O cálculo desta direção de busca é realizado em duas etapas e com base no método de Newton, como visto no sistema (3.27).

Na primeira etapa, define-se uma direção  $d_0^k = x^{k+1} - x^k$ , obtendo-se o seguinte sistema linear:

$$\begin{bmatrix} B^k & \nabla g(x^k) & \nabla A(x^k) \\ \Gamma^k \nabla g(x^k)^\top & G(x^k) & 0 \\ (I \otimes \Lambda^k) \nabla A(x^k)^\top & 0 & (A(x^k) \otimes I) \end{bmatrix} \begin{bmatrix} d_0^k \\ \gamma_0^k \\ \lambda_0^k \end{bmatrix} = - \begin{bmatrix} \nabla f(x^k) \\ 0 \\ 0 \end{bmatrix}. \quad (3.28)$$

A partir deste sistema, se  $d_0 = 0$  então  $x$  é um ponto estacionário de KKT, caso contrário, a direção encontrada não é necessariamente viável. Para que  $d_0$  seja viável, o sistema (3.28) é perturbado para produzir uma deflexão na direção de descida, no sentido do interior da região viável  $\Omega$ . Em busca desta direção, a segunda etapa consiste em resolver o seguinte sistema de equações:

$$\begin{bmatrix} B & \nabla g(x) & \nabla A(x) \\ \Gamma \nabla g(x)^\top & G(x) & 0 \\ (I \otimes \Lambda) \nabla A(x)^\top & 0 & (A(x) \otimes I) \end{bmatrix} \begin{bmatrix} d_1 \\ \gamma_1 \\ \lambda_1 \end{bmatrix} = - \begin{bmatrix} 0 \\ \gamma \\ \lambda \end{bmatrix}. \quad (3.29)$$

Percebe-se que as duas etapas envolvem a resolução de um sistema linear com a mesma matriz, (3.26), e apesar do algoritmo FDIPA-GSDP(1) trabalhar com as restrições vetoriais e matriciais de maneira independente, a matriz de Newton gerada por ele é muito grande,  $(n + m + q^2) \times (n + m + q^2)$ , principalmente quando a discretização da estrutura é realizada com uma grande quantidade de elementos finitos.

Segundo os estudos e testes realizados em [21], esta formulação só é capaz de resolver problemas com um número pequeno de elementos finitos ( $\leq 300$ ), pois acima disso a quantidade de memória necessária para o armazenamento da matriz de Newton completa também alcança a ordem dos Gigabytes, fazendo com que

o FDIPA-GSDP(1) não seja proveitoso para trabalhar com um grande número de variáveis de projeto e restrições, já que o problema torna-se computacionalmente intratável.

Aqui é, mais uma vez, importante destacar que quanto mais elementos finitos são utilizados em uma discretização, mais fiel a realidade será a sua representação, daí a necessidade de implementar um algoritmo que trabalhe de forma satisfatória com problemas de grande porte.

Ressalta-se também que a grande responsável pelo problema no armazenamento da matriz de Newton e a consequente ineficiência deste algoritmo nos casos de grande porte é a derivada da restrição matricial, cuja dimensão é  $n \times q^2$ .

### 3.1.3 O Algoritmo FDIPA-GSDP(2)

A busca por um algoritmo robusto com o objetivo de trabalhar com um grande número de elementos finitos na discretização de uma estrutura levou à implementação do FDIPA-GSDP(2), cujo desenvolvimento encontra-se em [22].

Este otimizador modifica o sistema de equações interno do FDIPA-GSDP(1), melhorando seu desempenho por possibilitar a solução de problemas com um maior número de variáveis de projeto e restrições como também por reduzir o tempo de cálculo da resposta.

Considerando o problema (3.1), a seguinte formulação é adotada para resolver o problema de armazenamento da matriz com um grande quantidade de elementos finitos em que pode ser discretizada a estrutura:

Do sistema (3.28) tem-se as seguintes equações:

$$Bd_0 + \nabla g \gamma_0 + \nabla A \lambda_0 = -\nabla f. \quad (3.30)$$

$$\Gamma \nabla g^\top d_0 + G \gamma_0 = 0. \quad (3.31)$$

$$(I \otimes \Lambda) \nabla A^\top d_0 + (A \otimes I) \lambda_0 = 0. \quad (3.32)$$

A partir de (3.32) tem-se que:

$$\lambda_0 = -(A \otimes I)^{-1} (I \otimes \Lambda) \nabla A^\top d_0, \quad (3.33)$$

seja  $U \in \mathbb{R}^{q^2 \times n}$ ,

$$U = -(A \otimes I)^{-1} (I \otimes \Lambda) \nabla A^\top. \quad (3.34)$$

Utilizando as propriedades do produto de Kronecker, definidas em (2.2.8), tem-se:

$$U = -(A^{-1} \otimes \Lambda) \nabla A^\top. \quad (3.35)$$

Seja  $U_k$  a  $k$ -ésima coluna de  $U$ , então:

$$U_k = \text{vec} \left( -\Lambda \frac{\partial A}{\partial x_k} A^{-1} \right); k = 1, \dots, n. \quad (3.36)$$

Substituindo (3.33) e (3.35) em (3.30), obtém-se o sistema:

$$\{B + \nabla AU\}d_0 + \nabla g\gamma_0 = -\nabla f. \quad (3.37)$$

O produto  $\left(\Lambda \frac{\partial A}{\partial x_k} A^{-1}\right) \in \mathbb{R}^{q \times q}$  é uma matriz que é transformada em um vetor  $U_k \in \mathbb{R}^{q^2 \times 1}$  mediante o operador  $\text{vec}$ , visto na Definição 11, e resolvido para  $k = 1, \dots, n$  variáveis do problema. Cada vetor será uma coluna da matriz  $U \in \mathbb{R}^{q^2 \times n}$ , ou seja:

$$U = \begin{bmatrix} | & \cdots & | \\ U_1 & \cdots & U_n \\ | & \cdots & | \end{bmatrix}. \quad (3.38)$$

Uma forma de calcular  $U_k$  (a coluna  $k$  de  $U$ ) é transformando a equação (3.36) em um sistema linear com a matriz  $A$  e no lado direito a matriz  $-\Lambda \frac{\partial A}{\partial x_k}$ . Portanto, para calcular  $U$  será necessário resolver  $n \times q$  sistemas lineares com a mesma matriz  $A(x)$ . Assim, (3.28) fica reduzido a um novo sistema de equações simétrico:

$$\{B + \nabla AU\}d_0 + \nabla g\gamma_0 = -\nabla f \quad (3.39)$$

$$\Gamma \nabla g^\top d_0 + G\gamma_0 = 0 \quad (3.40)$$

onde, (3.28) e (3.29) são equivalentes a:

$$\begin{bmatrix} B + \nabla A(x)U & \nabla g(x) \\ \Gamma \nabla g(x)^\top & G(x) \end{bmatrix} \begin{bmatrix} d_0 & d_1 \\ \gamma_0 & \gamma_1 \end{bmatrix} = - \begin{bmatrix} \nabla f(x) & 0 \\ 0 & \gamma \end{bmatrix}. \quad (3.41)$$

Nota-se que neste novo sistema, (3.41), a matriz de Newton foi reduzida de maneira significativa, pois agora possui dimensão  $(n + m) \times (n + m)$ .

Com estas modificações, percebe-se que a diferença entre o FDIPA-GSDP(1) e o FDIPA-GSDP(2) é que o primeiro resolve os sistemas lineares (3.28) e (3.29) enquanto que o segundo resolve o sistema linear (3.41), no qual foram eliminadas as variáveis  $\lambda_0$  e  $\lambda_1$ . Deste modo, também é possível inferir que a grande vantagem deste frente àquele está justamente no tamanho do sistema linear no qual executa, já que isto o permite lidar com problemas estruturais com um maior número de elementos ( $>1000$ ), em contrapartida, ele precisa, antes de resolver o sistema (3.41), calcular a matriz  $U$ .

O cálculo de  $U$  tem que ser realizado em cada iteração do algoritmo (isto é, para cada  $x$ ), portanto representa um enorme esforço computacional, levando tempos

muito altos, segundo [21], cerca de 5 horas para obter uma iteração e esta é uma grande desvantagem do FDIPA-GSDP(2).

Entretanto, ainda em conformidade com [21], mantendo-se o mesmo sistema de equações e armazenando a matriz de restrições e sua derivada como matrizes multidimensionais ( $A(x) \in \mathbb{R}^{(q \times q) \times n_e}$  e  $\nabla A(x) \in \mathbb{R}^{(n \times q^2) \times n_e}$ , onde  $n_e$  é o número de elementos finitos da discretização), consegue-se um desempenho ainda melhor, sendo possível trabalhar com pouco mais de 1200 elementos em um tempo de 3600 segundos por iteração. Estes valores, disponíveis no trabalho citado acima, foram obtidos em testes aplicados a um problema de otimização de material, portanto, diferente do abordado nesta dissertação.

## Capítulo 4

# FDIPA-GSDP(3): uma Nova Técnica de Programação Semidefinida para Problemas de Grande Porte

Como já foi mencionado, a técnica usada pelo FDIPA consiste em resolver, a cada iteração, dois sistemas lineares em busca de uma direção que seja ao mesmo tempo de descida e viável. A determinação desta direção representa o passo de maior esforço computacional, principalmente quando se trabalha com sistemas de grande porte.

Em aplicações que exigem uma grande quantidade de componentes na discretização, o armazenamento da matriz do sistema é muitas vezes inviável, pois o tamanho da memória para armazená-la por inteiro aumenta proporcionalmente ao número de elementos finitos, sendo este, então, um ponto indesejável para o objetivo deste trabalho.

Desta forma, a intenção deste capítulo é apresentar um novo algoritmo para resolver o problema (3.1), baseado no FDIPA-GSDP(2), capaz de reduzir ainda mais a matriz do sistema visando tornar possível a aplicação do FDIPA em problemas que exigem um grande esforço computacional e, conseqüentemente, diminuir o tempo gasto na busca pela solução ótima.

### 4.1 Aperfeiçoamento do Algoritmo

A partir da análise realizada no capítulo anterior, percebe-se que o principal obstáculo para resolver problemas de otimização de grande porte com o auxílio da família FDIPA é a formação do sistema de equações presente na formulação, mais especificamente o tamanho da matriz de Newton, que carrega, implicitamente, as restrições do problema e suas respectivas derivadas.

Apesar do FDIPA ter sido bastante aprimorado ao longo dos últimos anos, novas aplicações exigem do algoritmo melhorias em termos de robustez, eficiência e confiabilidade. Neste sentido, a presente dissertação apresenta uma técnica capaz de reduzir o tamanho do sistema (3.41), tornando mais simples e eficaz a sua resolução, ao passo que permite o tratamento de problemas com uma quantidade ainda maior de elementos finitos. Os detalhes desta nova técnica são exibidos abaixo.

Como  $x$  é um ponto localizado dentro da região viável ( $\Omega$ ), então  $g_i(x) < 0$ , isso implica que  $G(x)$  é inversível, pois  $G(x) \prec 0$ , assim, isolando-se o termo  $\gamma_0$  na equação (3.40):

$$\gamma_0 = -G^{-1}\Gamma\nabla g^\top d_0. \quad (4.1)$$

Substituindo (4.1) em (3.39) e colocando o termo  $d_0$  em evidência, tem-se:

$$(B + \nabla AU - \nabla g G^{-1}\Gamma\nabla g^\top)d_0 = -\nabla f. \quad (4.2)$$

Utilizando o mesmo raciocínio, mas agora partindo do sistema (3.41), vê-se que:

$$\{B + \nabla AU\}d_1 + \nabla g\gamma_1 = 0. \quad (4.3)$$

$$\Gamma\nabla g^\top d_1 + G\gamma_1 = -\gamma. \quad (4.4)$$

De (4.4) é possível inferir que:

$$\gamma_1 = -G^{-1}(\gamma + \Gamma\nabla g^\top d_1). \quad (4.5)$$

E, substituindo (4.5) em (4.3), chega-se a:

$$(B + \nabla AU - \nabla g G^{-1}\Gamma\nabla g^\top)d_1 = \nabla g G^{-1}\gamma. \quad (4.6)$$

Realizadas as modificações acima, obtém-se, na forma matricial, o novo sistema de equações do algoritmo FDIPA, usado para o cálculo da direção de busca:

$$\begin{bmatrix} B + \nabla AU - \nabla g G^{-1}\Gamma\nabla g^\top \\ \Gamma\nabla g^\top \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \end{bmatrix} = \begin{bmatrix} -\nabla f \\ \nabla g G^{-1}\gamma \end{bmatrix}. \quad (4.7)$$

Neste novo formato, as variáveis  $\gamma_0$  e  $\gamma_1$  foram retiradas do sistema, sendo calculadas através das equações (4.1) e (4.5), respectivamente, logo após a obtenção dos valores de  $d_0$  e  $d_1$ .

Em razão destas alterações, percebe-se que o tamanho da matriz de Newton no sistema (4.7) foi reduzido de maneira considerável, passando a ter dimensão  $n \times n$ , bem menor que a da, até então, melhor versão anterior do FDIPA, que era de  $(n + m) \times (n + m)$ . Esta redução permite ao algoritmo trabalhar com estruturas de portes maiores, uma vez que aumenta a quantidade de elementos finitos que ele é

capaz de executar.

Além de ocupar bem menos espaço em memória durante sua execução, uma outra característica desta nova matriz de Newton do sistema do FDIPA é o fato de ser simétrica e definida positiva, como pode ser visto em seguida:

Empregando o valor de  $U$  obtido em (3.35) na equação (4.2), chega-se a:

$$[B - \nabla A(A^{-1} \otimes \Lambda)\nabla A^\top - \nabla g G^{-1} \Gamma \nabla g^\top] d_0 = -\nabla f. \quad (4.8)$$

A fim de provar estas características, é essencial uma análise da expressão (4.8), para isso, considere as seguintes afirmações, vistas na Seção 3.1.2:

- $B$  é definida positiva;
- $A$  é semidefinida negativa;
- $\Lambda$  é definida positiva;
- $G$  é definida negativa;
- $\Gamma$  é definida positiva.

A partir do itens 6 e 7 do Teorema 2.2.8 e do Teorema 2.2.5, conclui-se que o produto de Kronecker ( $A^{-1} \otimes \Lambda$ ) resulta em uma matriz semidefinida negativa. E, levando-se em consideração o Teorema 2.2.6, o produto  $\nabla A(A^{-1} \otimes \Lambda)\nabla A^\top$  também é uma matriz semidefinida negativa, porém o sinal negativo ( $-$ ) que precede a expressão torna o resultado final uma matriz semidefinida positiva.

Pelas mesmas razões, o termo  $-\nabla g G^{-1} \Gamma \nabla g^\top$  também gera uma matriz definida positiva e, conseqüentemente, o somatório  $[B - \nabla A(A^{-1} \otimes \Lambda)\nabla A^\top - \nabla g G^{-1} \Gamma \nabla g^\top]$  é uma matriz definida positiva. O fato de haver uma comutatividade entre as matrizes  $A^{-1}$  e  $\Lambda$  e entre  $G^{-1}$  e  $\Gamma$  garante a simetria da matriz.

Uma das vantagens de ser definida positiva e simétrica é a possibilidade de resolução do sistema através de outros métodos, como o Método dos Gradientes Conjugados, por exemplo, que é totalmente viável para este caso. Entretanto, optou-se por resolver o sistema (4.7) pelo método direto.

## 4.2 Testes Preliminares

Em testes realizados nesta pesquisa - detalhados melhor no próximo capítulo - o algoritmo resolveu o problema proposto com uma estrutura discretizada em 1580 elementos, conseguindo atingir o tempo de 240 segundos por iteração. Esses números mostram que esta formulação é mais eficiente que as versões anteriores, apresentadas no Capítulo 3.

Porém, diante de todas as melhorias motivadas por este novo sistema, ficou evidenciado também um novo problema a ser resolvido no algoritmo, pois o cálculo da matriz  $U$  transforma-se agora no novo gargalo para a execução de problemas com uma grande quantidade de elementos finitos. No entanto, apesar da descoberta desta dificuldade, não abordou-se aqui esta questão, ficando como sugestão para trabalhos futuros.

### 4.3 O Novo Algoritmo

O FDIPA requer um ponto inicial no interior da região viável  $\Omega$  para gerar uma sequência de pontos interiores e calcular uma direção que seja, ao mesmo tempo, de descida e viável, de maneira que o valor da função objetivo decresça em cada iteração. No final de cada uma dessas iterações, realiza-se uma atualização de parâmetros para uma nova repetição.

Desta forma, a nova versão do algoritmo FDIPA-GSDP, denominado FDIPA-GSDP(3) <sup>1</sup>, cuja prova da convergência global a pontos de KKT pode ser realizada com base nas publicações [7, 9, 21, 22], consiste no seguinte:

**Parâmetros:**  $\xi \in (0, 1)$ ,  $\eta \in (0, 1)$ ,  $\varphi > 0$  e  $\nu \in (0, 1)$ .

**Dados Iniciais:**  $x \in \text{int}(\Omega)$ ,  $\gamma > 0$ ,  $B \in \mathbb{S}_{++}^n$  e  $\Lambda \in \mathbb{S}_{++}^q$  comutando com  $A(x)$ .

**Passo 1:** Cálculo da Direção de Busca

**Passo 1a:** Calcular a matriz  $U \in \mathbb{R}^{q^2 \times n}$ , conforme (3.36);

**Passo 1b:** Resolver o sistema linear para obter  $d_0 \in \mathbb{R}^n$  e  $d_1 \in \mathbb{R}^n$  e, em seguida, substituir seus valores nas equações (4.1) e (4.5) para obter  $\gamma_0 \in \mathbb{R}^m$  e  $\gamma_1 \in \mathbb{R}^m$ , respectivamente:

$$\begin{bmatrix} B + \nabla A U - \nabla g G^{-1} \Gamma \nabla g^\top \end{bmatrix} \begin{bmatrix} d_0 & d_1 \end{bmatrix} = \begin{bmatrix} -\nabla f & \nabla g G^{-1} \gamma \end{bmatrix};$$

Se  $d_0 = 0$ , parar.

**Passo 1c:** Se  $d_1^\top \nabla f(x) > 0$ , calcule:

$$\rho = \min \left\{ \varphi \|d_0\|^2, (\xi - 1) \frac{d_0^\top \nabla f(x)}{d_1^\top \nabla f(x)} \right\};$$

---

<sup>1</sup>Por questões de facilitar a leitura do texto, principalmente no que tange as tabelas de resultados, optou-se por esta denominação, apesar de já existir esta mesma nomenclatura, para um algoritmo distinto, em [21].

caso contrário, faça:

$$\rho = \varphi \|d_0\|^2;$$

**Passo 1d:** Calcule a direção de busca  $d$

$$d = d_0 + \rho d_1.$$

**Passo 2:** Busca Linear

Calcule o comprimento do passo,  $t$ , o primeiro número da série  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que verifique, simultaneamente, as três equações abaixo:

$$\begin{aligned} f(x + td) &\leq f(x) + t\eta \nabla f(x)^\top d \\ g_i(x + td) &< 0; \quad i = 1, \dots, m \\ A(x + td) &\prec 0. \end{aligned}$$

**Passo 3:** Atualizações

**Passo 3a:** Obtenha o novo ponto:  $x = x + td$ ;

**Passo 3b:** Defina novos valores para  $B \in \mathbb{S}_{++}^n$ ;

**Passo 3c:** Defina novos valores para  $\gamma_i > 0, i = 1, \dots, m$ ;

**Passo 3d:** Defina novos valores para  $\Lambda \in \mathbb{S}_{++}^q$  comutando com  $A(x)$ ;

**Passo 3e:** Retorne ao Passo 1.

Cumpra destacar que todas as versões do FDIPA exibidas nesse trabalho seguem esta mesma proposta de algoritmo, com exceção do Passo 1, mais especificamente os Passos 1a e 1b, já que eles resolvem um sistema de equações diferente, sendo justamente esta questão o diferencial de cada um.

Conforme [23], cada um desses sistemas tem uma nomenclatura específica, isto posto, os sistemas lineares resolvidos tanto no FDIPA-GSDP(1) quanto no FDIPA-GSDP(2) são denominados “Primal-Duais”, pois as incógnitas estão relacionadas tanto às variáveis primais (variáveis de projeto) quanto às duais (multiplicadores de Lagrange). As matrizes destes sistemas lineares não são simétricas.

E, por último, o sistema obtido pelo FDIPA-GSDP(3) pode ser chamado de “Primal”, já que sua incógnita relaciona-se diretamente às variáveis primais (primárias). Observa-se também, como aqui já foi comprovado, que este último sistema é simétrico e positivo definido.

# Capítulo 5

## Testes Numéricos

Com a finalidade de avaliar o comportamento do FDIPA-GSDP(3), algoritmo tema desta dissertação, no presente capítulo apresentam-se os resultados de testes numéricos correspondentes a sua aplicação em um problema clássico de otimização estrutural amplamente discutido na literatura da engenharia mecânica e já abordado na Seção 2.5: a maximização da frequência natural de estruturas sujeita ao equilíbrio de condições e restrições de volume e energia de deformação (complacência).

Para tanto, realizou-se uma comparação entre o novo algoritmo e as versões FDIPA-GSDP(1), FDIPA-GSDP(2). Pelo fato do FDIPA-SDP, discutido na Seção 3.1.1, ter sido descartado sumariamente em [21], quando se trata de problemas de grande porte, tal metodologia não foi alvo destes testes.

Optou-se, então, por focar em três exemplos onde o diferencial é a estrutura mecânica - treliça plana - que os compõe, resumidos agora, mas cujos detalhes serão mostrados nas seções subsequentes:

- **Exemplo 1:** treliça bidimensional de 26 barras;
- **Exemplo 2:** treliça bidimensional de 534 barras;
- **Exemplo 3:** treliça bidimensional de 1580 barras;

### 5.1 Detalhes do Problema Semidefinido

O grande objetivo da otimização estrutural é selecionar uma configuração ótima, dentro de um conjunto de estruturas viáveis candidatas, que minimize uma função custo e verifique um conjunto de restrições mecânicas impostas.

Considerando o problema em questão, apresentado em (2.62), o grande desafio é encontrar uma estrutura com distribuição de volumes  $x$  (vetor formado pelos volumes de todas as barras que compõem a treliça) e que tenha a maior frequência

natural possível, calculada através do seu menor autovalor,  $\lambda$ , limitando-se apenas pela quantidade de material ( $\bar{V}$ ) e nível de deformação ( $\bar{\gamma}$ ).

Apesar de serem compostos por estruturas mecânicas com diferentes arranjos e quantidade de barras, todos os exemplos abordados aqui tem as seguintes características em comum:

- **Função Objetivo:**

$$f(x, \lambda) = -\lambda \quad (5.1)$$

- **Restrição de Desigualdade (restrição de volume):**

$$\sum_{i=1}^{n_{elem}} x_i - \bar{V} \leq 0 \quad (5.2)$$

- **Função de Restrições Matriciais ( $A(x) \preceq 0$ ):**

$$A(x) = A_0 + \sum_{i=1}^{n_{elem}} x_i A_i, \quad (5.3)$$

onde:

$$A_0 = \begin{bmatrix} \begin{bmatrix} -\bar{\gamma} & p^\top \\ p & 0 \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix} \quad (5.4)$$

e

$$A_i = \begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -K_i \end{bmatrix} & 0 \\ 0 & -K_i + \lambda M_i \end{bmatrix}, \quad (5.5)$$

sendo  $K_i$  e  $M_i$ , respectivamente, as matrizes de rigidez e de massa da barra (elemento)  $i$ .

- **Restrições de Caixa:**

$$-x_i \leq 0, \quad i = 1, \dots, n_{elem}. \quad (5.6)$$

O cálculo do gradiente da função objetivo e das restrições em relação às variáveis de projeto foi realizado por métodos analíticos em todos os exemplos.

No que tange a dimensão do problema (número de variáveis de projeto e número de restrições), a notação empregada nos exemplos que serão desenvolvidos na sequência deste capítulo está descrita na Tabela 5.1.

Notação	Descrição
$n$	Número de variáveis de projeto
$m$	Número de restrições de desigualdade
$q$	Número de linhas ou colunas da restrição matricial
$nc$	Número de restrições de caixa

Tabela 5.1: Notação empregada para definir a dimensão do problema

De posse de todas estas informações, em todos os exemplos, pode-se desenvolver os cálculos:

- $n = n_{elem} + 1$  (número de barras da treliça mais  $\lambda$ , menor autovalor);
- $m = 1$ ;
- $q = 1 + n_{gdl} + n_{gdl}$ ;
- $nc = n_{elem}$ .

Por outro lado, a notação exibida na Tabela 5.2 representa o esforço computacional do algoritmo para atingir a solução ótima do problema:

Notação	Descrição
$NI$	Número de iterações para encontrar o ponto ótimo
$NI_{pv}$	Número de iterações para encontrar um ponto inicial viável
$TE$	Tempo de execução (em segundos)

Tabela 5.2: Notação empregada para definir o esforço computacional

Nos casos onde as restrições de desigualdade e/ou as restrições matriciais impostas no problema teste não sejam verificadas em um ponto inicial, deve-se empregar o seguinte problema matemático auxiliar

$$\begin{cases} \underset{x \in \mathbb{R}^n, z \in \mathbb{R}}{\text{minimizar}} & z \\ \text{sujeito a:} & g(x) \leq z \\ & A(x) - zI \leq 0 \end{cases} \quad (5.7)$$

o qual fornece um ponto inicial viável (dentro da região  $\Omega$ ). Aqui,  $z$  representa uma variável escalar auxiliar e, quando, no processo de otimização, seu valor torna-se negativo, tem-se que  $g(x) < 0$  e  $A(x) \prec 0$ , portanto,  $x$  é um ponto viável.

## 5.2 Dados de Controle do Algoritmo

Nesta seção destacam-se os dados de controle considerados para a realização dos testes: parâmetros de controle do algoritmo, os dados iniciais, os critérios de parada e as rotinas de atualização dos dados.

- **Parâmetros de Controle**

Para o cálculo da deflexão ( $\rho$ ), definiu-se  $\phi = 1$ , já o parâmetro  $\xi$ , cujo valor varia em cada teste realizado, será discutido especificamente em cada exemplo. Da mesma forma, mas agora para a realização da busca linear de Armijo, realizada de acordo com [5], fixou-se  $\nu = 0,7$ , enquanto que  $\eta$  terá valores distintos em cada execução.

- **Dados Iniciais**

Os dados iniciais correspondem aos valores iniciais dos multiplicadores de Lagrange e da matriz Quase-Newton. Assim, têm-se:

- Os valores iniciais dos multiplicadores de Lagrange das restrições vetoriais de desigualdade foram  $\gamma_i = 1$ , para  $i = 1, \dots, m$ ;
- Definiu-se uma matriz identidade como valor inicial dos multiplicadores de Lagrange das restrições matriciais semidefinidas, assim,  $\Lambda = I$ ;
- O valor inicial da matriz quase-Newton também foi uma identidade, ou seja  $B = I$ .

- **Critérios de Parada**

- Os critérios de parada do algoritmo, considerados em todos os testes, foram a norma da direção de busca  $\|d_0\| < 1 \times 10^{-4}$  e a norma do gradiente da função Lagrangeana  $\|\nabla L\| < 1 \times 10^{-4}$ ;
- As condições de complementariedade das restrições vetoriais e matriciais foram quantificadas em cada iteração para verificar suas precisões:  $\|G\gamma_0\| < 1 \times 10^{-4}$  e  $\|tr(\Lambda_0 A)\| < 1 \times 10^{-4}$ .
- O algoritmo encerra-se também caso alcance um número máximo de iterações, definida como 10000. Limita-se também a este mesmo valor a busca pelo comprimento do passo, dado em uma determinada direção, para obter um novo ponto viável (busca linear de Armijo).

- **Atualizações**

- As rotinas de atualização dos multiplicadores de Lagrange tanto para as restrições de desigualdade quanto para as matriciais é descrita em [23];
- Finalmente, na atualização da matriz quase-Newton empregou-se o método Broyden-Fletcher-Goldfarb-Shanno (BFGS) descrito em [5].

## 5.3 Escolha do Código

Como já foi citado, a proposta inicial do algoritmo FDIPA foi idealizada por Herskovits e, desde então, encontra-se em constante aperfeiçoamento, seja para adaptá-lo à resolução de novos problemas ou mesmo para melhorar o seu desempenho em termos de robustez, eficiência e confiabilidade.

Neste sentido, a implementação do seu código para realização de testes numéricos também acompanha a evolução do algoritmo. E, por esta razão, decidiu-se aqui trabalhar com uma codificação já existente, denominada “fdipamat\_sdp”, desenvolvida em MATLAB, e descrita nos moldes de [22].

A intenção foi facilitar a alteração sugerida nesta dissertação, uma vez que o código aproveitado já disponibiliza as modificações necessárias para lidar com problemas de programação semidefinida, além de possuir rotinas para a resolução dos sistemas do FDIPA-GSDP(1) e FDIPA-GSDP(2), ambos utilizando o produto de Kronecker, ao invés do produto simétrico de Kronecker, como disponibilizado em outras versões [9].

Apesar destas facilidades iniciais, muitas dificuldades foram encontradas durante a manipulação do código, principalmente relacionadas à sua sensibilidade a alguns parâmetros de controle, são eles:  $\eta \in (0, 1)$  e  $\xi \in (0, 1)$ , utilizados para controlar o tamanho do passo inicial na busca linear de Armijo e para o cálculo da deflexão na direção de busca, respectivamente.

Observou-se que sutis mudanças nos valores destas variáveis ocasionavam enormes alterações na quantidade de iterações necessária para a convergência do algoritmo. Este fator foi primordial para decidir que a cada exemplo um valor distinto para essas incógnitas fosse configurado, como pode ser visto nas seções posteriores.

Mesmo com os contratempos advindos da codificação disponível, foi adicionado a este código o procedimento utilizado para a resolução do sistema linear (4.7), objetivo desta dissertação, imprescindível para executar o Passo 1 do algoritmo descrito na Seção 4.3. Para o cálculo das matrizes de rigidez e de massa também foram desenvolvidas rotinas em MATLAB.

Todos os exemplos foram executados em um computador com as seguintes características técnicas: processador Intel Core i5 de 2.3 GHz, 8 GB de memória RAM e sistema operacional Windows 8.1 de 64 bits.

## 5.4 Resultados dos Testes

Para validar o algoritmo implementado neste trabalho, decidiu-se testar três exemplos com níveis de dificuldade distintas, principalmente no que diz respeito ao número de elementos finitos da discretização em cada uma das treliças. No primeiro,

através de uma estrutura pequena, explorou-se mais as condições de convergência do FDIPA-GSDP(3), enquanto que no segundo e terceiro analisou-se sua capacidade em trabalhar com estruturas que possuem um número maior de barras (estruturas de médio e grande portes).

Por tratar-se de exemplos acadêmicos, preferiu-se suprimir as unidades de medidas correspondentes às variáveis presentes nestes testes.

Para os três exemplos apresentados nesta seção, considerou-se, sobre as estruturas mecânicas utilizadas, as informações mostradas na Tabela 5.3.

Variável	Valor	Descrição
$Area$	0,01	Área inicial da seção transversal de cada barra
$\rho^{mat}$	1	Massa específica de cada barra
$Intensidade$	1	Módulo da intensidade da força em um nó
$\bar{\gamma}$	1	Complacência máxima admissível

Tabela 5.3: Notação empregada para definir parâmetros das estruturas mecânicas em todos os exemplos.

#### 5.4.1 Exemplo 1 - Treliça Bidimensional de 26 Barras

Para este primeiro exemplo optou-se por testar uma treliça quadrada, no formato  $3 \times 3$ , com todos os nós conectados por barras (resultando em um total de 26), cuja geometria é mostrada na Figura 5.1. A estrutura está submetida a uma força ( $p$ ) de intensidade igual a  $(-1, 0)$  exercida no nó 6 e as condições de contorno estabelecem que os nós 1, 4 e 7 estão fixos nas direções  $x$  e  $y$ . Portanto, tem-se um total de 18 graus de liberdade ( $9$  nós  $\times$   $2$  graus por nó), sendo  $6$  ( $3 \times 2$ ) deles fixos, chegando-se, assim, a um número reduzido de  $12$  graus ( $18 - 6 = 12$ ).

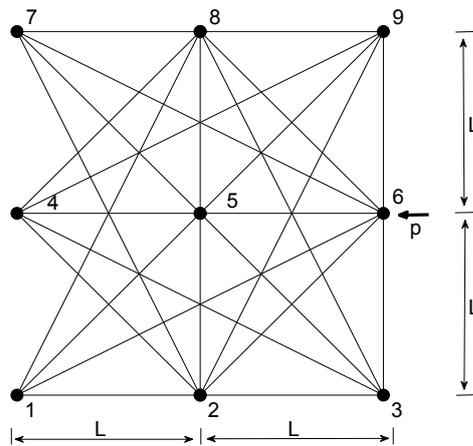


Figura 5.1: Geometria da treliça de 26 barras.

Na primeira execução, fixou-se os valores:  $\bar{V} = 0,406827$  (volume máximo da treliça),  $E = 10$  (módulo de Young de cada barra - elasticidade),  $\xi = 0,98$  e  $\eta = 0,5$ .

O número de variáveis de projeto e o número de restrições vetoriais e matriciais impostas neste problema são mostradas na Tabela 5.4.

$n$	$m$	$q$	$nc$
27	1	25	26

Tabela 5.4: Dimensão do problema para a treliça de 26 barras.

O teste inicial foi realizado com o mesmo ponto inicial viável para todas as versões, alcançando-se os resultados descritos na Tabela 5.5.

Algoritmo	$\lambda$	$NI$	$TE$
FDIPA-GSDP(1)	0,0502363	13	0,449787
FDIPA-GSDP(2)	0,0502547	13	0,145446
FDIPA-GSDP(3)	0,0502547	13	0,133480

Tabela 5.5: Exemplo 1. Execução com o mesmo ponto inicial viável.

Testou-se também o caso onde cada algoritmo encontra seu próprio ponto inicial viável, exibido com detalhes na Tabela 5.6.

Algoritmo	$\lambda$	$NI_{pv}$	$NI$	$TE$
FDIPA-GSDP(1)	0,0502477	25	12	1,248459
FDIPA-GSDP(2)	0,0501032	26	13	0,564037
FDIPA-GSDP(3)	0,0501022	26	13	0,425971

Tabela 5.6: Exemplo 1. Execução com cada algoritmo buscando seu próprio ponto inicial viável.

Percebe-se, a partir destes números, que as três versões do algoritmo resolvem de forma satisfatória o problema proposto, já que as soluções obtidas são praticamente iguais, bem como o número de iterações necessárias para atingir o valor ótimo. Cabe destacar que, para este exemplo, o algoritmo proposto neste trabalho foi mais eficiente em todas as execuções, pois sempre alcançou o objetivo em um intervalo de tempo menor que o das outras formulações.

No sentido de comprovar a sensibilidade do código aos parâmetros  $\eta$  e  $\xi$ , submeteu-se os algoritmos a uma nova execução, conforme a Tabela 5.7, porém, agora fixando os respectivos valores para estas variáveis:  $0,5 \times 10^{-3}$  e  $0,8$ .

Algoritmo	$\lambda$	$NI_{pv}$	$NI$	$TE$
FDIPA-GSDP(1)	0,0503397	61	32	3,079049
FDIPA-GSDP(2)	0,0503507	72	26	1,197396
FDIPA-GSDP(3)	0,0503450	72	24	1,029694

Tabela 5.7: Exemplo 1. Comprovação da sensibilidade do código aos parâmetros  $\eta$  e  $\xi$ .

Vê-se aqui, agora com resultados comprobatórios, a grande sensibilidade do algoritmo às variáveis supracitadas, uma vez que, além do FDIPA-GSDP(3) ter sido, novamente, destaque no quesito eficiência, o número de iterações (tanto para obter

o ponto inicial viável quanto para otimizar) e o tempo de execução aumentaram significativamente, tendo, o FDIPA-GSDP(1), o pior desempenho.

Dando maior ênfase ao FDIPA-GSDP(3), a Figura 5.2 detalha sua convergência, mostrando a redução da função objetivo obtida de acordo com os valores expostos na Tabela 5.5. Foram necessárias 13 iterações e um tempo de aproximadamente 0,13 segundos para encontrar a solução ótima.

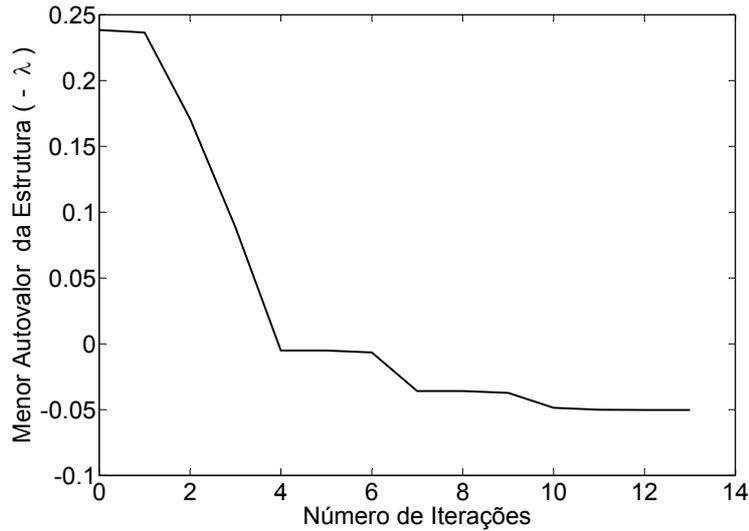


Figura 5.2: Convergência do FDIPA-GSDP(3) para a treliça de 26 barras.

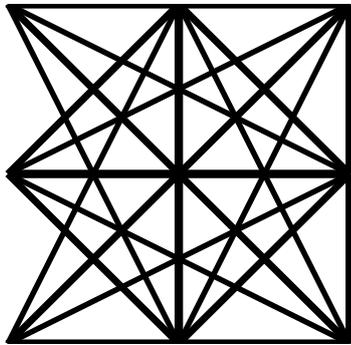
Ainda para ilustrar o desempenho do FDIPA-GSDP(3), a Tabela 5.8 evidencia os valores iniciais e finais das variáveis de projeto ( $x$  e  $\lambda$ ), do volume da estrutura e sua complacência. Para realçar ainda mais os resultados, inicializou-se o volume de todas as barras com o valor 1, totalizando 26 (volume da estrutura).

Variáveis do Problema	Configuração Inicial	Configuração Final
$x_1$	1	0,0000000
$x_2$	1	0,0000000
$x_3$	1	0,0000000
$x_4$	1	0,0000000
$x_5$	1	0,1955413
$x_6$	1	0,1964342
$x_7$	1	0,0000000
$x_8$	1	0,0000000
$x_9$	1	0,0000000
$x_{10}$	1	0,0000000
$x_{11}$	1	0,0018977
$x_{12}$	1	0,0000000
$x_{13}$	1	0,0000000
$x_{14}$	1	0,0000000

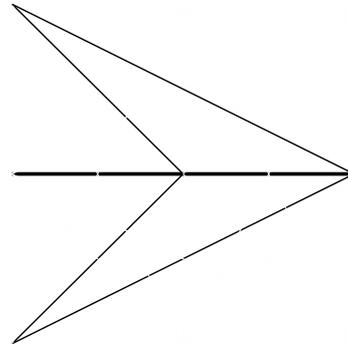
$x_{15}$	1	0,0000000
$x_{16}$	1	0,0018983
$x_{17}$	1	0,0000000
$x_{18}$	1	0,0000000
$x_{19}$	1	0,0055280
$x_{20}$	1	0,0000000
$x_{21}$	1	0,0000000
$x_{22}$	1	0,0000000
$x_{23}$	1	0,0000000
$x_{24}$	1	0,0000000
$x_{25}$	1	0,0000000
$x_{26}$	1	0,0055275
$\lambda$	-0,2384552	0,0502547
$\bar{\gamma}$	1	1
$\bar{V}$	0,4068270	0,4068270

Tabela 5.8: Estrutura ótima obtida para a treliça de 26 barras.

Nota-se que, após a otimização, de um total de 26 barras, 20 tenderam para um volume final zero, modificando e muito a geometria da treliça, como pode ser visto na Figura 5.3. Consequentemente, o volume final da treliça também foi, substancialmente, reduzido.



(a) Configuração Inicial



(b) Configuração Final

Figura 5.3: Configurações inicial e final para a treliça de 26 barras.

Conhecendo essas informações, pode-se concluir que para que o menor autovalor da estrutura (igual ao quadrado da frequência natural,  $\lambda = w^2$ ) seja de aproximadamente 0,05, ao aplicar uma força de intensidade  $(-1,0)$  no nó 6, nem todas as barras presentes na geometria inicial da treliça são necessárias.

Finalizando este exemplo, a Tabela A.2, do Apêndice A, mostra a redução da função objetivo, o histórico das condições de complementariedade das restrições de desigualdade e matricial, valores empregados como critérios de parada e dados referentes à busca linear de Armijo, obtidos com o FDIPA-GSDP(3).

### 5.4.2 Exemplo 2 - Treliça Bidimensional de 534 Barras

Neste exemplo testou-se uma treliça retangular de porte médio, no formato  $11 \times 5$ , com 534 barras, cuja geometria é mostrada na Figura 5.4. A estrutura está submetida a uma força de intensidade igual a  $(-1, 0)$  exercida no nó 33 e as condições de contorno estabelecem que os nós 1, 12, 23, 34 e 45 estão fixos nas direções  $x$  e  $y$ . Portanto, tem-se um total de 110 graus de liberdade ( $55$  nós  $\times$   $2$  graus por nó), sendo  $10$  ( $5 \times 2$ ) deles fixos, chegando-se, assim, a um número reduzido de  $100$  graus ( $110 - 10 = 100$ ).

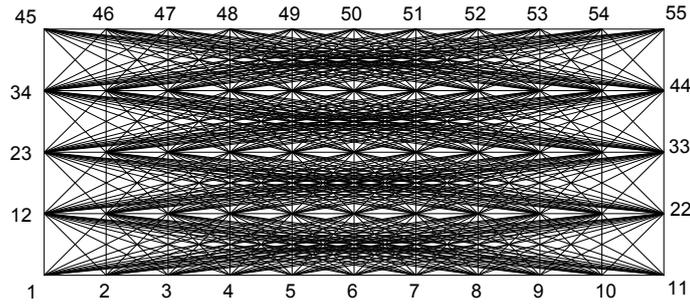


Figura 5.4: Geometria da treliça de 534 barras.

Para a realização deste teste, fixou-se, para os parâmetros da busca linear e da deflexão, os mesmos valores da primeira execução do exemplo anterior, já para  $\bar{V}$  e  $E$  trabalhou-se com  $5,033$  e  $100$ , respectivamente.

A dimensão do problema pode ser vista na Tabela 5.9, onde é possível perceber valores bem maiores que os do Exemplo 1, com exceção da quantidade de restrições de desigualdade, que permanece a mesma.

$n$	$m$	$q$	$nc$
535	1	201	534

Tabela 5.9: Dimensão do problema para a treliça de 534 barras.

A partir dos valores descritos acima e partindo do mesmo ponto inicial viável, os resultados do teste são apresentados na Tabela 5.10. Destaca-se que, conforme [21], o FDIPA-GSDP(1) só é capaz de trabalhar com, no máximo, 300 elementos e, por conta desta limitação, esta versão não foi avaliada.

Algoritmo	$\lambda$	$NI$	$TE$
FDIPA-GSDP(1)	n/a	n/a	n/a
FDIPA-GSDP(2)	0,883149	135	2923,19
FDIPA-GSDP(3)	0,883285	135	2918,84

Nota: n/a = não avaliado.

Tabela 5.10: Exemplo 2. Execução com o mesmo ponto inicial viável.

Mais uma vez, é perceptível que as duas versões avaliadas resolvem de forma satisfatória o problema proposto, pois tanto as soluções obtidas quanto o número de iterações necessárias para atingir o valor ótimo são muito próximos. De forma análoga ao primeiro exemplo, o algoritmo desenvolvido nesta dissertação foi mais eficiente, já que, novamente, alcançou o objetivo em um intervalo de tempo menor que o FDIPA-GSDP(2).

A Figura 5.5 detalha a convergência do FDIPA-GSDP(3), mostrando a redução da função objetivo obtida de acordo com os valores expostos na Tabela 5.10. Foram necessárias 135 iterações e um tempo de aproximadamente  $2,9e+03$  segundos para encontrar a solução ótima.

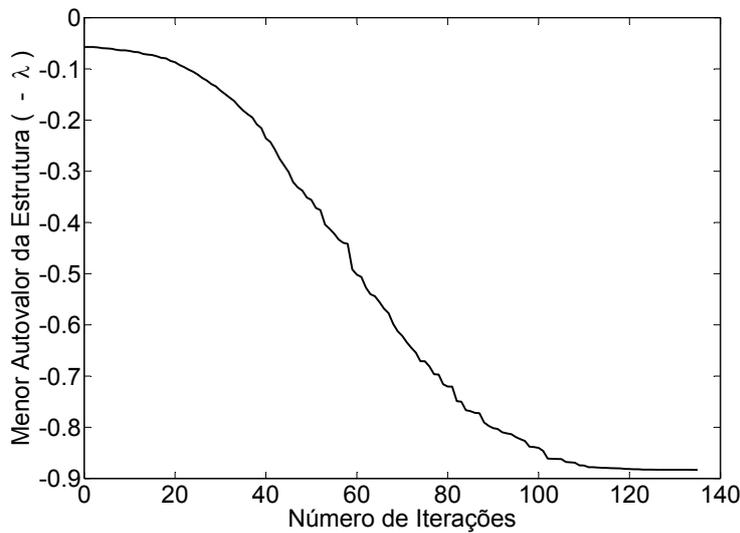


Figura 5.5: Convergência do FDIPA-GSDP(3) para a treliça de 534 barras.

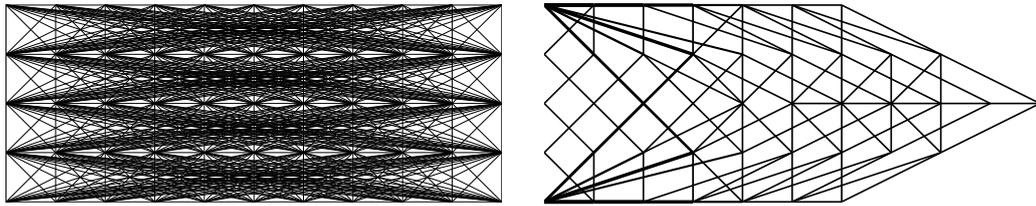
Em relação ao Exemplo 1, o tempo gasto para otimizar a estrutura de 534 barras foi altamente superior, realizando a conversão, pouco mais de 48 minutos. Este fato deve-se principalmente pelo processamento, em cada iteração, da derivada da matriz de restrições ( $\nabla A(x) \in \mathbb{R}^{n \times q^2}$ ), que, associando a este exemplo, possui uma dimensão de  $535 \times 40401$ , bem como da matriz  $U \in \mathbb{R}^{q^2 \times n}$ , calculada em (3.36) e que também carrega, implicitamente, os valores da matriz  $\nabla A$ .

Para ilustrar o desempenho do FDIPA-GSDP(3), a Tabela 5.11 evidencia os valores iniciais e finais do menor autovalor ( $\lambda$ ) e das restrições de volume e complacência.

Variáveis do Problema	Configuração Inicial	Configuração Final
$\lambda$	0,0572193	0,883285
$\bar{\gamma}$	1	0,999982
$\bar{V}$	5,033000	5,033000

Tabela 5.11: Estrutura ótima obtida para a treliça de 534 barras

Após o processo de otimização, a geometria da treliça pode ser visualizada na Figura 5.6, onde é feita uma comparação com a sua forma inicial.



(a) Configuração Inicial

(b) Configuração Final

Figura 5.6: Configurações inicial e final para a treliça de 534 barras.

Para mais detalhes sobre este exemplo, a Tabela A.3, do Apêndice A, mostra a redução da função objetivo, o histórico das condições de complementariedade das restrições de desigualdade e matricial, valores empregados como critérios de parada e dados referentes à busca linear de Armijo, obtidos com o FDIPA-GSDP(3).

### 5.4.3 Exemplo 3 - Treliça Bidimensional de 1580 Barras

Com o intuito de ratificar que o FDIPA-GSDP(3) é capaz de resolver problemas de grande porte, neste último exemplo foi testada uma treliça retangular semelhante ao do Exemplo 2, porém maior, no formato  $13 \times 5$  e com 1580 barras, cuja geometria é apresentada na Figura 5.7. A estrutura está submetida a uma força de intensidade igual a  $(-1, 0)$  exercida no nó 39 e as condições de contorno estabelecem que os nós 1, 14, 27, 40 e 53 estão fixos nas direções  $x$  e  $y$ . Portanto, tem-se um total de 130 graus de liberdade ( $65$  nós  $\times$   $2$  graus por nó), sendo 10 ( $5 \times 2$ ) deles fixos, chegando-se, assim, a um número reduzido de 120 graus ( $130 - 10 = 120$ ).

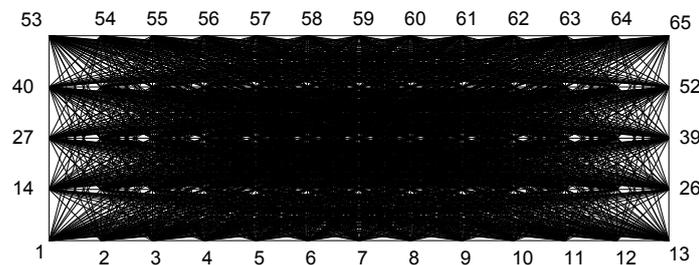


Figura 5.7: Geometria da treliça de 1580 barras.

Para a execução deste teste, fixou-se, para as variáveis indicativas da busca linear, da deflexão e do módulo de elasticidade, os mesmos valores do exemplo anterior, já para o volume máximo admissível atribuiu-se  $\bar{V} = 10,5$ .

A dimensão deste exemplo, vista na Tabela 5.12, é bem maior que a dos anteriores, justificando a classificação na categoria de grande porte.

$n$	$m$	$q$	$nc$
1581	1	241	1580

Tabela 5.12: Dimensão do problema para a treliça de 1580 barras.

Configurados os parâmetros destacados e a partir do mesmo ponto inicial viável, os resultados do teste são apresentados na Tabela 5.13. Mais uma vez, cumpre destacar que, conforme [21], em razão da dimensão deste exemplo superar o limite máximo de processamento de elementos finitos do FDIPA-GSDP(1), 300 elementos, esta versão não foi avaliada. E, embora mencionado pelo autor supracitado que em problemas de otimização de material o FDIPA-GSDP(2) limitava-se a executar testes com até 1200 elementos, nota-se que, para o problema em questão, este algoritmo também foi capaz de encontrar a solução ótima.

Algoritmo	$\lambda$	$NI$	$TE$
FDIPA-GSDP(1)	n/a	n/a	n/a
FDIPA-GSDP(2)	0,674378	274	6,6305e+04
FDIPA-GSDP(3)	0,674378	274	6,6301e+04

Nota: n/a = não avaliado.

Tabela 5.13: Exemplo 3. Execução com o mesmo ponto inicial viável.

Observa-se aqui que a nova formulação foi a mais eficiente na resolução do problema de grade porte apresentado, cumprindo com o objetivo deste trabalho. A considerável redução na matriz de Newton é responsável por aumentar a capacidade de armazenamento em memória, permitindo o processamento de estruturas com grandes quantidades de elementos finitos.

A Figura 5.8 detalha a convergência do FDIPA-GSDP(3), mostrando a redução da função objetivo obtida de acordo com os valores expostos na Tabela 5.13.

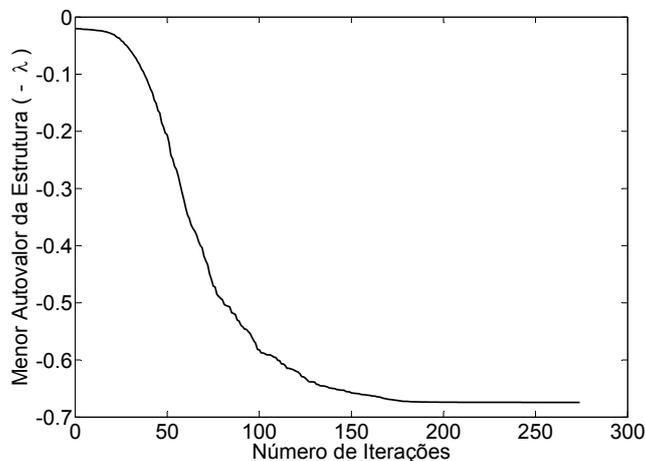


Figura 5.8: Convergência do FDIPA-GSDP(3) para a treliça de 1580 barras.

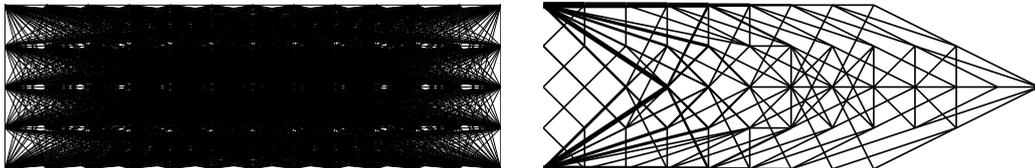
Como é nítido, uma consequência da otimização estrutural de grande porte é o enorme tempo gasto neste processo, que, como já foi detalhado, deve-se principalmente pela necessidade de calcular, no sistema de Newton, a derivada das restrições matriciais, que, aqui, possui uma dimensão de  $1581 \times 58081$ . Para este exemplo, o otimizador precisou de aproximadamente 18,4 horas (274 iterações em  $6,6301e+04$  segundos) para atingir a solução ótima, cerca de pouco mais de 4 minutos por iteração. Ainda assim, o desempenho desta versão é melhor que o do FDIPA-GSDP(2), já que este executou o problema em um tempo superior, como exibido na Tabela 5.13.

Para ilustrar o desempenho do FDIPA-GSDP(3), a Tabela 5.14 evidencia os valores iniciais e finais do menor autovalor ( $\lambda$ ) e das restrições de volume e complacência.

Variáveis do Problema	Configuração Inicial	Configuração Final
$\lambda$	0,0204494	0,674378
$\bar{\gamma}$	1	0,999753
$\bar{V}$	10,5000	10,5000

Tabela 5.14: Estrutura ótima obtida para a treliça de 1580 barras

Após o processo de otimização, a geometria da treliça pode ser visualizada na Figura 5.9, onde é feita uma comparação com a sua forma inicial.



(a) Configuração Inicial

(b) Configuração Final

Figura 5.9: Configurações inicial e final para a treliça de 1580 barras.

Para finalizar os testes, na Tabela A.4, do Apêndice A, fica evidente a redução da função objetivo, o histórico das condições de complementariedade das restrições de desigualdade e matricial, valores empregados como critérios de parada e dados referentes à busca linear de Armijo na solução deste exemplo teste, obtidos com o FDIPA-GSDP(3).

# Capítulo 6

## Considerações Finais

### 6.1 Conclusão

Baseado em técnicas de programação semidefinida não linear, desenvolveu-se, nesta dissertação, um novo algoritmo de pontos interiores por direções viáveis, o FDIPA-GSDP(3), cujo objetivo é resolver problemas de otimização estrutural de grande porte. Seu diferencial está na formulação de um novo sistema de Newton, cuja função é encontrar uma direção que seja ao mesmo tempo de descida e viável, com dimensões bem reduzidas em relação ao das versões já existentes, o que facilita o seu armazenamento em memória e torna possível sua aplicação em problemas que requerem um grande número de elementos finitos na discretização da estrutura.

Visando criar um ambiente de validação para o algoritmo implementado, ele foi aplicado na resolução de um problema de otimização clássico amplamente conhecido na literatura da engenharia mecânica: a maximização da frequência natural de estruturas sujeito ao equilíbrio de condições e restrições de volume e complacência, culminando na construção de corpos que, frente ações externas periódicas, não venham a ter oscilações com grandes amplitudes, evitando que sofram deformações que comprometam sua integridade devido ao fenômeno da ressonância mecânica.

Pelos resultados numéricos obtidos, comprovou-se a eficiência do FDIPA-GSDP(3) frente a outras versões de códigos da família FDIPA, já que alcançou valores bem melhores tanto em relação ao número de iterações quanto ao tempo gasto para atingir a solução ótima. Nos testes a que foi submetido, destaca-se a capacidade de resolver problemas estruturais discretizados em até 1580 elementos, necessitando, para isso, de  $6,6301e+04$  segundos (aproximadamente 18,4 horas), desempenho melhor que o conquistado pelo, até então, mais bem sucedido algoritmo desta família, o FDIPA-GSDP(2), que processou o mesmo teste em um tempo superior, cerca de  $6,6305e+04$  segundos.

Apesar de constatada a robustez do novo otimizador, algumas dificuldades ainda

o impedem de melhorar ainda mais o seu rendimento. Pelos dados analisados, o maior obstáculo é o cálculo da matriz  $U$ , que necessita diretamente da derivada das restrições matriciais,  $\nabla A$ .

Diante do exposto, resta-se comprovada a importância do algoritmo FDIPA-GSDP(3) para resolução de problemas estruturais de grande porte, pois a nova implementação cumpre com todos os requisitos das técnicas de programação semi-definida bem como com os objetivos desta pesquisa.

## 6.2 Sugestões para Trabalhos Futuros

Como parte dos estudos futuros, destacam-se os seguintes tópicos:

- Resolver o sistema linear do FDIPA-GSDP(3), (4.7), utilizando o Método dos Gradientes Conjugados;
- Provar a convergência global do FDIPA-GSDP(3);
- Estudar possíveis técnicas para melhorar o cálculo da matriz  $U$ ;
- Explorar outros problemas que possam ser resolvidos com o FDIPA-GSDP(3), principalmente aqueles que possuem mais de uma restrição de desigualdade;
- Pesquisar sobre a discretização ótima de estruturas mecânicas;
- Trabalhar com estruturas mecânicas 3D.

# Referências Bibliográficas

- [1] ARORA, J. S. *Introduction to optimum design*. Academic Press, 2004.
- [2] MASSERA, J. M. A. *Uma nova técnica para otimização de estruturas de grande porte*. Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2006.
- [3] RAO, S. S. *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.
- [4] FÁLCON, G. A. S. *Uma Família de Algoritmos de Ponto Interior para Projeto Ótimo em Engenharia*. Tese de Doutorado, COPPE/PEM-Universidade Federal do Rio de Janeiro, 1996.
- [5] LUENBERGER, D. G. *Linear and Nonlinear Programming*. 2 ed. , Addison Wesley, 1984.
- [6] VANDENBERGHE, L., BOYD, S. “Semidefinite Programming”, *SIAM*, v. 38, mar 1996.
- [7] MASSERA, J. M. A. *Técnicas de Programação Semidefinida e Aplicações em Otimização de Material*. Tese de doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2010.
- [8] HERSKOVITS, J., SANTOS, G. *A Feasible arc interior point algorithms for nonlinear optimization*. Computational Mechanics and New trends and Applications hal-00758803, CIMNE, Barcelona, Spain, 2014.
- [9] AROZTEGUI, M., HERSKOVITS, J., ROCHE, J. R., et al. “A feasible direction interior point algorithm for nonlinear semidefinite programming.” *Structural and Multidisciplinary Optimization*, nov 2014.
- [10] HERSKOVITS, J. “An interior point technique for nonlinear optimization”, *Rapports de recherche- INRIA*, 1992.
- [11] DE KLERK, E. *Aspects of semidefinite programming: interior point algorithms and selected applications*, v. 65. Springer Science & Business Media, 2006.

- [12] CHONG, E. K., ZAK, S. H. *An introduction to optimization*, v. 76. John Wiley & Sons, 2013.
- [13] HORN, R. A., JOHNSON, C. R. *Matrix analysis*. Cambridge University Press, 2012.
- [14] GOLUB, G. H., VAN LOAN, C. F. *Matrix computations*, v. 3. JHU Press, 2012.
- [15] ACHTZIGER, W., KOČVARA, M. “Structural topology optimization with eigenvalues”, *SIAM Journal on Optimization*, v. 18, n. 4, pp. 1129–1164, 2007.
- [16] HERSKOVITS, J. “Feasible direction interior-point technique for nonlinear optimization”, *Journal of optimization theory and applications*, v. 99, n. 1, pp. 121–146, 1998.
- [17] SHAPIRO, A. “On uniqueness of Lagrange multipliers in optimization problems subject to cone constraints”, *SIAM Journal on Optimization*, v. 7, n. 2, pp. 508–518, 1997.
- [18] RODRIGUES, R. D. O. *Análise dinâmica bidimensional não-linear física e geométrica de treliças de aço e pórticos de concreto armado*. Tese de Doutorado, Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos, SP, Brasil, 1997.
- [19] ACHTZIGER, W., KOČVARA, M. “On the maximization of the fundamental eigenvalue in topology optimization”, *Structural and Multidisciplinary Optimization*, v. 34, n. 3, pp. 181–195, 2007.
- [20] BARROS, L. N. *Técnica de programação semidefinida por arco viável e aplicação à maximização da frequência natural de estruturas mecânicas*. Tese de Mestrado, UFPB, João Pessoa, PB, Brasil, 2017.
- [21] RESTREPO, A. G. Z. *Otimização de Material de Estruturas Elásticas Mediante uma Técnica de Programação Semidefinida*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2016.
- [22] ROCHE, J. R., HERSKOVITS, J., BAZÁN, E., et al. “A feasible direction algorithm for general nonlinear semidefinite programming”, *Structural and Multidisciplinary Optimization*, v. 4, n. 55, pp. 1261–1279, 2016.
- [23] HERSKOVITS, J., GOULART, E., AROZTEGUI, M. “Optimization of Large Scale Systems”, *Optimization of Structural and Mechanical Systems. Sin-gapura: World Scientific*, pp. 35–57, 2007.

# Apêndice A

## Informações Gerais sobre as Iterações Obtidas pelo FDIPA-GSDP(3)

As seções a seguir mostram tabelas que contêm detalhes das iterações obtidas com o FDIPA-GSDP(3) em cada exemplo teste. Nelas é possível observar a redução da função objetivo, o histórico das condições de complementariedade das restrições de desigualdade e matricial, valores empregados como critérios de parada e dados referentes à busca linear de Armijo. A notação exibida abaixo foi empregada para descrever estes detalhes:

Notação	Descrição
$iter$	Iteração $k$
$f$	Valor da função objetivo avaliada no ponto $x^k$
$\ d_0\ $	Norma da direção de busca avaliada no ponto $x^k$
$\ \nabla L\ $	Norma do gradiente da função Lagrangiana avaliada no ponto $x^k$
$\ G\gamma_0\ $	Norma da condição de complementariedade das restrições de desigualdade avaliada no ponto $x^k$
$\ tr(\Lambda_0 A)\ $	Valor absoluto do traço da condição de complementariedade das restrições matriciais avaliada no ponto $x^k$
$t$	Comprimento do passo $t^k$ da busca linear de Armijo
$itlin$	Número de iterações na busca linear de Armijo

Tabela A.1: Notação empregada para definir detalhes da execução do algoritmo.

### A.1 Tabela de Iterações - Treliça de 26 Barras

$iter$	$f$	$\ d_0\ $	$\ \nabla L\ $	$\ G\gamma_0\ $	$\ tr(\Lambda_0 A)\ $	$t$	$itlin$
0	0,2384550						
1	0,2365180	0,5665400	0,5665400	0,0002497	0,2307410	0,0001381	17
2	0,1709880	0,4981200	0,2115260	0,0001976	0,3076530	0,1328990	3
3	0,0884428	0,1641780	0,0148635	0,0001496	0,1683010	0,5079060	1
4	-0,0051009	0,1184520	0,0021313	0,0002290	0,0746423	0,7977310	1
5	-0,0051040	0,0372728	0,0041978	0,0002078	0,0645333	8,67175e-05	1

6	-0,0066211	0,0354945	0,0008726	0,0001991	0,0650637	0,0435292	1
7	-0,0358802	0,0403327	0,0004503	0,0001176	0,0221166	0,7411340	1
8	-0,0358857	0,0145921	0,0002743	0,0001161	0,0142333	0,0003828	1
9	-0,0372524	0,0143883	6,2918e-05	0,0001076	0,0127797	0,0974975	1
10	-0,0485660	0,0130217	3,0568e-05	3,6591e-05	0,0017550	0,8917840	1
11	-0,0501125	0,0017187	1,5637e-05	6,1267e-06	0,0001275	0,9260190	1
12	-0,0502512	0,0001453	1,1321e-06	2,6850e-07	2,8257e-06	0,9808910	1
13	-0,0502547	3,7403e-06	9,5157e-08	1,6554e-09	1,9089e-08	0,9942190	1

Tabela A.2: Exemplo 1. Redução da função objetivo, histórico das condições de complementariedade e critérios de parada obtidos pelo FDIPA-GSDP(3).

## A.2 Tabela de Iterações - Treliça de 534 Barras

De acordo com a Tabela A.3, pode-se observar que, neste exemplo, o número de iterações para encontrar a solução ótima foi menor que o número de variáveis de projeto ( $n$ ) que, neste exemplo, é igual a 535.

<i>iter</i>	<i>f</i>	$\ d_0\ $	$\ \nabla L\ $	$\ G_{\gamma_0}\ $	$\ tr(\Lambda_0 A)\ $	<i>t</i>	<i>itlin</i>
0	-0,0572193						
1	-0,0572301	0,0001356	0,0001356	1,1276e-09	0,0052356	0,0823543	8
2	-0,0572483	0,0133669	0,0132012	1,2718e-08	0,0211214	0,0138413	13
3	-0,0580585	0,0117777	0,0059887	1,6814e-05	0,0089825	1	1
4	-0,0593160	0,0220082	0,0057217	1,8675e-08	0,0065084	0,9651700	1
5	-0,0598223	0,0274758	0,0157335	2,4734e-08	0,0046704	0,1747180	1
6	-0,0606569	0,0330983	0,0131456	2,9325e-08	0,0056605	0,2028670	1
7	-0,0623558	0,0306051	0,0103673	4,2077e-08	0,0060895	0,4315310	1
8	-0,0635450	0,0517642	0,0178833	8,8193e-09	0,0049472	0,1652740	1
9	-0,0635806	0,0488422	0,0123783	3,2755e-08	0,0038948	0,0046018	1
10	-0,0647350	0,0753807	0,0148813	1,8197e-07	0,0054172	0,1056030	1
11	-0,0666973	0,0579597	0,0099632	2,0342e-07	0,0064453	0,2086860	1
12	-0,0673996	0,1069460	0,0190904	5,1200e-07	0,0053886	0,0473288	1
13	-0,0707263	0,0885048	0,0167963	6,2646e-07	0,0078789	0,2308330	1
14	-0,0719138	0,1136150	0,0187278	1,9846e-06	0,0051032	0,0706305	1
15	-0,0728605	0,1407160	0,0271133	9,8101e-06	0,0109037	0,0405327	2
16	-0,0753457	0,1306540	0,0237681	1,1748e-05	0,0158667	0,1125560	1
17	-0,0783322	0,1996720	0,0317368	1,0376e-05	0,0152144	0,0962286	1
18	-0,0793429	0,1540100	0,0441715	9,8022e-06	0,0162113	0,0355133	1
19	-0,0841766	0,2798520	0,0357604	1,1681e-05	0,0193327	0,1097680	1
20	-0,0869045	0,2500770	0,0361074	9,2899e-06	0,0143755	0,0639216	1
21	-0,0922473	0,2480650	0,0527569	1,0528e-05	0,0214122	0,1314860	1
22	-0,0963458	0,2513660	0,0382157	8,6063e-06	0,0174421	0,0923478	1
23	-0,1012440	0,2438160	0,0689646	1,6051e-05	0,0542822	0,0979997	1
24	-0,1057000	0,2306840	0,0511178	1,3640e-05	0,0466592	0,0777983	1

25	-0,1110310	0,2889460	0,0700970	1,6029e-05	0,0634903	0,0822306	1
26	-0,1177520	0,2160190	0,0474381	1,0674e-05	0,0399378	0,1257370	1
27	-0,1226840	0,3022040	0,0589453	1,2425e-05	0,0556505	0,0778873	1
28	-0,1294790	0,2357390	0,0413436	9,4751e-06	0,0410554	0,1215240	1
29	-0,1340240	0,2552370	0,0688039	1,2222e-05	0,0673805	0,0758059	1
30	-0,1423090	0,2652230	0,0496535	9,7976e-06	0,0536438	0,1288460	1
31	-0,1488620	0,3535070	0,0559341	9,9557e-06	0,0617169	0,0910009	1
32	-0,1558140	0,3337290	0,0529937	8,8377e-06	0,0570100	0,0987377	1
33	-0,1624620	0,3960620	0,0404296	9,1165e-06	0,0653581	0,0873470	1
34	-0,1724270	0,2801890	0,0666192	9,2964e-06	0,0758166	0,1404280	1
35	-0,1814570	0,2740950	0,0700401	1,0129e-05	0,0967339	0,1098020	1
36	-0,1884860	0,3242530	0,0722336	9,9162e-06	0,1008110	0,0715360	1
37	-0,1946880	0,3586290	0,0617851	9,2379e-06	0,0971321	0,0654651	1
38	-0,2081190	0,4451390	0,0644379	8,6164e-06	0,0931997	0,1302140	1
39	-0,2155820	0,5493380	0,0708630	9,1239e-06	0,1098760	0,0605461	1
40	-0,2355320	0,4113700	0,1253990	9,3128e-06	0,1219360	0,1539940	1
41	-0,2431860	0,4259090	0,0615495	8,6691e-06	0,1213640	0,0591633	1
42	-0,2579440	0,2810290	0,0655390	8,3972e-06	0,1283480	0,1423090	1
43	-0,2754020	0,4374830	0,0881821	8,6538e-06	0,1435640	0,1106060	1
44	-0,2884600	0,2654120	0,0483392	7,0264e-06	0,1230960	0,1441800	1
45	-0,3012800	0,2281610	0,0708362	6,7074e-06	0,1349890	0,1418560	1
46	-0,3211410	0,5553250	0,0862878	7,1178e-06	0,1518200	0,1077810	1
47	-0,3315610	0,1323470	0,0288802	4,7707e-06	0,1124010	0,2253150	1
48	-0,3377400	0,2086310	0,0188681	4,3869e-06	0,1160790	0,1134560	1
49	-0,3511030	0,4497270	0,0265896	4,5902e-06	0,1332210	0,1297770	1
50	-0,3558490	0,3028160	0,0326911	4,7742e-06	0,1504400	0,0421219	1
51	-0,3716020	0,2339070	0,0335332	4,6184e-06	0,1535250	0,1222160	1
52	-0,3757330	0,2448390	0,0520617	4,7621e-06	0,1700280	0,0277607	1
53	-0,4040010	0,2420450	0,0400437	4,2637e-06	0,1501810	0,1922050	1
54	-0,4122040	0,3593880	0,0937570	4,5543e-06	0,1820170	0,0383577	1
55	-0,4214950	0,1697670	0,0403830	4,0003e-06	0,1554910	0,0881464	1
56	-0,4330920	0,2512940	0,0329121	3,8450e-06	0,1564490	0,0919546	1
57	-0,4395100	0,4612810	0,0448166	4,0092e-06	0,1725850	0,0354657	1
58	-0,4416360	0,1693650	0,0227555	3,6367e-06	0,1497270	0,0283207	1
59	-0,4916210	0,2173490	0,0152589	1,9614e-06	0,1083430	0,6073130	1
60	-0,5018710	0,4994850	0,0276102	2,1642e-06	0,1433840	0,0609797	1
61	-0,5064860	0,1580020	0,0134528	1,9518e-06	0,1256530	0,0757584	1
62	-0,5271260	0,2593130	0,0141777	1,7381e-06	0,1154230	0,2227130	1
63	-0,5400150	0,3876100	0,0344119	1,8842e-06	0,1456450	0,0730718	1
64	-0,5441630	0,0846089	0,0188611	1,7516e-06	0,1351730	0,0583501	1
65	-0,5557420	0,1066770	0,0174276	1,6232e-06	0,1291230	0,1386550	1
66	-0,5686030	0,2918960	0,0303719	1,6811e-06	0,1422000	0,0807810	1
67	-0,5772400	0,0823112	0,0181770	1,4881e-06	0,1240490	0,1319060	1
68	-0,5978160	0,1281650	0,0127130	1,3053e-06	0,1104730	0,2390380	1
69	-0,6124390	0,3236790	0,0242420	1,3434e-06	0,1218040	0,0963688	1
70	-0,6215840	0,0864958	0,0093761	1,1646e-06	0,0991922	0,1599240	1
71	-0,6340950	0,1632170	0,0093332	1,0982e-06	0,0942610	0,1456000	1

72	-0,6447910	0,4033160	0,0237093	1,1267e-06	0,1058790	0,0754297	1
73	-0,6543750	0,0929196	0,0094613	9,4306e-07	0,0779963	0,1986890	1
74	-0,6708600	0,1655600	0,0090760	7,9920e-07	0,0639560	0,2704020	1
75	-0,6712350	0,5499300	0,0149891	8,2890e-07	0,0825354	0,0033752	1
76	-0,6814960	0,1153160	0,0031651	7,1012e-07	0,0581922	0,2095430	1
77	-0,6964560	0,1680480	0,0052305	6,4711e-07	0,0502768	0,2268480	1
78	-0,6969280	0,3702420	0,0163494	6,6432e-07	0,0647860	0,0047570	1
79	-0,7160420	0,0804126	0,0014643	4,5438e-07	0,0332969	0,4835380	1
80	-0,7205090	0,1261930	0,0019800	4,4848e-07	0,0358425	0,0937751	1
81	-0,7207290	0,2396200	0,0040124	4,4519e-07	0,0407804	0,0032912	1
82	-0,7490520	0,1531010	0,0024874	3,6163e-07	0,0137135	0,5408190	1
83	-0,7501580	0,2203950	0,0039555	3,6305e-07	0,0196535	0,0195224	1
84	-0,7666540	0,1685700	0,0021750	3,2554e-07	0,0059820	0,3454260	1
85	-0,7685660	0,0680059	0,0174375	3,2785e-07	0,0248155	0,0439620	1
86	-0,7719480	0,1213700	0,0069512	3,2475e-07	0,0200944	0,0707057	1
87	-0,7725890	0,1524710	0,0095919	3,2195e-07	0,0203568	0,0130437	1
88	-0,7907280	0,2009450	0,0101355	3,0875e-07	0,0070792	0,3335890	1
89	-0,7972550	0,1166930	0,0043712	2,8287e-07	0,0013099	0,1860420	1
90	-0,8018380	0,2580790	0,0052730	2,8026e-07	0,0002549	0,0976129	1
91	-0,8033910	0,3606590	0,0080952	2,8222e-07	0,0033291	0,0294932	1
92	-0,8098100	0,2713090	0,0249949	2,8845e-07	0,0136513	0,1152270	1
93	-0,8119920	0,0479428	0,0067155	2,6791e-07	0,0021418	0,0984524	1
94	-0,8136160	0,0957086	0,0053826	2,6130e-07	0,0037516	0,0536489	1
95	-0,8193900	0,3966420	0,0219746	2,7335e-07	0,0177922	0,0973812	1
96	-0,8232850	0,0376727	0,0052879	2,1815e-07	0,0076119	0,2532910	1
97	-0,8269450	0,0738905	0,0026387	1,9441e-07	0,0118397	0,1847740	1
98	-0,8380600	0,2545550	0,0084986	1,8220e-07	0,0164402	0,3305400	1
99	-0,8387550	0,2672980	0,0069463	1,8421e-07	0,0143712	0,0242423	1
100	-0,8403540	0,2343940	0,0113322	1,8296e-07	0,0076896	0,0577440	1
101	-0,8463780	0,1388650	0,0055665	1,5450e-07	0,0133434	0,3065100	1
102	-0,8612780	0,2061730	0,0056080	1,1279e-07	0,0216726	0,7013820	1
103	-0,8616670	0,1739940	0,0059362	1,1357e-07	0,0146946	0,0318584	1
104	-0,8618430	0,2276240	0,0098200	1,1304e-07	0,0140929	0,0128701	1
105	-0,8623050	0,5438560	0,0121946	1,1489e-07	0,0076859	0,0175321	1
106	-0,8677670	0,3082170	0,0046380	1,0289e-07	0,0132764	0,3191630	1
107	-0,8685120	0,1028000	0,0155129	1,0034e-07	0,0068864	0,0709830	1
108	-0,8693320	0,0768697	0,0056852	9,3543e-08	0,0074275	0,1048700	1
109	-0,8745730	0,1440750	0,0055136	7,4798e-08	0,0070208	0,4562750	1
110	-0,8746600	0,1048050	0,0050342	7,5176e-08	0,0026696	0,0119753	1
111	-0,8780030	0,1234730	0,0073312	5,5084e-08	0,0048032	0,4695750	1
112	-0,8780400	0,1454930	0,0128784	5,5529e-08	0,0017091	0,0064185	1
113	-0,8787590	0,0843458	0,0076932	4,9127e-08	0,0008743	0,1763260	1
114	-0,8791710	0,0553271	0,0057602	4,4357e-08	0,0003352	0,1367940	1
115	-0,8794170	0,0611843	0,0074589	4,1767e-08	0,0006666	0,0848504	1
116	-0,8800460	0,0673670	0,0050830	3,5598e-08	0,0007466	0,2181160	1
117	-0,8802880	0,0726744	0,0037253	3,3339e-08	0,0010885	0,0961831	1
118	-0,8803540	0,1125510	0,0049443	3,2836e-08	0,0014565	0,0271467	1

119	-0,8814990	0,0821096	0,0021990	1,8843e-08	0,0002639	0,5857040	1
120	-0,8815200	0,0919244	0,0035475	1,8700e-08	0,0005629	0,0156575	1
121	-0,8821440	0,1084520	0,0026889	1,2813e-08	0,0001249	0,4677980	1
122	-0,8821480	0,1207460	0,0033004	1,2825e-08	0,0005709	0,0036992	1
123	-0,8827690	0,0927509	0,0018358	5,9166e-09	0,0001841	0,7606410	1
124	-0,8827820	0,1064360	0,0027119	5,8346e-09	0,0002256	0,0269477	1
125	-0,8827820	0,0592624	0,0014805	5,8285e-09	8,1818e-05	0,000304	1
126	-0,8830140	0,0985900	0,0011222	3,5932e-09	4,1281e-06	0,564280	1
127	-0,8830170	0,1027170	0,0015358	3,5769e-09	0,00019402	0,009264	1
128	-0,8831460	0,0709190	0,0008514	1,9094e-09	4,2371e-06	0,628472	1
129	-0,8831470	0,0671693	0,0007515	1,9013e-09	8,2583e-05	0,007445	1
130	-0,8831470	0,0621288	0,0003987	1,9018e-09	7,7644e-05	6,3383e-05	1
131	-0,8831470	0,0680072	0,0002678	1,8928e-09	8,4283e-05	0,007000	1
132	-0,8831470	0,0718694	0,0002345	1,8931e-09	9,4118e-05	7,6300e-05	1
133	-0,8831490	0,0692808	0,0001255	1,8753e-09	8,9102e-05	0,729662	1
134	-0,8832840	0,0013304	0,0001002	5,9302e-10	2,0252e-06	0,2997430	1
135	-0,8832850	0,0009897	5,9481e-05	4,237e-10	1,3961e-06	0,3512570	1

Tabela A.3: Exemplo 2. Redução da função objetivo, histórico das condições de complementariedade e critérios de parada obtidos pelo FDIPA-GSDP(3).

### A.3 Tabela de Iterações - Treliça de 1580 Barras

Assim como na seção anterior, percebe-se, a partir da Tabela A.4, que o número de iterações para encontrar a solução ótima também foi bem menor que o número de variáveis de projeto ( $n$ ) que, neste exemplo, é igual a 1581.

$iter$	$f$	$\ d_0\ $	$\ \nabla L\ $	$\ G_{\gamma_0}\ $	$\ tr(\Lambda_0 A)\ $	$t$	$itlin$
0	-0,0204494						
1	-0,0204578	7,1597e-05	7,1597e-05	1,1658e-10	0,0011438	0,1176490	7
2	-0,0204711	0,0502357	0,0498468	2,5644e-09	0,0122224	0,0035347	10
3	-0,0204761	0,0415399	0,0298718	5,7965e-09	0,0129034	0,0032714	17
4	-0,0212684	0,0559070	0,0282022	6,3682e-07	0,0085012	0,4862940	1
5	-0,0214447	0,1078600	0,0302952	1,1653e-06	0,0192043	0,0658181	1
6	-0,0216027	0,1179130	0,0241941	1,2915e-06	0,0185226	0,0495151	1
7	-0,0217243	0,0644498	0,0175686	1,2664e-06	0,0196529	0,0672070	1
8	-0,0221916	0,0723569	0,0203839	1,1018e-06	0,0244788	0,2307220	1
9	-0,0223011	0,0630857	0,0204293	1,3592e-06	0,0303355	0,0530222	1
10	-0,0227039	0,0908089	0,0252042	1,4439e-06	0,0347982	0,1290360	1
11	-0,0229147	0,0820133	0,0247367	1,6864e-06	0,0409273	0,0634093	1
12	-0,0234153	0,0999870	0,0258702	1,6752e-06	0,0440013	0,1164060	1
13	-0,0236001	0,1312460	0,0267339	1,7014e-06	0,0455273	0,0314529	1
14	-0,0241074	0,1031390	0,0241636	1,5613e-06	0,0440937	0,1009020	1
15	-0,0246244	0,1125490	0,0249446	1,5491e-06	0,0463474	0,0885936	1

16	-0,0252828	0,1292820	0,0320553	1,4639e-06	0,0454101	0,0869265	1
17	-0,0260613	0,1664980	0,0337027	1,4819e-06	0,0472131	0,0744596	1
18	-0,0270006	0,1826740	0,0296859	1,4801e-06	0,0483470	0,0799936	1
19	-0,0280954	0,1851900	0,0315660	1,4742e-06	0,0496201	0,0907369	1
20	-0,0295067	0,2214090	0,0286811	1,4889e-06	0,0509192	0,0956537	1
21	-0,0306390	0,2130510	0,0413997	1,4004e-06	0,0489571	0,0741656	1
22	-0,0331995	0,2860480	0,0377453	1,4952e-06	0,0525520	0,1217350	1
23	-0,0358687	0,3015210	0,0439737	1,4843e-06	0,0531602	0,1148120	1
24	-0,0369630	0,2695460	0,0490639	1,4372e-06	0,0535868	0,0499375	1
25	-0,0409386	0,3882890	0,0521139	1,5597e-06	0,0568540	0,1242700	1
26	-0,0434581	0,3335740	0,0583440	1,4449e-06	0,0551347	0,0842524	1
27	-0,0475433	0,3546440	0,0596179	1,4808e-06	0,0580622	0,1251540	1
28	-0,0501322	0,4305090	0,0741979	1,5078e-06	0,0612932	0,0608821	1
29	-0,0542736	0,3858040	0,0697436	1,3964e-06	0,0574267	0,1040760	1
30	-0,0586614	0,4016730	0,0692445	1,4254e-06	0,0611812	0,1034140	1
31	-0,0627935	0,4541160	0,0843083	1,3897e-06	0,0612423	0,0801892	1
32	-0,0677539	0,4400010	0,0748716	1,3199e-06	0,0591396	0,0964438	1
33	-0,0720631	0,4553680	0,0746439	1,3329e-06	0,0620613	0,0794920	1
34	-0,0780962	0,4918750	0,0991532	1,2522e-06	0,0589755	0,0985855	1
35	-0,0831498	0,5852690	0,0857429	1,2333e-06	0,0593871	0,0710107	1
36	-0,0908532	0,5369750	0,1020950	1,2102e-06	0,0591052	0,1144770	1
37	-0,0954281	0,5658850	0,0932545	1,2072e-06	0,0617947	0,0588161	1
38	-0,1029770	0,4248190	0,1150660	1,1379e-06	0,0596536	0,1086940	1
39	-0,1096270	0,6662860	0,1041560	1,1737e-06	0,0629585	0,0648984	1
40	-0,1176300	0,6802340	0,1181420	1,1567e-06	0,0614645	0,0817822	1
41	-0,1264110	0,6322560	0,1049070	1,1571e-06	0,0632749	0,0909858	1
42	-0,1325610	0,4791070	0,1218990	1,1488e-06	0,0664962	0,0633596	1
43	-0,1450760	0,5587600	0,1106730	1,1201e-06	0,0637336	0,1115020	1
44	-0,1511830	0,6834250	0,1212980	1,1331e-06	0,0678863	0,0484824	1
45	-0,1633720	0,8123260	0,1151820	1,1343e-06	0,0653186	0,0942546	1
46	-0,1679310	0,4826870	0,1131820	1,1422e-06	0,0699512	0,0439080	1
47	-0,1852060	0,4847100	0,1072230	1,0499e-06	0,0645112	0,1590690	1
48	-0,1924300	0,7321400	0,1289540	1,0284e-06	0,0686113	0,0520324	1
49	-0,2029530	0,7849080	0,1071520	9,4003e-07	0,0578693	0,0881752	1
50	-0,2061200	0,3896420	0,1046160	1,0256e-06	0,0719851	0,0352305	1
51	-0,2196530	0,5048040	0,0928581	9,9455e-07	0,0720365	0,1174170	1
52	-0,2419380	0,4357710	0,1055240	8,2078e-07	0,0597757	0,2124940	1
53	-0,2482030	0,8612880	0,1394170	8,5029e-07	0,0717247	0,0400316	1
54	-0,2611270	0,8460830	0,1119970	7,1790e-07	0,0466493	0,1067160	1
55	-0,2669070	0,4815520	0,1117160	8,1068e-07	0,0673169	0,0573498	1
56	-0,2780150	0,6618320	0,0844489	8,3836e-07	0,0760975	0,0808860	1
57	-0,2918880	0,2958580	0,0713684	7,1142e-07	0,0631726	0,1567490	1
58	-0,3064850	0,4443570	0,0795994	6,5027e-07	0,0631101	0,1372880	1
59	-0,3187780	0,7645400	0,0843674	5,9252e-07	0,0544604	0,0987450	1
60	-0,3330130	0,4138620	0,0648717	4,6750e-07	0,0353370	0,2006980	1
61	-0,3459520	0,3619050	0,0951707	4,4929e-07	0,0520181	0,1788070	1
62	-0,3524430	1,0184800	0,0866263	4,8405e-07	0,0666914	0,0456401	1

63	-0,3635570	0,2461990	0,0473057	3,6722e-07	0,0406093	0,2171070	1
64	-0,3701700	0,2655250	0,0515152	3,4810e-07	0,0482070	0,1211910	1
65	-0,3746690	0,3688100	0,0681397	3,6175e-07	0,0619061	0,0547038	1
66	-0,3812990	0,4438100	0,0660563	3,5551e-07	0,0649436	0,0649122	1
67	-0,3921310	0,2861100	0,0639000	3,1070e-07	0,0512107	0,1439530	1
68	-0,3995380	0,5695050	0,0778560	3,1148e-07	0,0592998	0,0671349	1
69	-0,4038800	0,3838190	0,0576525	2,7805e-07	0,0389553	0,0640424	1
70	-0,4189090	0,4933760	0,0680553	2,5305e-07	0,0379134	0,1902050	1
71	-0,4265840	0,7566000	0,0906495	2,6230e-07	0,0508825	0,0680717	1
72	-0,4333260	0,2442550	0,0465137	2,2342e-07	0,0306710	0,1314740	1
73	-0,4495110	0,2283250	0,0472972	1,7037e-07	0,0239256	0,3223120	1
74	-0,4589530	0,2715630	0,0641888	1,6599e-07	0,0341354	0,1372430	1
75	-0,4708240	0,3572380	0,0599402	1,4510e-07	0,0180691	0,1801760	1
76	-0,4723600	0,3558990	0,0428384	1,3954e-07	0,0047234	0,0316334	1
77	-0,4850970	0,4310390	0,0328448	1,1491e-07	0,0059525	0,2646130	1
78	-0,4889330	0,6353830	0,0348284	1,1759e-07	0,0047941	0,0609476	1
79	-0,4921540	0,1867090	0,0484230	1,1926e-07	0,0224750	0,0586903	1
80	-0,4948430	0,0925895	0,0243098	1,0694e-07	0,0101945	0,0948041	1
81	-0,5030470	0,1499450	0,0265058	9,4038e-08	0,0103405	0,1998620	1
82	-0,5049170	0,2529880	0,0583106	9,5489e-08	0,0207071	0,0303845	1
83	-0,5057070	0,1399250	0,0287689	9,0302e-08	0,0049594	0,0263217	1
84	-0,5075000	0,3007610	0,0279387	8,8963e-08	0,0060971	0,0404246	1
85	-0,5174250	0,3440420	0,0298426	8,2152e-08	0,0044990	0,1849700	1
86	-0,5185500	0,1515470	0,0458905	8,2681e-08	0,0106867	0,0239041	1
87	-0,5205100	0,1176320	0,0262638	7,6858e-08	0,0009013	0,0677764	1
88	-0,5300570	0,3019520	0,0251175	6,7853e-08	0,0084713	0,2187540	1
89	-0,5317050	0,2171700	0,0348769	6,7875e-08	0,0046096	0,0436603	1
90	-0,5391490	0,3486880	0,0565852	6,5115e-08	0,0033441	0,1413510	1
91	-0,5421100	0,3072590	0,0308566	6,0949e-08	0,0166792	0,0821179	1
92	-0,5457660	0,2661960	0,0250687	5,6358e-08	0,0214254	0,1213400	1
93	-0,5465640	0,4938250	0,0593679	5,8065e-08	0,0082840	0,0160809	1
94	-0,5501730	0,2867690	0,0359657	5,2690e-08	0,0210187	0,1249610	1
95	-0,5532840	0,5289920	0,0284018	5,0802e-08	0,0243432	0,0803665	1
96	-0,5602760	0,6649840	0,0345533	4,7925e-08	0,0283917	0,1624770	1
97	-0,5661440	0,9501950	0,0356172	4,7878e-08	0,0262314	0,1050300	1
98	-0,5706590	0,5223260	0,0354701	4,6635e-08	0,0238534	0,1137250	1
99	-0,5815620	0,3268240	0,0309761	4,1867e-08	0,0219044	0,2868260	1
100	-0,5822980	0,0772442	0,0197662	4,0628e-08	0,0242071	0,0428484	1
101	-0,5875820	0,1978730	0,0364265	3,8901e-08	0,0139127	0,1597820	1
102	-0,5877330	0,0534339	0,0145048	3,7301e-08	0,0283940	0,0158701	1
103	-0,5890050	0,1318070	0,0198368	3,5988e-08	0,0245396	0,0732364	1
104	-0,5908890	0,4331230	0,0539746	3,7095e-08	0,0084357	0,0470811	1
105	-0,5910520	0,1152210	0,0220138	3,4021e-08	0,0381922	0,0166231	1
106	-0,5911660	0,1150520	0,0134074	3,3863e-08	0,0388966	0,0113559	1
107	-0,5930710	0,4948620	0,0217555	3,3458e-08	0,0324039	0,0709784	1
108	-0,5951160	0,4835680	0,0370466	3,3534e-08	0,0218171	0,0693258	1
109	-0,5956560	0,2450110	0,0222079	3,1824e-08	0,0314704	0,0362975	1

110	-0,6006050	0,4210080	0,0218182	3,0648e-08	0,0205120	0,1765150	1
111	-0,6011830	0,0654238	0,0204367	2,9467e-08	0,0220644	0,0547294	1
112	-0,6068830	0,2505010	0,0268497	2,6975e-08	0,0175320	0,2398060	1
113	-0,6069710	0,1456350	0,0172891	2,6351e-08	0,0278595	0,0081929	1
114	-0,6093080	0,2218030	0,0129521	2,3983e-08	0,0279671	0,1596900	1
115	-0,6144940	0,3350470	0,0515062	2,3972e-08	0,0115691	0,1949660	1
116	-0,6145540	0,0740837	0,0148539	2,2627e-08	0,0338163	0,0113338	1
117	-0,6157360	0,1557480	0,0101751	2,0539e-08	0,0352670	0,1337560	1
118	-0,6164600	0,2629500	0,0252990	2,0213e-08	0,0303150	0,0543200	1
119	-0,6180760	0,5284190	0,0167440	2,0223e-08	0,0260548	0,0693259	1
120	-0,6196860	0,3932250	0,0451153	1,9991e-08	0,0200309	0,0830915	1
121	-0,6209780	0,2552330	0,0122783	1,8284e-08	0,0311499	0,1123720	1
122	-0,6239720	0,4608650	0,0113483	1,7469e-08	0,0322461	0,1646550	1
123	-0,6291150	1,2624500	0,0333195	2,0005e-08	0,0194144	0,1154110	1
124	-0,6295050	0,0985236	0,0087045	1,8125e-08	0,0356624	0,0732324	1
125	-0,6317370	0,1925380	0,0089203	1,4617e-08	0,0389707	0,2775390	1
126	-0,6349330	0,5996550	0,0326646	1,5008e-08	0,0289204	0,1541600	1
127	-0,6382480	0,3170500	0,0134565	1,3857e-08	0,0277117	0,2428020	1
128	-0,6383800	0,1266280	0,0247257	1,4152e-08	0,0141248	0,0129043	1
129	-0,6384120	0,0473983	0,0133502	1,3760e-08	0,0237017	0,0082434	1
130	-0,6386850	0,1362800	0,0113977	1,3532e-08	0,0231073	0,0342521	1
131	-0,6425720	0,3705580	0,0292719	1,3839e-08	0,0155134	0,2080780	1
132	-0,6428870	0,0549798	0,0123508	1,3000e-08	0,0185588	0,0728410	1
133	-0,6451230	0,1507510	0,0158492	1,1320e-08	0,0173219	0,2582100	1
134	-0,6455090	0,2322110	0,0154028	1,1374e-08	0,0126269	0,0354421	1
135	-0,6455630	0,1425420	0,0287290	1,1332e-08	0,0119845	0,0074624	1
136	-0,6463500	0,2621490	0,0169568	1,1145e-08	0,0122467	0,0722835	1
137	-0,6473340	0,2174310	0,0165093	1,0583e-08	0,0137996	0,1159690	1
138	-0,6488500	0,2206030	0,0233312	9,8348e-09	0,0134125	0,1801660	1
139	-0,6493470	0,3012470	0,0156289	9,7282e-09	0,0135377	0,0509127	1
140	-0,6499290	0,3612650	0,0127270	9,7429e-09	0,0115136	0,0532837	1
141	-0,6500750	0,2112200	0,0193855	9,6288e-09	0,0116374	0,0209362	1
142	-0,6518360	0,3169910	0,0160719	9,1265e-09	0,0121045	0,1905470	1
143	-0,6520020	0,4111300	0,0182499	9,1681e-09	0,0100576	0,0159160	1
144	-0,6527790	0,5397280	0,0218250	9,2716e-09	0,0078082	0,0623856	1
145	-0,6529320	0,3962070	0,0117387	9,1169e-09	0,0120243	0,0171991	1
146	-0,6529630	0,5418670	0,0100287	9,1554e-09	0,0108643	0,0027204	1
147	-0,6550900	0,5243060	0,0086393	8,9289e-09	0,0127692	0,1972710	1
148	-0,6555960	0,6690460	0,0098946	9,0235e-09	0,0103848	0,0405712	1
149	-0,6557650	0,5591360	0,0120130	9,0111e-09	0,0104725	0,0161591	1
150	-0,6576490	0,7183660	0,0178384	9,1442e-09	0,0098165	0,1475070	1
151	-0,6580650	0,6739170	0,0168653	9,1753e-09	0,0087162	0,0348782	1
152	-0,6583790	0,8394410	0,0158067	9,3143e-09	0,0056281	0,0214014	1
153	-0,6587870	0,6361170	0,0085240	9,2324e-09	0,0077892	0,0362596	1
154	-0,6589610	0,7515140	0,0080035	9,2717e-09	0,0071141	0,0135008	1
155	-0,6601500	0,7624700	0,0084223	9,4212e-09	0,0044269	0,0891331	1
156	-0,6605590	0,5734320	0,0091356	9,3694e-09	0,0048595	0,0376231	1

157	-0,6608520	0,6057400	0,0121333	9,4825e-09	0,0009131	0,0238566	1
158	-0,6611870	0,4370540	0,0131010	9,3353e-09	0,0032840	0,0362348	1
159	-0,6617090	0,4852760	0,0136861	9,2402e-09	0,0034184	0,0511445	1
160	-0,6619110	0,6610160	0,0183085	9,3555e-09	0,0010600	0,0154643	1
161	-0,6627850	0,3746930	0,0109739	8,7828e-09	0,0066852	0,1152230	1
162	-0,6631950	0,3773800	0,0091074	8,5441e-09	0,0093091	0,0561718	1
163	-0,6633550	0,7186260	0,0112008	8,7519e-09	0,0017202	0,0127073	1
164	-0,6645350	0,4755610	0,0078176	8,4208e-09	0,0040504	0,1357090	1
165	-0,6646940	0,4518040	0,0081594	8,4021e-09	0,0033183	0,0190921	1
166	-0,6653760	0,4660750	0,0103288	8,3485e-09	0,0000063	0,0757019	1
167	-0,6667210	0,3428550	0,0108867	7,7824e-09	0,0009045	0,1870510	1
168	-0,6675400	0,2758200	0,0158237	7,3386e-09	0,0006673	0,1365080	1
169	-0,6680410	0,3114860	0,0155845	7,1585e-09	0,0004617	0,0778169	1
170	-0,6687200	0,2443900	0,0125489	6,6456e-09	0,0020489	0,1380540	1
171	-0,6691400	0,3180240	0,0102630	6,4858e-09	0,0007164	0,0751239	1
172	-0,6694200	0,4024550	0,0098538	6,4647e-09	0,0019490	0,0443807	1
173	-0,6700120	0,2261490	0,0082336	5,9170e-09	0,0010769	0,1469520	1
174	-0,6704400	0,2333010	0,0062724	5,6056e-09	0,0006224	0,1074990	1
175	-0,6707170	0,2175040	0,0094655	5,4296e-09	0,0010224	0,0721916	1
176	-0,6714780	0,1832330	0,0105586	4,7729e-09	0,0007081	0,2237040	1
177	-0,6717310	0,1610830	0,0103763	4,5158e-09	0,0001656	0,0916472	1
178	-0,6723580	0,1937980	0,0099009	3,9669e-09	0,0001699	0,2227340	1
179	-0,6723790	0,1216630	0,0105119	3,9383e-09	0,0004651	0,0107544	1
180	-0,6724490	0,1205250	0,0088942	3,8233e-09	0,0003758	0,0395276	1
181	-0,6725630	0,2045000	0,0094818	3,7003e-09	0,0001288	0,0543264	1
182	-0,6729580	0,2824140	0,0082452	3,3579e-09	0,0000022	0,1695390	1
183	-0,6731400	0,0771605	0,0097536	3,0243e-09	0,0002725	0,1385970	1
184	-0,6731410	0,0984494	0,0078399	3,0211e-09	0,0002083	0,0007964	1
185	-0,6731590	0,2264450	0,0078042	3,0089e-09	0,0002441	0,0108934	1
186	-0,6734230	0,3299990	0,0071640	2,8024e-09	0,0005195	0,1322940	1
187	-0,6734390	0,1676740	0,0041876	2,7769e-09	0,0004770	0,0122170	1
188	-0,6734930	0,0835370	0,0042877	2,6740e-09	0,0004312	0,0501891	1
189	-0,6735030	0,0608800	0,0039722	2,6537e-09	0,0003222	0,0094263	1
190	-0,6736170	0,0766793	0,0046325	2,4433e-09	0,0003411	0,1097500	1
191	-0,6736180	0,1105030	0,0053145	2,4437e-09	0,0004471	0,0004711	1
192	-0,6736730	0,1450730	0,0056408	2,3639e-09	0,0004411	0,0486683	1
193	-0,6736740	0,1456160	0,0052297	2,3623e-09	0,0004228	0,0008714	1
194	-0,6737760	0,1432080	0,0047700	2,2015e-09	0,0003628	0,0981287	1
195	-0,6737850	0,1379710	0,0052169	2,1882e-09	0,0004082	0,0090040	1
196	-0,6737880	0,1579280	0,0059022	2,1839e-09	0,0004385	0,0032887	1
197	-0,6738130	0,1960390	0,0058189	2,1512e-09	0,0004137	0,0230922	1
198	-0,6738460	0,1911880	0,0048728	2,1043e-09	0,0003326	0,0319956	1
199	-0,6738470	0,2337520	0,0043126	2,1024e-09	0,0004800	0,0015734	1
200	-0,6738920	0,2781060	0,0039321	2,0530e-09	0,0006108	0,0379585	1
201	-0,6739580	0,2656620	0,0034835	1,9758e-09	0,0005724	0,0589624	1
202	-0,6739610	0,2685520	0,0033444	1,9709e-09	0,0006107	0,0029504	1
203	-0,6739740	0,2644180	0,0031365	1,9558e-09	0,0006259	0,0117887	1

204	-0,6739860	0,2655900	0,0026207	1,9419e-09	0,0006436	0,0115753	1
205	-0,6739910	0,2608390	0,0025585	1,9374e-09	0,0006544	0,0041150	1
206	-0,6739940	0,2597040	0,0026085	1,9341e-09	0,0006452	0,0029560	1
207	-0,6739970	0,2606590	0,0023865	1,9297e-09	0,0006341	0,0034804	1
208	-0,6740060	0,2657560	0,0022880	1,9198e-09	0,0006314	0,0079928	1
209	-0,6740160	0,2729550	0,0023386	1,9083e-09	0,0006344	0,0094112	1
210	-0,6740220	0,2791360	0,0024307	1,9013e-09	0,0006347	0,0056946	1
211	-0,6740250	0,2859570	0,0025156	1,8984e-09	0,0006302	0,0023593	1
212	-0,6740280	0,2938770	0,0024854	1,8953e-09	0,0006212	0,0028919	1
213	-0,6740340	0,2899600	0,0021122	1,8898e-09	0,0005899	0,0051212	1
214	-0,6740420	0,2893900	0,0016738	1,8808e-09	0,0005843	0,0082436	1
215	-0,6740500	0,2773210	0,0013712	1,8729e-09	0,0005774	0,0072755	1
216	-0,6740550	0,2668710	0,0012229	1,8683e-09	0,0005623	0,0041434	1
217	-0,6740580	0,2563990	0,0012102	1,8650e-09	0,0005433	0,0029595	1
218	-0,6740620	0,2506770	0,0013268	1,8602e-09	0,0005287	0,0040697	1
219	-0,6740680	0,2464870	0,0014988	1,8531e-09	0,0005156	0,0059498	1
220	-0,6740750	0,2461240	0,0016847	1,8445e-09	0,0005059	0,0072617	1
221	-0,6740810	0,2499730	0,0018726	1,8381e-09	0,0005019	0,0053637	1
222	-0,6740840	0,2580610	0,0020166	1,8336e-09	0,0005048	0,0038689	1
223	-0,6740880	0,2633730	0,0020134	1,8292e-09	0,0005077	0,0036781	1
224	-0,6740930	0,2680620	0,0019245	1,8235e-09	0,0005147	0,0049535	1
225	-0,6741000	0,2704900	0,0018460	1,8163e-09	0,0005220	0,0062141	1
226	-0,6741060	0,2702130	0,0017050	1,8089e-09	0,0005264	0,0062465	1
227	-0,6741110	0,2662430	0,0014336	1,8033e-09	0,0005280	0,0047626	1
228	-0,6741150	0,2599180	0,0011265	1,7982e-09	0,0005256	0,0043649	1
229	-0,6741200	0,2543060	0,0008449	1,7929e-09	0,0005202	0,0046082	1
230	-0,6741250	0,2502190	0,0006427	1,7860e-09	0,0005132	0,0059992	1
231	-0,6741320	0,2467700	0,0005936	1,7788e-09	0,0005053	0,0063794	1
232	-0,6741370	0,2441180	0,0006348	1,7725e-09	0,0004981	0,0055586	1
233	-0,6741410	0,2427740	0,0007088	1,7674e-09	0,0004923	0,0044668	1
234	-0,6741460	0,2435510	0,0008780	1,7621e-09	0,0004885	0,0046404	1
235	-0,6741510	0,2462630	0,0011517	1,7560e-09	0,0004873	0,0052213	1
236	-0,6741570	0,2506230	0,0013433	1,7486e-09	0,0004884	0,0065485	1
237	-0,6741630	0,2561380	0,0013138	1,7419e-09	0,0004928	0,0062333	1
238	-0,6741680	0,2573760	0,0012024	1,7361e-09	0,0004980	0,0055754	1
239	-0,6741730	0,2546750	0,0011825	1,7311e-09	0,0005022	0,0048056	1
240	-0,6741780	0,2553230	0,0010853	1,7255e-09	0,0005071	0,0052028	1
241	-0,6741830	0,2584760	0,0009378	1,7191e-09	0,0005126	0,0057611	1
242	-0,6741890	0,2610210	0,0008290	1,7119e-09	0,0005171	0,0064276	1
243	-0,6741940	0,2608740	0,0006680	1,7052e-09	0,0005204	0,0058943	1
244	-0,6742000	0,2594740	0,0004977	1,6990e-09	0,0005218	0,0055841	1
245	-0,6742040	0,2574280	0,0004212	1,6932e-09	0,0005214	0,0052625	1
246	-0,6742100	0,2556230	0,0004113	1,6870e-09	0,0005196	0,0056931	1
247	-0,6742150	0,2543770	0,0004229	1,6805e-09	0,0005169	0,0059941	1
248	-0,6742210	0,2538460	0,0004611	1,6735e-09	0,0005132	0,0064005	1
249	-0,6742260	0,2537210	0,0005011	1,6671e-09	0,0005085	0,0059366	1
250	-0,6742320	0,2534900	0,0004893	1,6606e-09	0,0005032	0,0059583	1

251	-0,6742370	0,2529050	0,0004468	1,6545e-09	0,0004985	0,0057601	1
252	-0,6742420	0,2523120	0,0004283	1,6479e-09	0,0004945	0,0062530	1
253	-0,6742480	0,2516400	0,0004429	1,6414e-09	0,0004912	0,0062420	1
254	-0,6742530	0,2511960	0,0004716	1,6350e-09	0,0004883	0,0061200	1
255	-0,6742580	0,2510330	0,0005029	1,6291e-09	0,0004858	0,0057116	1
256	-0,6742630	0,2510290	0,0005235	1,6229e-09	0,0004833	0,0060351	1
257	-0,6742690	0,2507900	0,0005228	1,6165e-09	0,0004809	0,0061176	1
258	-0,6742740	0,2504910	0,0005101	1,6097e-09	0,0004783	0,0064307	1
259	-0,6742790	0,2501850	0,0004948	1,6033e-09	0,0004760	0,0061648	1
260	-0,6742850	0,2500020	0,0004624	1,5968e-09	0,0004737	0,0062752	1
261	-0,6742900	0,2495750	0,0003927	1,5905e-09	0,0004719	0,0060201	1
262	-0,6742950	0,2488810	0,0003323	1,5841e-09	0,0004703	0,0061935	1
263	-0,6743000	0,2480730	0,0003211	1,5777e-09	0,0004687	0,0061768	1
264	-0,6743060	0,2474930	0,0003302	1,5708e-09	0,0004669	0,0066176	1
265	-0,6743110	0,2474030	0,0003382	1,5643e-09	0,0004656	0,0062873	1
266	-0,6743160	0,2478740	0,0003501	1,5581e-09	0,0004648	0,0060294	1
267	-0,6743190	0,2480090	0,0003484	1,5544e-09	0,0004645	0,0036184	1
268	-0,6743220	0,2481490	0,0003103	1,5499e-09	0,0004628	0,0043817	1
269	-0,6743280	0,2478360	0,0002509	1,5429e-09	0,0004597	0,0068151	1
270	-0,6743360	0,2467780	0,0002034	1,5325e-09	0,0004551	0,0102796	1
271	-0,6743430	0,2447270	0,0001666	1,5233e-09	0,0004515	0,0090272	1
272	-0,6743480	0,2428440	0,0001269	1,5175e-09	0,0004495	0,0057550	1
273	-0,6743500	0,1414200	9,6781e-05	1,5133e-09	0,0004481	0,0281888	1
274	-0,6743780	0,0087200	9,9593e-05	1,5562e-09	0,0004500	0,0241490	1

Tabela A.4: Exemplo 3. Redução da função objetivo, histórico das condições de complementariedade e critérios de parada obtidos pelo FDIPA-GSDP(3).