RODOLFO GOTARDI BEGIATO

Métodos híbridos e livres de derivadas para resolução de sistemas não lineares

Campinas
2012

Universidade Estadual de Campinas
Instituto de Matemática, Estatística e Computação Científica

Rodolfo Gotardi Begiato

Métodos híbridos e livres de derivadas para resolução de sistemas não lineares

Orientadora: Profa. Dra. Márcia Aparecida Gomes Ruggiero (IMECC-UNICAMP)
Coorientadora: Profa. Dra. Sandra Augusta Santos (IMECC-UNICAMP)

Tese de doutorado apresentada ao Instituto de Matemática, Estatística e Computação Científica da Unicamp para obtenção do título de Doutor em Matemática Aplicada.

Este exemplar corresponde à redação final da tese defendida pelo aluno Rodolfo Gotardi Begiato e orientada pela Profa. Dra. Márcia Aparecida Gomes Ruggiero.

Profa. Dra. Márcia Aparecida Gomes Ruggiero

Profa. Dra. Sandra Augusta Santos

Campinas
2012

FICHA CATALOGRÁFICA ELABORADA POR
MARIA FABIANA BEZERRA MULLER - CRB8/6162
BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E
COMPUTAÇÃO CIENTÍFICA - UNICAMP

Begiato, Rodolfo Gotardi, 1980-
B394m Métodos híbridos e livres de derivadas para resolução de sistemas não lineares / Rodolfo Gotardi Begiato. – Campinas, SP : [s.n.], 2012.

Orientador: Márcia Aparecida Gomes Ruggiero.
Coorientador: Sandra Augusta Santos.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Sistemas não lineares. 2. Otimização sem derivadas. 3. Métodos iterativos (matemática). 4. Busca não-monótona. 5. Newton, método de. I. Ruggiero, Márcia Aparecida Gomes,1956-. II. Santos, Sandra Augusta,1964-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em inglês: Hybrid derivative-free methods for nonlinear systems
Palavras-chave em inglês:
Nonlinear systems
Derivative-free optimization
Iterative methods (Mathematics)
Nommonotone line search
Newton method
Área de concentração: Matemática Aplicada
Titulação: Doutor em Matemática Aplicada
Banca examinadora:
Márcia Aparecida Gomes Ruggiero [Orientador]
José Mario Martínez Pérez
Maria Aparecida Diniz Ehrhardt
Ademir Alves Ribeiro
Juliano de Bem Francisco
Data de defesa: 05-09-2012
Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 05 de setembro de 2012 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). MÁRCIA APARECIDA GOMES RUGGIERO

Prof(a). Dr(a). ADEMIR ALVES RIBEIRO

Prof(a). Dr(a). JULIANO DE BEM FRANCISCO

Prof(a). Dr(a). MARIA APARECIDA DINIZ EHRHARDT

Prof(a). Dr(a). JOSÉ MARIO MARTÍNEZ PÉREZ

Agradecimentos

Agradeço:

- Às Profas. Dras. Márcia Aparecida Gomes Ruggiero, Sandra Augusta Santos e Ana Luísa Custódio, pela orientação e dedicação.
- Aos Profs. Drs. Mario Martínez, Ademir Alves Ribeiro, Juliano de Bem Francisco e às Profas. Dras. Maria Aparecida Diniz Ehrhardt e Vera Lúcia da Rocha Lopes, pelos valiosos comentários a respeito deste trabalho.
- Aos funcionários do IMECC, em especial àqueles da secretaria de pós-graduação, por me ajudar com prazos e datas, impedindo minha exclusão do programa por motivos burocráticos.
- Ao grupo de otimização da Unicamp, pela aprendizagem proporcionada.
- À Capes e ao CNPQ, pelo auxílio financeiro.
- A Lia, Wado, Maurício, João Paulo, Maria Rosa e Mario pelo apoio e compreensão.
- A meus amigos, pelo companheirismo (tomado pelo enorme temor de que alguém seja esquecido, não os citarei nominalmente).
- A Paula e Marina, por tudo.

Resumo

O objetivo desta tese é tratar da resolução de sistemas não lineares de grande porte, em que as funções são continuamente diferenciáveis, por meio de uma abordagem híbrida que utiliza um método iterativo com duas fases. A primeira fase consiste de versões sem derivadas do método do ponto fixo empregando parâmetros espectrais para determinar o tamanho do passo da direção residual. A segunda fase é constituída pelo método de Newton inexato em uma abordagem matrix-free, em que é acoplado o método GMRES para resolver o sistema linear que determina a nova direção de busca. O método híbrido combina ordenadamente as duas fases de forma que a segunda é acionada somente em caso de falha na primeira e, em ambas, uma condição de decréscimo não-monótono deve ser verificada para aceitação de novos pontos.

Desenvolvemos ainda um segundo método, em que uma terceira fase de busca direta é acionada em situações em que o excesso de buscas lineares faz com que o tamanho de passo na direção do método de Newton inexato torne-se demasiadamente pequeno. São estabelecidos os resultados de convergência dos métodos propostos. O desempenho computacional é avaliado em uma série de testes numéricos com problemas tradicionalmente encontrados na literatura.

Tanto a análise teórica quanto a numérica evidenciam a viabilidade das abordagens apresentadas neste trabalho.

Palavras-chave: Sistemas não lineares; otimização sem derivadas; métodos espectrais; Newton inexato; busca direta; busca linear não monótona.

Abstract

This thesis handles large-scale nonlinear systems for which all the involved functions are continuously differentiable. They are solved by means of a hybrid approach based on an iterative method with two phases. The first phase is defined by derivative-free versions of a fixed-point method that employs spectral parameters to define the steplength along the residual direction. The second phase consists of a matrix-free inexact Newton method that employs the GMRES to solve the linear system that computes the search direction. The proposed hybrid method neatly combines the two phases in such a way that the second is called only in case the first one fails. To accept new points in both phases, a nonmonotone decrease condition upon a merit function has to be verified.

A second method is developed as well, with a third phase based on direct search, that should act whenever too many line searches have excessively decreased the steplenght along the inexact-Newton direction. Convergence results for the proposed methods are established. The computational performance is assessed in a set of numerical experiments with problems from the literature.

Both the theoretical and the experimental analysis corroborate the feasibility of the proposed strategies.

Keywords: Nonlinear systems; derivative free; spectral methods; inexact Newton; direct search; nonmonotone linesearch.

Lista de Notações

$\mathbb{N}$	conjunto dos números naturais
$\mathbb{Z}$	conjunto dos números inteiros
$\mathbb{Q}$	conjunto dos números racionais
$\mathbb{R}$	conjunto dos números reais
$\mathbb{R}_+$	conjunto dos números reais não negativos
$\mathbb{R}^n$	conjunto dos vetores colunas $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$
$A : m \times n$	matriz com m linhas e n colunas
$A^\top$	matriz transposta
$A(i, j)$	elemento da i-ésima linha e j-ésima coluna da matriz A
$A = [a_1, \ldots, a_n]$	matriz cujas colunas são, ordenadamente, $a_1, \ldots, a_n$
$\lvert . \rvert$	aplicação módulo
$\lVert . \rVert$	aplicação norma euclidiana
$F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^\top$	imagem do ponto x por uma função $F : \mathbb{R}^n \to \mathbb{R}^n$
$\nabla f(x)$	gradiente da função $f : \mathbb{R}^n \to \mathbb{R}$ no ponto x
$\nabla^2 f(x)$	hessiana da função $f : \mathbb{R}^n \to \mathbb{R}$ no ponto x
$J(x)$	matriz jacobiana da função $F : \mathbb{R}^n \to \mathbb{R}^n$ no ponto x
I	matriz identidade
e_i	i-ésimo vetor da base canônica de $\mathbb{R}^n$
e	vetor com todas as componentes iguais a 1
$\#(A)$	cardinalidade do conjunto A

Lista de Algoritmos

2.1 DFSANE 15

2.2 DFSANE - O1 18

2.3 DFSANE - O2 19

2.4 DFSANE com terceira direção 21

3.1 Processo de Arnoldi para ortonormalização 39

3.2 Método FDGMRES 40

3.3 FDGMRES(m) 42

3.4 HIB1 43

3.5 Busca linear não monótona 49

4.1 Busca direta direcional 77

4.2 HIB2 87

4.3 Busca multidirecional 89

Sumário

1 Introdução **1**

2 Métodos espectrais **9**

2.1 Introdução teórica 10

2.2 Método do resíduo espectral para sistemas não lineares 12

2.3 Modificações propostas 17

2.3.1 Estratégias de ordenação 17

2.3.2 Novas direções de busca 20

2.4 Problemas-teste 22

2.5 Ferramenta de análise de desempenho 23

2.6 Testes iniciais 24

3 Método híbrido para resolução de sistemas não lineares **35**

3.1 Método Newton-GMRES 35

3.1.1 GMRES com recomeços 41

3.2 Método proposto 42

3.2.1 Análise de convergência 45

3.3 Testes numéricos 54

4 Busca direta **73**

4.1 Métodos de Busca Direta 74

4.1.1 Convergência para problemas de minimização irrestrita 78

4.2 Novas bases positivas 83

4.3 Resolvendo sistemas não lineares de grande porte 86

4.3.1 Análise de convergência 90

4.4 Testes numéricos 94

5 Considerações finais e indicações para trabalhos futuros **103**

A Apêndice **107**

Capítulo

1 Introdução

Problemas originários das mais diversas áreas do conhecimento, como a própria matemática, a física, a química, a economia e as engenharias, necessitam da resolução de sistemas de equações não lineares com as seguintes características:

$$\begin{aligned} &F(x) = 0, \\ &F : \mathbb{R}^n \to \mathbb{R}^n. \end{aligned} \tag{1.1}$$

Embora, a rigor, a terminologia adequada seja *sistema de equações não lineares*, a fim de permitir uma melhor fluência no texto, adotaremos neste trabalho o termo simplificado *sistemas não lineares*, o que é uma prática comum na área.

Em geral, problemas de grande porte do tipo (1.1), onde a função F é continuamente diferenciável, são resolvidos através de métodos iterativos. *Métodos iterativos* são métodos que partem de um ponto inicial qualquer do domínio da função F e geram uma sequência $\{x_k\}$:

$$x_{k+1} = x_k + \lambda_k d_k, \tag{1.2}$$

onde o vetor d_k é chamado direção de busca e λ_k é um escalar no intervalo $(0, 1]$ e é chamado de tamanho do passo.

Entre os métodos clássicos para a resolução de sistemas não lineares encontra-se o método de Newton. Denotando a matriz Jacobiana de F no ponto x_k por $J(x_k)$, o método de Newton aplicado à resolução de (1.1) consiste em realizar iterações onde a direção d_k é obtida pela resolução do seguinte sistema linear:

$$J(x_k)d = -F(x_k). \tag{1.3}$$

A principal vantagem do método de Newton é sua boa taxa de convergência, que é quadrática, sob certas hipóteses. Resultados sobre a convergência local do método de Newton são facilmente encontrados na literatura, como, por exemplo, em [19] ou [40]. No entanto, a utilização prática do

método de Newton apresenta algumas dificuldades quando se trabalha com problemas de grande porte. O cálculo da matriz Jacobiana, bem como a posterior resolução do sistema linear, podem ser computacionalmente caros ou até impraticáveis. Uma alternativa bastante utilizada é o uso do método de Newton inexato proposto por Dembo, Eisenstat e Steihaug [18]. Ao invés de obter a direção de busca através da resolução do sistema linear (1.3), a proposta dos autores consiste em obter uma direção d_k que satisfaça à seguinte condição:

$$\|J(x_k)d_k + F(x_k)\| \leq \eta_k \|F(x_k)\|, \tag{1.4}$$

onde $\eta_k \in [0, 1)$ é chamado termo forçante.

Métodos baseados em projeções sobre subespaços de Krylov [34] têm sido utilizados para resolver aproximadamente o sistema linear (1.3) a fim de obter a direção d_k satisfazendo a condição (1.4). Um dos mais conhecidos métodos de projeção sobre subespaços de Krylov é o método GMRES (*Generalized Mininimal Residual*) [47]. A utilização do método GMRES pelo método de Newton inexato para resolução do sistema linear interno dá origem ao método conhecido como *Newton-GMRES*.

Em alguns casos [13, 29], a matriz Jacobiana da função F não pode ser fornecida, quer pela complexidade do cálculo, quer pela indisponibilidade das derivadas parciais. Esse cenário estimulou o desenvolvimento de algoritmos que não fazem o uso da matriz Jacobiana e, ao invés disso, ou utilizam uma estratégia que não faz uso de derivadas ou encontram aproximações para a matriz Jacobiana.

Os métodos de projeção sobre subespaços de Krylov têm a vantagem prática de que a matriz Jacobiana num ponto qualquer do espaço das variáveis somente é utilizada em produtos matriz-vetor e estes, por sua vez, podem ser aproximados através de séries de Taylor, utilizando somente o valor de F na vizinhança do ponto em questão. Esse procedimento é conhecido na literatura como *matrix-free* [47].

Também aproveitando as ideias do método de Newton, mas tentando evitar possíveis limitações práticas ocasionadas pelo cálculo da matriz Jacobiana e posterior resolução do sistema linear, temos os métodos quase-Newton. Os métodos quase-Newton têm como objetivo evitar a avaliação da matriz Jacobiana e simplificar a obtenção da direção de busca.

Neste sentido, esta direção é obtida através da resolução do sistema linear

$$B_k d + F(x_k) = 0, \tag{1.5}$$

no qual a matriz B_k é uma aproximação para a matriz Jacobiana de F no ponto x_k.

Em geral, pede-se que as matrizes da sequência $\{B_k\}$ satisfaçam a equação:

$$B_k(x_k - x_{k-1}) = F(x_k) - F(x_{k-1}), \tag{1.6}$$

chamada de *equação secante*.

Os métodos que satisfazem a equação (1.6) são denominados métodos secantes e têm taxa de convergência superlinear, sob determinadas hipóteses. Informações adicionais sobre os métodos quase-Newton podem ser encontradas em [9, 10, 40, 42].

Ainda se tratando de métodos derivados do método de Newton para resolução de sistemas não lineares, temos a classe dos métodos tensoriais. Neste caso, ao invés de procurar uma redução na carga computacional, sacrificando a taxa de convergência, é adotada uma filosofia contrária, e, em vez de utilizar uma aproximação linear para encontrar a direção de descida, a direção d_k é encontrada através de uma modelagem mais complexa, incluindo informação de segunda ordem. A nova direção deve ser solução do sistema não-linear:

$$F(x_k) + J(x_k)d + \frac{1}{2}d^\top(T_k d)d = 0, \tag{1.7}$$

onde T_k é um tensor com n^3 elementos $(T_k)_{ijl}$.

O principal objetivo deste tipo de modificação é fornecer métodos mais eficientes que o método de Newton para os sistemas não lineares em que a matriz Jacobiana é singular na solução, situação em que o método de Newton tem convergência lenta [48]. O uso de um método puro, em que as segundas derivadas são calculadas, é praticamente inviável pois requer excesso de cálculos computacionais (cálculos das derivadas parciais de segunda ordem e posterior resolução do sistema não-linear) e de requerimentos de mémoria (requer mais que $n^3/2$ posições de armazenamento).

Schnabel e Frank [48] propõem uma aproximação de posto baixo para os tensores que são representados por T_k. Essa aproximação torna os custos computacionais comparáveis aos do método de Newton. Conforme [48], o método é competitivo em problemas em que a Jacobiana pode ser avaliada e, além disso, mostra uma melhora de desempenho substancial nos testes envolvendo equações em que a matriz Jacobiana na solução tenha posto $n-1$ ou $n-2$.

Outra maneira usual de solucionar sistemas não lineares (1.1) de maneira satisfatória é através da utilização de métodos de ponto fixo, os quais são métodos iterativos que geram a sequência $\{x_k\}$ utilizando o vetor resíduo como direção de busca. Assim sendo, na k-ésima iteração trabalha-se com o vetor $d_k = \alpha_k F(x_k)$, com $\alpha_k \in \mathbb{R}$, evitando qualquer tipo de direção de busca que faça uso da matriz Jacobiana ou aproximações para esta.

Em geral, os métodos iterativos aqui apresentados têm somente garantida a sua convergência local. Isto é, os principais resultados de convergência somente são verificados sob a hipótese do algoritmo partir de um ponto suficientemente próximo da solução. Nem sempre é possível garantir a satisfação desta hipótese, principalmente quando não temos qualquer conhecimento sobre a solução do sistema.

Para garantir a convergência global, isto é, garantir que o algoritmo irá convergir a partir de qualquer ponto inicial, uma estratégia de globalização deve ser empregada. Para definir uma estratégia de globalização, usualmente se define uma função

$$f : \mathbb{R}^n \to \mathbb{R}, \tag{1.8}$$

chamada *função de mérito*, que indica quando determinado ponto é melhor ou pior, no sentido de estarmos nos aproximando de uma solução para o sistema não-linear em questão. Geralmente, se define como função de mérito $f(x) = \|F(x)\|$ ou $f(x) = \frac{1}{2}\|F(x)\|^2$, já que, em ambas, valores mais próximos de zero tendem a indicar maior proximidade com a solução do problema. E assim, determinar uma estratégia de globalização para a resolução do sistema não-linear equivale a determinar uma estratégia de globalização para o problema de minimizar irrestritamente a função de mérito definida.

Em problemas de minimização irrestrita, definida uma função $f : \mathbb{R}^n \to \mathbb{R}$, a qual chamamos *função objetivo*, devemos encontrar um vetor no espaço $\mathbb{R}^n$ que seja um ponto de mínimo da função objetivo. Um vetor $x \in \mathbb{R}^n$ pode ter a característica de ser um *ponto de mínimo local*, quando $f(x)$ é o menor valor encontrado para f em um conjunto aberto contendo x, ou ainda de ser um *ponto de mínimo global*, quando $f(x)$ é o menor valor encontrado para f considerando todos os elementos do espaço $\mathbb{R}^n$.

É importante ressaltar que, embora todo ponto $x \in \mathbb{R}^n$ que seja solução do sistema não linear (1.1) seja necessariamente um ponto de mínimo para a função de mérito, nem todo vetor que minimiza a função de mérito representa uma solução para o sistema não linear. Por exemplo, para o caso em que $f(x) = \frac{1}{2}\|F(x)\|^2$, temos $\nabla f(x) = J(x)^\top F(x)$ e, assim, só temos a garantia de que um ponto de mínimo de f é uma solução para o sistema não linear quando a matriz Jacobiana da função f é não-singular. Portanto, não devemos esperar que métodos globalizados para sistemas não lineares tenham a mesma eficácia que os mesmos métodos para problemas de minimização.

Problemas de minimização irrestrita também podem ser resolvidos através de métodos iterativos e têm sua convergência garantida para um ponto estacionário, desde que uma *estratégia de globalização* seja acionada. Ao determinar uma estratégia de globalização, duas abordagens clássicas são utilizadas: busca linear e regiões de confiança.

Quando se trabalha com busca linear, deve-se inicialmente determinar uma direção de descida. Uma direção d_k é chamada *direção de descida* para f a partir de x_k se possibilita determinar um valor $\bar{\lambda}$ tal que se $\lambda \in (0, \bar{\lambda})$, então $x_k + \lambda d_k$ provoca uma redução no valor da função objetivo. Sob o ponto de vista geométrico, direções de descida a partir de um determinado ponto são direções que formam um ângulo obtuso com o gradiente de f no ponto em questão.

Após a determinação de uma direção de descida, a estratégia de busca linear consiste em obter um escalar λ_k no intervalo $(0, 1]$, de forma que o ponto $x_{k+1} = x_k + \lambda_k d_k$ produza um decréscimo no valor de f. Devido às características das direções de descida, valores suficientementes próximos de zero para o tamanho do passo irão provocar o decréscimo requerido. No entanto, passos muito pequenos podem prejudicar a velocidade de convergência. Assim sendo, as estratégias de busca linear têm como objetivo encontrar um valor para tamanho do passo que se aproxime do máximo valor que permita a redução requerida no valor da função de mérito.

Por outro lado, se a opção é trabalhar com regiões de confiança, a função objetivo deve ser aproximada por um modelo, geralmente quadrático, numa região em torno do ponto x_k. A nova direção de busca será determinada pelos minimizadores do modelo em uma região em que se supõe que o modelo irá descrever bem a situação real.

O uso de uma estratégia de globalização só é eficaz se for exigido um decréscimo adequado no valor da função de mérito. Visando evitar a estagnação do algoritmo e/ou a convergência para pontos não estacionários, tradicionalmente se estabelece como critério de decréscimo suficiente que o novo ponto satisfaça a desigualdade proposta por Armijo [2].

A condição de Armijo exige que um ponto só pode ser aceito caso provoque um decréscimo na função de mérito maior do que o simples descenso. A diferença entre o descenso simples e o decréscimo proposto por Armijo depende do valor do ângulo entre a direção de descida utilizada e o gradiente da função objetivo:

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \gamma \lambda_k \nabla f(x_k)^\top d_k, \tag{1.9}$$

onde $\gamma \in (0, 1)$.

No entanto, alguns métodos de ponto fixo como, por exemplo, o método do resíduo espectral ([13], [14]) do qual falaremos adiante, tendem a um mau funcionamento quando são utilizadas muitas reduções no tamanho do passo. Essa tendência estimulou a utilização de estratégias mais flexíveis que Armijo. Assim sendo, trabalhos recentes como, por exemplo, [14] e [28], desenvolvem algoritmos do tipo espectral utilizando uma versão do critério de aceitação desenvolvido no trabalho de Grippo, Lampariello e Lucidi [27]. Nesse trabalho, os autores propõem e demonstram a convergência do método de Newton dotado de um critério de aceitação que permite a geração de sequências não-

monótonas, no sentido de que não necessariamente teremos $f(x_{k+1}) < f(x_k)$, em contraposição à proposta de Armijo que irá obrigatoriamente gerar uma sequência monótona.

Além disso, quando se está interessado em trabalhar com problemas em que as derivadas de F não podem ser avaliadas, as propostas para garantir a convergência global do problema que exijam o conhecimento do gradiente da função de mérito devem ser descartadas e, novamente, deve-se optar por esquemas em que o gradiente da função de mérito seja aproximado ou, então, uma estratégia livre de derivadas deve ser utilizada. Os trabalhos [13] e [37] apresentam propostas neste sentido.

O objetivo desta tese é tratar da resolução de sistemas não lineares de grande porte em que a função F é continuamente diferenciável. Essa hipótese, no entanto, não indica qualquer acesso a dados sobre a Jacobiana de F, mas tão somente visa garantir os resultados de convergência teórica.

A resolução dos sistemas não lineares é proposta através de uma abordagem híbrida, criando um método iterativo com duas fases:

1. uma versão *derivative-free* do método do ponto fixo utilizando *parâmetros espectrais* [5] para determinar o tamanho do passo da direção residual;

2. o método de Newton inexato em uma abordagem *matrix-free*, em que é acoplado o método GMRES [47] para resolver o sistema linear que determina a nova direção de busca.

O método combina ordenadamente as duas fases de forma que a segunda é acionada somente em caso de falha na primeira e, em ambas, uma condição de decréscimo não-monótono deve ser verificada para aceitação de novos pontos.

Proporemos ainda um segundo método, em que uma terceira fase de busca direta é acionada em situações em que o excesso de buscas lineares faz com que o tamanho de passo na direção Newton-inexata torne-se demasiadamente pequeno.

Métodos de busca direta são métodos iterativos e consistem em, a cada iteração, avaliar a função objetivo num número finito de pontos e determinar em quais pontos devemos continuar persistindo e quais pontos devem ser abandonados [12, 15, 35]. No nosso caso, utilizamos a chamada busca direta direcional, em que, a cada iteração k, um conjunto de direções, chamadas *direções de sondagem*, é determinado e são avaliados os pontos resultantes da soma de x_k com os elementos deste conjunto, escalados por um comprimento de passo.

Devido à sua taxa de convergência lenta, o uso de busca direta não é recomendável para resolução de sistemas não lineares de grande porte. Ainda assim, trabalhos recentes [28, 30] têm acoplado aos seus algoritmos uma etapa de busca direta quando os mesmos estejam em vias de declarar falhas de

execução por excesso de buscas lineares, permitindo assim melhorias práticas em termos de robustez. É essa abordagem que pretendemos adotar neste trabalho.

O texto é dividido em cinco capítulos. O segundo capítulo trata do método do resíduo espectral. Após uma revisão bibliográfica, é apresentado um relato teórico do método e são enunciados os principais resultados já conhecidos. Em seguida, apresentamos as mudanças propostas para aprimorar o método e finalizamos o capítulo com testes numéricos que comprovam o ganho obtido com algumas das mudanças realizadas, mas ainda indicando a necessidade de outras modificações que proporcionem um algoritmo mais robusto.

No terceiro capítulo, é apresentado o método híbrido proposto. Na primeira seção desse capítulo há uma abordagem teórica sobre o método Newton-GMRES, onde são relatados os principais resultados de convergência. Após isso, o método híbrido é descrito e são apresentados os resultados teóricos com respeito à convergência do método desenvolvido. O capítulo é finalizado com resultados práticos que comprovam o bom desempenho algorítmico do método.

Seguindo a tendência dos capítulos anteriores e, após uma descrição teórica dos métodos de busca direta e posterior análise dos principais resultados de convergência, o quarto capítulo apresenta um novo método híbrido que adiciona ao método anterior uma etapa de busca direta. Para essa etapa, desenvolvemos um novo conjunto de direções de sondagem. O capítulo segue com os resultados de convergência teórica dos algoritmos no contexto de resolução de sistemas não lineares.

Terminamos o texto com uma análise geral do trabalho desenvolvido, destacando as principais contribuições que apresenta, tanto do ponto de vista teórico quanto do ponto de vista computacional. Também relatamos algumas propostas de trabalhos futuros bem como resultados ainda em aberto.

Capítulo

2 Métodos espectrais

Utilizar métodos de ponto fixo é uma forma clássica de resolver sistemas não lineares. Métodos que adotam uma abordagem do tipo ponto fixo geram iterativamente a sequência $\{x_k\}$ utilizando como direção de busca $F(x_k)$ ou $-F(x_k)$, chamadas de *direções residuais*.

La Cruz e Raydan [14] propuseram o método do resíduo espectral (*Spectral Approach for Nonlinear Equations* - SANE), que elege como direções de buscas um vetor múltiplo da direção do resíduo e o vetor oposto a este:

$$d_k = (1/\alpha_k)F(x_k) \text{ e } d_k = -(1/\alpha_k)F(x_k). \tag{2.1}$$

O parâmetro de escalamento α_k é uma adaptação do parâmetro espectral, proposto por Barzilai e Borwein [5] no desenvolvimento do método do *gradiente espectral* utilizado para resolução de problemas de minimização irrestrita. Para garantir a convergência global do método, é empregado um critério de aceitação de pontos que permite a geração de uma sequência não monótona, isto é, considerando uma função de mérito $f : \mathbb{R}^n \to \mathbb{R}$, o valor de $\{f(x_k)\}$ não é necessariamente decrescente.

Posteriormente, La Cruz, Martínez e Raydan [13] dotaram o método do resíduo espectral com uma estratégia para aceitação de novos pontos que não faz uso explícito de qualquer tipo de derivada, definindo, assim, um método *derivative-free* (DFSANE), ao contrário do critério utilizado pelo método SANE, que utiliza o gradiente da função de mérito.

Iniciamos este capítulo com uma seção teórica, descrevendo o método do gradiente espectral desde a proposta inicial para a minimização de quadráticas em duas dimensões e passando para o posterior desenvolvimento de modificações que possibilitaram a resolução de problemas de minimização mais complexos. A seguir, relatamos como essas ideias foram aproveitadas para resolver sistemas não lineares, utilizando o algoritmo SANE [14] e sua posterior adaptação *derivative-free* DFSANE. São propostas alterações ao método DFSANE que visam estabelecer melhorias práticas para o método. O capítulo é finalizado com relatos dos testes numéricos que permitiram avaliar o desempenho das modificações propostas, além de fornecer indicativos para o prosseguimento da investigação.

Com o objetivo de solucionar problemas de minimização irrestrita:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{2.2}$$

onde a função $f:\mathbb{R}^n\to\mathbb{R}$ possui derivadas primeiras contínuas, Barzilai e Borwein [5] propuseram um método iterativo que utiliza um tamanho especial de passo para a direção de máxima descida.

O método foi pensado como um método quase-Newton [19, 40], que consiste em realizar iterações do tipo:

$$x_{k+1} = x_k - B_k^{-1}\nabla f(x_k), \tag{2.3}$$

onde B_k é uma matriz $n\times n$.

Para fins práticos, é conveniente que a matriz B_k possa ser atualizada sem grandes custos computacionais a partir da matriz B_{k-1}. Além disso, nos casos em que a função f é continuamente diferenciável, resultados teóricos garantem melhores taxas de convergência conforme B_k se aproxime da matriz Hessiana da função f no ponto x_k. Para resultados teóricos sobre os métodos quase-Newton, consultar [19, 40].

Geralmente, os métodos do tipo quase-Newton sugerem que a matriz B_k seja escolhida de modo a satisfazer a equação secante:

$$B_k s_{k-1} = y_{k-1}, \tag{2.4}$$

onde $s_{k-1} = x_k - x_{k-1}$ e $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$.

Como o interesse é trabalhar com a direção de máxima descida, Barzilai e Borwein [5] propõem o uso de uma matriz da forma $B_k = \alpha_k I$, em que α_k é determinado de maneira que B_k seja uma aproximação baseada em dois pontos para a equação secante (2.4).

O valor de α_k é, portanto, obtido como a solução do problema:

$$\min_{\alpha\in\mathbb{R}} \|\alpha s_k - y_k\|^2, \tag{2.5}$$

sendo denominado *parâmetro espectral* e dado pela expressão:

$$\alpha_k = \frac{s_k^\top y_k}{s_k^\top s_k}. \tag{2.6}$$

Observação 2.1: Analogamente, podemos encontrar uma aproximação para a inversa de B_k. Para isso, basta encontrar uma aproximação do tipo $B_k^{-1} = \alpha_k^{-1} I$ que seja a solução do seguinte problema:

$$\min_{\alpha \in \mathbb{R}} \left\| s_k - \alpha^{-1} y_k \right\|^2. \tag{2.7}$$

Neste caso, a solução será:

$$\alpha_k = \frac{y_k^\top y_k}{y_k^\top s_k}. \tag{2.8}$$

Na literatura é possível encontrar alguns métodos que utilizam somente o valor (2.6), outros métodos preferem alternar a escolha, utilizando ora o valor determinado pela expressão (2.6), ora o valor encontrado através da aproximação da inversa (2.8).

Para entender a denominação "parâmetro espectral", deve-se considerar o caso em que é adicionada a hipótese de que a função f é duas vezes continuamente diferenciável. Neste caso, o Teorema Fundamental do Cálculo permite determinar a seguinte relação:

$$\nabla f(x_k) - \nabla f(x_{k-1}) = \int_0^1 \nabla^2 f(x_{k-1} + t(x_k - x_{k-1}))dt(x_k - x_{k-1}), \tag{2.9}$$

e, substituindo (2.9) na equação (2.6), temos:

$$\alpha_k = \frac{s_k^\top y_k}{s_k^\top s_k} = \frac{s_k^\top \int_0^1 \nabla^2 f(x_{k-1} + t s_k)dt s_k}{s_k^\top s_k}. \tag{2.10}$$

O valor α_k assim definido é equivalente ao quociente de Rayleigh da Hessiana média, daí a justificativa para o uso do termo.

Em Barzilai e Borwein [5] é demonstrada a convergência r-superlinear para problemas de minimização quadráticos em duas dimensões. Além disso, para esse tipo de problema, é relatado menor esforço computacional e menor sensibilidade ao mau condicionamento em relação ao método com a direção de máxima descida clássica.

Para o caso geral, Fletcher [23] garante que podemos obter no máximo a convergência r-linear. Raydan [44] demonstra a convergência para quadráticas estritamente convexas em qualquer dimensão. Por fim, Friedlander, Martínez e Raydan [24] demonstram a convergência para quadráticas não necessariamente estritamente convexas.

Para garantir a convergência do método para o caso não-quadrático, a incorporação de uma estratégia de globalização é necessária. No entanto, o método do gradiente espectral tem como principal característica o fato de ser um método de iterações rápidas, pois requer somente $\mathcal{O}(n)$ operações em aritmética de ponto flutuante, e somente uma avaliação do gradiente da função a cada iteração.

Assim, para que o método preserve seu desempenho deve-se garantir que mantenha suas principais características:

Característica 1 *a direção com passo completo é aceita na maioria das vezes, já que o excesso de buscas lineares torna o método muito parecido com o de máxima descida, que por sua vez tem convergência lenta;*

Característica 2 *apenas informação de primeira ordem é armazenada.*

Com a finalidade de encontrar uma estratégia de globalização que não comprometa o desempenho do método de Barzilai e Borwein, Raydan [45] utiliza buscas lineares não-monótonas baseadas no critério de aceitação proposto por Grippo, Lampariello e Lucidi [27]:

$$f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min\{k, M-1\}} f(x_{k-j}) + \gamma \lambda \nabla f(x_k)^\top d_k, \tag{2.11}$$

com $\gamma \in (0,1)$ e $M \in \mathbb{N}$.

Ao adotar essa estratégia, a tendência é que o método aceite o passo espectral completo em algumas iterações em que este seria rejeitado se fosse adotada uma busca linear monótona. Com essa nova estratégia de aceitação e considerando f uma função continuamente diferenciável, conforme [45], a convergência para um ponto estacionário é garantida, desde que o número de pontos estacionários seja finito no conjunto $\Omega_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$.

Além disso, neste mesmo trabalho é demonstrado que, caso o conjunto de pontos estacionários de f não seja finito e $\nabla f(x_k) \neq 0$ para todo $k = 1, 2, 3, \ldots$, então $\lim_{k \to \infty} \|\nabla f(x_k)\| = 0$ e nenhum ponto limite da sequência é um ponto de máximo.

2.2. Método do resíduo espectral para sistemas não lineares

Com o objetivo de ampliar o conjunto de problemas resolvidos através de métodos espectrais, La Cruz e Raydan [14] propõem o método SANE (*Spectral Approach for Nonlinear Equations*), que consiste numa modificação do método espectral globalizado, com o intuito de adaptá-lo para resolução de sistemas não lineares de grande porte.

Esse método considera:

$$f(x) = \|F(x)\|^2 = F(x)^\top F(x) \tag{2.12}$$

como função de mérito e força a chegada de $F(x_k)$ a zero através da resolução do problema de minimizar o valor de f irrestritamente em $\mathbb{R}^n$. Um escalamento do vetor resíduo $F(x_k)$ e seu respectivo vetor oposto são utilizados como direções de busca

$$d_k = -(1/\alpha_k)F(x_k) \text{ e } d_k = (1/\alpha_k)F(x_k). \tag{2.13}$$

Utilizando como fator de escalamento α_k, uma adaptação do parâmetro espectral para problemas de minimização, é definido o método que recebe posteriormente a denominação *método do resíduo espectral* (ver [13]).

O parâmetro espectral, neste caso, é determinado pela equação:

$$\alpha_k = \frac{s_k^\top y_k}{s_k^\top s_k} = \frac{(x_k - x_{k-1})^\top (F(x_k) - F(x_{k-1}))}{(x_k - x_{k-1})^\top (x_k - x_{k-1})}. \tag{2.14}$$

Observação 2.2: Ainda que o método tenha se consolidado com o nome de resíduo espectral, cabe observar aqui que embora o Teorema Fundamental do Cálculo aplicado sobre a função $F : \mathbb{R}^n \to \mathbb{R}^n$ garanta que:

$$F(x_k) - F(x_{k-1}) = \int_0^1 J(x_{k-1} + t(x_k - x_{k-1}))dt(x_k - x_{k-1}), \tag{2.15}$$

e isso possibilite afirmar que a equação (2.14) corresponda ao quociente de Rayleigh da matriz Jacobiana média, não existe a garantia de que este valor esteja dentro do raio espectral, já que a simetria da matriz em questão é condição necessária para garantir que isso aconteça.

Para o caso de matrizes não simétricas, há um resultado que fornece um limitante superior para a distância entre um quociente de Rayleigh e um autovalor da matriz . Este resultado é formalizado a seguir:

Teorema 2.1 *Sejam $A \in \mathbb{C}^{n\times n}$ e v um autovetor de A associado ao autovalor λ e tal que $\|v\| = 1$. Considere $q \in \mathbb{C}^n$ com $\|q\| = 1$. Se α é igual a $q^H Aq$, o quociente de Rayleigh de q, então:*

$$|\lambda - \alpha| \leq 2\|A\|\|v - q\| \tag{2.16}$$

Prova: Ver [53] (página 326, Teorema 5.3.25). ∎

Assim, ao chamar o valor α_k determinado pela expressão (2.14) de parâmetro espectral, podemos dizer que estamos levando em consideração o fato de que este valor minimiza a expressão

$$\|\bar{J}_k s_k - \alpha s_k\|^2, \tag{2.17}$$

onde $\bar{J}_k$ é a matriz Jacobiana média da k-ésima iteração, e não mais o fato de que o valor α_k pertença ao intervalo determinado pelo raio espectral.

Denotando por $J(x_k)$ a matriz Jacobiana da função F no ponto x_k, teremos o gradiente da função de mérito dado por:

$$\nabla f(x_k) = 2J(x_k)^\top F(x_k), \tag{2.18}$$

e, daí

$$\nabla f(x_k)^\top d_k = 2F(x_k)^\top J(x_k)(-1/\alpha_k)F(x_k) \tag{2.19}$$

não é necessariamente negativo e, portanto, $-(1/\alpha_k)F(x_k)$ não é necessariamente uma direção de descida para f. Esta é a razão pela qual os autores propõem utilizar duas direções de busca e testar, a cada iteração, os pontos $x_+ = x_k - (1/\alpha_k)F(x_k)$ e $x_- = x_k + (1/\alpha_k)F(x_k)$.

Pelos mesmos motivos já discutidos no caso de problemas de minimização irrestrita, o algoritmo deve utilizar um critério de aceitação que possibilite a realização de uma busca linear não monótona, e opta-se pela utilização da estratégia (2.11).

Antes de prosseguir, é importante notar que a expressão (2.19) pode ser nula ou assumir um valor muito próximo de zero, sem que o vetor $F(x_k)$ esteja relativamente próximo do vetor nulo. Na prática, quando isso ocorre, pode não ser possível encontrar um valor para λ que satisfaça o critério de aceitação (2.11). Neste caso, o algoritmo irá estagnar ou fracassar durante o processo de busca linear. O método DFSANE [13], do qual falaremos a partir do próximo parágrafo, irá corrigir este problema.

Para trabalhar com métodos de grande porte em que a derivada não pode ser determinada pelos motivos que já adiantamos, La Cruz, Martínez e Raydan desenvolveram o método DFSANE [13], uma adaptação *derivative-free* para o algoritmo SANE, onde são preservados: o tamanho do passo determinado pelo parâmetro espectral e as direções de busca $d_+ = -(1/\alpha_k)F(x_k)$ e $d_- = (1/\alpha_k)F(x_k)$. A novidade nesse novo método é o critério de aceitação, agora modificado para uma estratégia que não depende do uso de derivadas.

Como já dissemos na seção anterior, uma boa estratégia de aceitação para pontos advindos de métodos de passos espectrais deve levar em conta a preservação das Características 1 e 2.

A estratégia (2.11) utilizada no algoritmo SANE, além de carregar a possibilidade de estagnação ou fracasso do método, exige ainda o conhecimento do valor do gradiente na função de mérito. Para realizar esse teste, se faz necessário o uso da matriz Jacobiana da função F no ponto x_k, que, no caso, não está disponível. Uma alternativa é o critério de aceitação desenvolvido por Li e Fukushima [37]:

$$\|F(x_k + \lambda_k d_k)\| \leq (1 + \zeta_k)\|F(x_k)\| - \gamma\lambda_k^2\|d_k\|^2, \tag{2.20}$$

com $\zeta_k > 0$ para todo k, $\sum_k \zeta_k = \zeta < \infty$ e $\gamma \in (0, 1)$.

No entanto, desde que o valor de ζ_k seja muito pequeno quando o valor de k for suficientemente grande, a estratégia irá impor uma condição quase monótona quando estivermos perto da solução. Isso não é desejável quando estamos trabalhando com gradientes espectrais ou resíduos espectrais pois o algoritmo irá perder uma de suas principais características, já que, neste caso, tende a se comportar como o método de máxima descida clássico.

La Cruz, Martínez e Raydan [13] optaram por mesclar as estratégias (2.11) e (2.20) e estabelecer um novo critério:

$$f(x_{k+1}) \leq \max_{0\leq j\leq \min\{k,M-1\}} f(x_{k-j}) + \zeta_k - \gamma\lambda_k^2 f(x_k) \tag{2.21}$$

evitando, assim, uma possível monotonicidade quando o algoritmo estiver se aproximando da solução.

Considere x_0 um ponto arbitrário de $\mathbb{R}^n$ e suponha definidos $\alpha_{\max}$, $\alpha_{\min} \in \mathbb{R}$ com $0 < \alpha_{\min} < \alpha_{\max}$, $M \in \mathbb{N}$, $\gamma \in (0,1)$ e uma sequência de escalares $\{\zeta_k\}$ tal que $\zeta_k > 0$ para todo $k \in \mathbb{N}$ e $\sum_k \zeta_k = \zeta < \infty$. O Algoritmo 2.1 descreve a obtenção do ponto x_{k+1} a partir de x_k através do método DFSANE, conforme proposto em [13].

Algoritmo 2.1 *DFSANE*

1. - *Escolha α_k tal que $|\alpha_k| \in [\alpha_{\min}, \alpha_{\max}]$ e faça $d = -(1/\alpha_k)F(x_k)$.*
   - *Faça $\lambda_+ = \lambda_- = 1$.*
   - *Defina $\tilde{f} = \max_{0\leq j\leq \min\{k,M-1\}} f(x_{k-j})$.*
2. *Se $f(x_k + \lambda_+ d) \leq \tilde{f} + \zeta_k - \gamma\lambda_+^2 f(x_k)$,*

   defina $d_k = d$, $\lambda_k = \lambda_+$ e $x_{k+1} = x_k + \lambda_k d_k$ e vá para o passo 3.

   Caso contrário, se $f(x_k - \lambda_- d) \leq \tilde{f} + \zeta_k - \gamma\lambda_-^2 f(x_k)$,

   defina $d_k = -d$, $\lambda_k = \lambda_-$ e $x_{k+1} = x_k + \lambda_k d_k$ e vá para o passo 3.

   Caso contrário, reduza λ_+ e λ_- e vá para o início do passo 2.
3. *Se $F(x_{k+1}) = 0$ termine o algoritmo. Caso contrário, faça $k = k + 1$ e vá para o passo 1.*

Para que os resultados de convergência do método tenham efeito, deve-se assumir as Hipóteses 2.1, 2.2 e, 2.3:

Hipótese 2.1 *A função $F : \mathbb{R}^n \to \mathbb{R}^n$ tem derivadas parciais contínuas.*

Hipótese 2.2 *A sequência $\{\zeta_k\}$ é tal que $\zeta_k > 0$ para todo $k \in \mathbb{N}$ e $\sum_{k=1}^{\infty} \zeta_k = \zeta < \infty$.*

Hipótese 2.3 *A função de mérito $f : \mathbb{R}^n \to \mathbb{R}$ deve ser definida como:*

$$f(x) = \|F(x)\| \quad \forall x \in \mathbb{R}^n \qquad ou \qquad f(x) = \|F(x)\|^2 \quad \forall x \in \mathbb{R}^n. \tag{2.22}$$

Assumindo as Hipóteses 2.1, 2.2 e 2.3 acima, La Cruz, Martínez e Raydan [13] estabeleceram o seguinte resultado de convergência:

Teorema 2.2 *1. Se $\{x_k\}$ é a sequência gerada pelo método DFSANE e K é o conjunto de índices $\{\nu(1)-1, \nu(2)-1, \nu(3)-1, \ldots\}$, onde $\nu(k) \in \{(k-1)M+1, \ldots, kM\}$ e é tal que, para todo $k = 1, 2, \ldots$, tem-se $f(x_{\nu(k)}) = W_k$. Então todo ponto limite de $\{x_k\}_{k\in K}$ satisfaz $F(x^*)^\top J(x^*)^\top F(x^*) = 0$.*

2. Se existe um ponto limite x^* da sequência gerada pelo algoritmo `DFSANE` *tal que $F(x^*) = 0$, então $\lim_{k\to\infty} F(x_k) = 0$.*

3. Na situação anterior, se existir $\delta > 0$ tal que $F(x) \neq 0$ para todo $x \neq x^*$ satisfazendo $\|x - x^*\| \leq \delta$, então $\lim_{k\to\infty} x_k = x^*$.

4. Sob as hipóteses do item 2, se existir $\varepsilon > 0$ tal que $F(x)^\top J(x)^\top F(x) \neq 0$ para todo $x \neq x^*$ satisfazendo $\|x - x^*\| \leq \varepsilon$, então existe um valor $\delta > 0$ tal que $\|x_0 - x^*\| \leq \delta$ implica que $\lim_{k\to\infty} x_k = x^*$.

Prova: Ver [13], onde cada item acima corresponde, respectivamente, aos Teoremas 1, 2, 3 e 4. ■

Resultados mais fortes podem ser obtidos como consequências dos resultados acima se considerarmos algumas hipóteses sobre a matriz Jacobiana de F. Segue do item 1 do Teorema 2.2 o seguinte corolário:

Corolário 2.1 *Se $J(x)$ é definida positiva ou definida negativa e o conjunto de nível $\{x \in \mathbb{R} | f(x) \leq f(x_0) + \zeta\}$ é limitado, então a sequência gerada pelo algoritmo* `DFSANE` *admite uma subsequência que converge para a solução do sistema não-linear (1.1).*

Prova: Ver [13], Corolários 1, 2 e 3. ■

Se a matriz Jacobiana de F é definida positiva ou definida negativa em alguma solução do sistema não-linear, o resultado do item 4 do Teorema 2.2 irá implicar em um segundo corolário:

Corolário 2.2 *Seja $\{x_k\}$ a sequência gerada pelo algoritmo* `DFSANE` *e assuma que x^* é uma solução para $F(x) = 0$ e, além disso, $J(x^*)$ é definida positiva ou definida negativa. Então existe um valor $\delta > 0$ tal que $\|x_0 - x^*\| \leq \delta$ implica que $\lim_{k\to\infty} x_k = x^*$.*

Prova: Ver [13], Corolário 4. ■

Cabe observar que os resultados de convergência obtidos para o método DFSANE, conforme o Algoritmo 2.1 e descritos pelo Teorema 2.2, em momento algum supõem um valor particular para α_k. As propriedades de convergência são garantidas para qualquer escolha de α_k limitada pelos parâmetros $\alpha_{\max}$ e $\alpha_{\min}$. Ou seja, embora o método DFSANE esteja configurado para ser um método do tipo espectral, os resultados obtidos não exigem que o algoritmo tenha essa característica. No entanto, conforme os autores relatam em [13], na prática, outras escolhas para α_k não funcionaram tão bem quanto a escolha do parâmetro espectral.

2.3. Modificações propostas

Com o objetivo de melhorar a robustez do algoritmo sem, no entanto, prejudicar a eficiência demonstrada nos testes presentes em [13], propusemos algumas alterações no algoritmo DFSANE. Tais modificações ocorreram em duas vertentes:

1. elaboração de estratégias para ordenação da avaliação dos pontos resultantes do uso das direções do resíduo espectral;

2. proposta de uma terceira direção a testar em caso de falha nas direções $d_+ = -(1/\alpha_k)F(x_k)$ e $d_- = (1/\alpha_k)F(x_k)$.

2.3.1 Estratégias de ordenação

A utilização do método do resíduo espectral para resolução de sistemas não lineares com o uso da estratégia de busca não-monótona, conforme o Algoritmo 2.1 DFSANE, pode nos levar a optar por uma direção de subida ao invés de uma direção de descida. Isso ocorre porque o método testa sempre o ponto $x_k - \lambda_+(1/\alpha_k)F(x_k)$ antes do ponto $x_k + \lambda_-(1/\alpha_k)F(x_k)$. A direção $-(1/\alpha_k)F(x_k)$ pode ser de subida e ainda assim satisfazer o critério de aceitação (2.21), já que este critério não exige necessariamente um decréscimo na função de mérito. Essa situação foi constatada na prática e tende a ocorrer principalmente nas iterações onde se tem um valor grande para ζ_k. A seguir exibimos um exemplo onde isso ocorre.

Exemplo 2.1 *Considere*

$$F : \mathbb{R}^2 \to \mathbb{R}^2$$
$$F(x_1, x_2) = \begin{bmatrix} -0.3x_1 \\ -0.5x_2 \end{bmatrix}.$$

Neste caso, teremos como função de mérito $f(x_1, x_2) = \|F(x_1, x_2)\|^2 = 0.09x_1^2 + 0.25x_2^2$. *Usando os parâmetros iniciais* $\alpha_0 = 1$, $\lambda_+ = \lambda_- = 1$, $\gamma = 10^{-4}$, *definindo* $\zeta_k = 1/(k+1)^2$ *e tomando* $x_0 = (0.1, 0.1)^\top$, *temos que* $f(x_0) = 3.4 \times 10^{-3}$ *e, assim,* $\tilde{f} + \zeta_k - \gamma\lambda_+^2 f(x_k) = 1.0034$.

Na primeira iteração, o método deverá encontrar a direção:

$$d = -(1/\alpha_k)F(x_k) = -(-0.03, -0.05)^\top. \tag{2.23}$$

E, consequentemente, teremos $x_+ = (0.13, 0.15)^\top$ *e* $x_- = (0.07, 0.05)^\top$.

Aplicando a função de mérito em cada um desses pontos, obtemos os valores $f(x_+) = 0.07416$ *e* $f(x_-) = 0.01066$. *Como o ponto* x_+ *satisfaz o critério de aceitação, o algoritmo irá prosseguir para esse ponto, adotando uma direção desfavorável e afastando-se da solução do problema,* $x^* = (0, 0)^\top$, *pois* $\|x_0 - x^*\| < \|x_+ - x^*\|$ *enquanto* $\|x_0 - x^*\| > \|x_- - x^*\|$.

Considerando esta dificuldade, propomos que, ao invés de testar pela ordem usual, ou seja, primeiramente o ponto proveniente da direção $-(1/\alpha_k)F(x_k)$ e depois a direção oposta, utilizemos dois novos tipos de ordenação:

1. avaliar a função de mérito f nos pontos $x_k - (1/\alpha_k)F(x_k)$ e $x_k + (1/\alpha_k)F(x_k)$ e, após isso, realizar o teste de aceitação somente no ponto que obtiver a melhor redução no valor de f (`DFSANE-O1`);

2. testar primeiramente a direção no sentido utilizado na iteração anterior. Isto é, se o ponto x_k for obtido a partir de $d_{k-1} = -(1/\alpha_{k-1})F(x_{k-1})$, testa-se primeiramente o ponto proveniente da direção $-(1/\alpha_k)F(x_k)$ e, caso contrário, realiza-se o primeiro teste de aceitação utilizando o ponto proveniente da direção $(1/\alpha_k)F(x_k)$ (`DFSANE-O2`).

Os passos para obter x_{k+1} a partir de x_k para as duas modificações propostas estão detalhados nos Algoritmos 2.2 e 2.3. Em ambos os algoritmos, novamente devemos considerar x_0 um ponto arbitrário de $\mathbb{R}^n$ e supor definidos, $\alpha_{\max}, \alpha_{\min} \in \mathbb{R}$ com $0 < \alpha_{\min} < \alpha_{\max}$, $M \in \mathbb{N}$, $\gamma \in (0, 1)$ e uma sequência de escalares $\{\zeta_k\}$ tal que $\zeta_k > 0$ para todo $k \in \mathbb{N}$ e $\sum_k \zeta_k = \zeta < \infty$.

Algoritmo 2.2 *DFSANE-O1*

1.
- *Escolha* α_k *tal que* $|\alpha_k| \in [\alpha_{\min}, \alpha_{\max}]$ *e faça* $d = -(1/\alpha_k)F(x_k)$.
- *Faça* $\lambda_+ = \lambda_- = 1$.
- *Defina* $\tilde{f} = \max_{0 \le j \le \min\{k, M-1\}} f(x_{k-j})$.

2. *Calcule* $f(x_k+\lambda_+ d)$ *e* $f(x_k-\lambda_- d)$. *Se* $f(x_k+\lambda_+ d) \leq f(x_k-\lambda_- d)$ *vá para 3a. Caso contrário, vá para 3b.*

3. (a) *Se* $f(x_k+\lambda_+ d) \leq \tilde{f} + \zeta_k - \gamma\lambda_+^2 f(x_k)$, *defina* d_k=d, $\lambda_k = \lambda_+$ *e* $x_{k+1} = x_k + \lambda_k d_k$, *e vá para o passo 4.*

   Caso contrário, vá para 3c.

   (b) *Se* $f(x_k-\lambda_- d) \leq \tilde{f} + \zeta_k - \gamma\lambda_-^2 f(x_k)$, *defina* d_k=$-d$, $\lambda_k = \lambda_-$ *e* $x_{k+1} = x_k + \lambda_k d_k$, *e vá para o passo 4.*

   Caso contrário, vá para 3c.

   (c) *Diminua* λ_+ *e* λ_-, *e vá para o passo 2.*

4. *Se* $F(x_{k+1}) = 0$ *termine o algoritmo. Caso contrário, faça* $k = k+1$ *e vá para o passo 1.*

Algoritmo 2.3 *DFSANE-O2*

1. *Defina* $lastpos = true$.

2. - *Escolha* α_k *tal que* $|\alpha_k| \in [\alpha_{\min}, \alpha_{\max}]$ *e faça* $d = -(1/\alpha_k)F(x_k)$.
   - *Faça* $\lambda_+ = \lambda_- = 1$.
   - *Defina* $\tilde{f} = \max_{0\leq j\leq \min\{k,M-1\}} f(x_{k-j})$.

3. *Defina* $first = true$. *Se* $lastpos = true$, *vá para 4a. Caso contrário, vá para 4b.*

4. (a) *Se* $f(x_k+\lambda_+ d) \leq \tilde{f} + \zeta_k - \gamma\lambda_+^2 f(x_k)$, *defina* d_k=d, $\lambda_k = \lambda_+$ *e* $x_{k+1} = x_k + \lambda_k d_k$, $lastpos = true$ *e vá para o passo 5.*

   Caso contrário, $\begin{cases} \text{se } first = true, \text{ defina } first = false \text{ e vá para 4b.} \\ \text{se } first = false \text{ vá para 4c.} \end{cases}$

   (b) *Se* $f(x_k-\lambda_- d) \leq \tilde{f} + \zeta_k - \gamma\lambda_-^2 f(x_k)$, *defina* d_k=$-d$, $\lambda_k = \lambda_-$ *e* $x_{k+1} = x_k + \lambda_k d_k$, $lastpos = false$ *e vá para o passo 5.*

   Caso contrário, $\begin{cases} \text{se } first = true, \text{ defina } first = false \text{ e vá para 4a.} \\ \text{se } first = false \text{ vá para 4c.} \end{cases}$

   (c) *Diminua* λ_+ *e* λ_-, *e vá para o passo 3.*

5. *Se* $F(x_{k+1}) = 0$, *termine o algoritmo. Caso contrário, faça* $k = k+1$ *e vá para o passo 2.*

É claro que a estratégia `DFSANE-O1` elimina completamente o problema de prosseguir na direção que não proporciona o maior decréscimo na função de mérito. No entanto, temos de nos atentar que o número de avaliações de função deve certamente aumentar, podendo até dobrar nos piores casos. Já a estratégia `DFSANE-O2`, embora não garanta a total eliminação do problema, não prevê qualquer custo adicional em termos de avaliações de função.

Tanto a estratégia DFSANE-01 quanto a estratégia DFSANE-02 preservam os resultados de convergência obtidos pelo método original, já que em momento algum a ordem pela qual se realiza o teste de aceitação interfere na demonstração teórica dos resultados. Sendo assim, o Teorema 2.2 pode ser reescrito, a fim de ampliar os resultados para os dois novos algoritmos:

Teorema 2.3 *1. Seja $\{x_k\}$ é a sequência gerada pelo método DFSANE, ou por uma de suas versões modificadas* DFSANE-01 *e* DFSANE-02*. Se K é o conjunto de índices $\{\nu(1)-1, \nu(2)-1, \nu(3)-1, \ldots\}$, onde $\nu(k) \in \{(k-1)M+1, \ldots, kM\}$ e é tal que, para todo $k = 1, 2, \ldots$, tem-se $f(x_{\nu(k)}) = W_k$. Então todo ponto limite de $\{x_k\}_{k\in K}$ satisfaz $F(x^*)^\top J(x^*)^\top F(x^*) = 0$.*

2. Se existe um ponto limite x^* da sequência gerada pelo algoritmo DFSANE*, ou por uma de suas versões modificadas* DFSANE-01 *e* DFSANE-02*, tal que $F(x^*) = 0$, então $\lim_{k\to\infty} F(x_k) = 0$.*

3. Na situação anterior, se existir $\delta > 0$ tal que $F(x) \neq 0$ para todo $x \neq x^*$ satisfazendo $0 < \|x - x^*\| \leq \delta$, então $\lim_{k\to\infty} x_k = x^*$.

4. Sob as hipóteses do item 2, se existir $\varepsilon > 0$ tal que $F(x)^\top J(x)^\top F(x) \neq 0$ para todo $x \neq x^*$ satisfazendo $0 < \|x - x^*\| \leq \varepsilon$, então existe um valor $\delta > 0$ tal que $\|x_0 - x^*\| \leq \delta$ implica que $\lim_{k\to\infty} x_k = x^*$.

Prova: Ver [13], onde cada item acima corresponde, respectivamente, aos Teoremas 1, 2, 3 e 4. ■

Também são preservados os resultados dos Corolários 2.1 e 2.2:

Corolário 2.3 *Se $J(x)$ é definida positiva ou definida negativa e o conjunto de nível $\{x \in \mathbb{R} | f(x) \leq f(x_0) + \zeta\}$ é limitado, então a sequência gerada pelo algoritmo* DFSANE*, ou por uma de suas versões modificadas* DFSANE-01 *e* DFSANE-02*, admite uma subsequência que converge para a solução do sistema não-linear (1.1).*

Prova: Ver [13], Corolários 1, 2 e 3. ■

Corolário 2.4 *Seja $\{x_k\}$ a sequência gerada pelo algoritmo* DFSANE*, ou por uma de suas versões modificadas* DFSANE-01 *e* DFSANE-02*, e assuma que x^* é uma solução para $F(x) = 0$ e, além disso, $J(x^*)$ é definida positiva ou definida negativa. Então existe um valor $\delta > 0$ tal que $\|x_0 - x^*\| \leq \delta$ implica que $\lim_{k\to\infty} x_k = x^*$.*

Prova: Ver [13], Corolário 4. ■

2.3.2 Novas direções de busca

Nesta parte do trabalho desenvolvemos um estudo sobre direções alternativas que poderiam ser adicionadas ao Algoritmo DFSANE em caso de falha nos dois sentidos tradicionalmente testados [16].

Isto é, caso os pontos $x_k - \lambda_+(1/\alpha_k)F(x_k)$ e $x_k + \lambda_-(1/\alpha_k)F(x_k)$ não proporcionem uma redução satisfatória, uma terceira direção é testada antes de começar uma nova redução no tamanho do passo. A nova direção não deve ter um alto custo de cálculo, nem excessivos requerimentos de memória, mantendo assim as principais características dos métodos espectrais.

Na primeira abordagem, a qual chamamos `DFSANE-D3`, elaboramos uma aproximação do tipo diferenças centrais para um múltiplo do gradiente da função $\|F(x)\|$, quando este existir:

$$d_3 = \frac{F(x_k - \lambda_+(1/\alpha_k)F(x_k)) - F(x_k + \lambda_-(1/\alpha_k)F(x_k))}{(\lambda_+ + \lambda_-)\|(1/\alpha_k)F(x_k)\|}. \tag{2.24}$$

Essa abordagem apoia-se no fato de que, diante da situação em que sejam necessárias muitas reduções no valor λ_+ e λ_-, a aproximação de diferenças centrais para a terceira direção vá ficando mais precisa.

É importante notar que os valores para $F(x_k - \lambda_+(1/\alpha_k)F(x_k))$ e para $F(x_k + \lambda_-(1/\alpha_k)F(x_k))$ já foram previamente calculados, visto que a direção d_3 só será empregada quando houver falha nos dois pontos obtidos anteriormente. Com isso, o custo de obter a direção d_3 é relativamente baixo.

Uma segunda versão usa esta mesma ideia, agora numa abordagem componente a componente. Neste caso, teremos:

$$d_4 = (d^1, \ldots, d^n)^\top, \tag{2.25}$$

em que, para $i = 1, 2, \ldots, n$:

$$d^i = \frac{F_i(x_k - \lambda_+(1/\alpha_k)F(x_k)) - F_i(x_k + \lambda_-(1/\alpha_k)F(x_k))}{(\lambda_+ + \lambda_-)|(1/\alpha_k)F_i(x_k)|}. \tag{2.26}$$

Essa nova estratégia será denotada por `DFSANE-D4`.

A descrição do funcionamento do método DFSANE dotado de uma terceira direção alternativa pode ser conferido no Algoritmo 2.4, no qual a direção d_{new} pode ser tanto d_3 quanto d_4.

Algoritmo 2.4 *DFSANE com terceira direção*

1. - *Escolha* α_k *tal que* $|\alpha_k| \in [\alpha_{\min}, \alpha_{\max}]$ *e faça* $d = -(1/\alpha_k)F(x_k)$.
   - *Faça* $\lambda_+ = \lambda_- = 1$.
   - *Defina* $\tilde{f} = \max_{0 \le j \le \min\{k, M-1\}} f(x_{k-j})$.
2. *Se* $f(x_k + \lambda_+ d) \le \tilde{f} + \zeta_k - \gamma\lambda_+^2 f(x_k)$,
   defina $d_k = d$, $\lambda_k = \lambda_+$ *e* $x_{k+1} = x_k + \lambda_k d_k$ *e vá para o passo 7.*
3. *Se* $f(x_k - \lambda_- d) \le \tilde{f} + \zeta_k - \gamma\lambda_-^2 f(x_k)$,
   defina $d_k = -d$, $\lambda_k = \lambda_-$ *e* $x_{k+1} = x_k + \lambda_k d_k$ *e vá para o passo 7.*

4. *Defina d_{new} de acordo com a equação (2.24) ou com a equação (2.25).*

5. *Se $f(x_k + d_{new}) \leq \tilde{f} + \zeta_k - \gamma \left(\frac{\lambda_+ + \lambda_-}{2}\right)^2 f(x_k)$,*
   defina $d_k = d_{new}$ e $x_{k+1} = x_k + d_{new}$ e vá para o passo 7.

6. *Reduza λ_+ e λ_- e vá para o início do passo 2.*

7. *Se $F(x_{k+1}) = 0$ termine o algoritmo. Caso contrário, faça $k = k + 1$ e vá para o passo 1.*

Diferentemente das propostas envolvendo ordenação, as modificações no algoritmo DFSANE que envolvem o uso de uma terceira direção além das duas usuais, não preservam as demonstrações dos resultados encontrados em [13] para o algoritmo `DFSANE`. Isso ocorre porque existe a possibilidade de os novos algoritmos gerarem uma subsequência de forma que exista um determinando valor $\bar{k} \in \mathbb{N}$ tal que todo ponto x_k, $k \geq \bar{k}$, foi obtido utilizando a direção d_3 ou d_4. Quando isso acontece, algumas propriedades essenciais para a demonstração do Teorema 2.2, como, por exemplo, o fato de que $\|x_{k+1} - x_k\| \to 0$ quando $k \to \infty$, não são trivialmente satisfeitas.

Uma hipótese que contorna esse problema é a limitação do número de vezes que as direções d_3 ou d_4 podem ser acionadas. Essa limitação, na prática, não interfere no desempenho dos algoritmos `DFSANE-D3` e `DFSANE-D4`, já que o número limitante poderia ser, por exemplo, o número máximo de avaliações de função permitido.

2.4. Problemas-teste

A fim de analisar a qualidade de nossas modificações perante o algoritmo original e outros algoritmos já consolidados na área, realizamos uma pesquisa nos artigos que tratavam da resolução de sistemas não lineares publicados nos últimos anos para definir um conjunto de problemas-teste [16].

Após esta revisão bibliográfica, foi possível classificar os algoritmos em dois tipos distintos: aqueles que se propõem a resolver problemas de pequeno porte (dimensão n variando entre 1 e 30) e aqueles que se propõem a resolver problemas de grande porte, que é o objetivo principal desta tese.

Os problemas direcionados a realizar testes com algoritmos que propõem resolução de problemas de grande porte estão, em sua maioria, compilados em La Cruz e Raydan [14] e na seção 4 de Lukšan e Vlček [39].

Nas Tabelas A.1, A.2, A.3 e A.4, encontradas no Apêndice deste texto, especificamos os problemas-teste propostos em cada trabalho, bem como o valor padrão sugerido para o ponto inicial. Além disso, as Tabelas A.1, A.2 ainda disponibilizam os valores das dimensões testadas em cada problema, cf. [14].

Neste capítulo, optamos por realizar os testes computacionais com o conjunto de problemas descritos em [14].

2.5. Ferramenta de análise de desempenho

Para analisar o desempenho dos algoritmos criados, utilizamos a ferramenta perfil de desempenho, proposta em 2002 por Dolan e Moré [20]. Essa ferramenta é ideal para comparar a eficiência e a robustez entre vários algoritmos que são aplicados ao mesmo conjunto de problemas, sob as mesmas condições.

Escolhida uma medida de desempenho (no nosso caso foram três: tempo de execução, número de avaliação de funções, número de iterações externas), devemos determinar a taxa de desempenho do algoritmo s na resolução do problema p. Definindo S e P o conjunto de todos os algoritmos e problemas respectivamente e ainda $m_{s,p}$ a medida de desempenho do algoritmo s para o problema p, sua taxa de desempenho será dada por:

$$r_{s,p} = \begin{cases} \frac{m_{s,p}}{\min\{m_{s,p}, \forall s \in S\}}, & \text{se o algoritmo } s \text{ resolveu o problema } p \\ r_M, & \text{caso contrário,} \end{cases} \tag{2.27}$$

onde r_M é um parâmetro pré-definido e suficientemente grande.

Para cada problema, devemos construir uma função acumulativa $\rho_s : \mathbb{R} \to [0, 1]$. Considerando $\#(A)$ a cardinalidade de um conjunto A, essa função é definida como:

$$\rho_s(\tau) = \frac{\#(p \in P \mid r_{s,p} \leq \tau)}{\#(P)}. \tag{2.28}$$

Obviamente, esta função é não-decrescente, constante por partes e não tem sentido para $\tau < 1$. A eficiência do algoritmo pode ser analisada pelo valor dado por $\rho_s(1)$. Algoritmos com maiores valores para $\rho_s(1)$ são mais eficientes. Para analisar a robustez devemos observar o valor de τ para o qual teremos $\rho_s(\tau) = 1$; quanto menor for esse valor mais robusto será o algoritmo. Atentemo-nos para o fato de que, se o algoritmo s não obtiver convergência para ao menos um problema, teremos $\rho_s(\tau) = 1$ somente para $\tau = r_M$. Nesses casos, existem valores $C \in (0, 1)$ e $\bar{\tau} \in (1, r_M)$ tais que

$\rho(\tau) = C$ para $\tau \in [\bar{\tau}, r_M)$, o valor C é a proporção de problemas para os quais o algoritmo analisado obteve convergência.

Neste trabalho, apresentamos os gráficos de cada medida de desempenho com dois intervalos distintos para o valor de τ. O primeiro deles tem o objetivo de destacar o comportamento das curvas dos algoritmos próximas de $\tau = 1$ e facilitar a análise da eficiência, sendo os gráficos das funções ρ_s traçados no intervalo $1 \leq \tau \leq 2$. O segundo deles mostra as curvas no intervalo $1 \leq \tau \leq 10$, permitindo a detecção da estagnação das curvas de desempenho e uma melhor análise da robustez.

2.6. Testes iniciais

Inicialmente, nossa intenção foi quantificar o impacto que as estratégias desenvolvidas nesse capítulo tiveram em relação ao algoritmo `DFSANE`.

Realizamos um teste comparativo em que foram mantidos do algoritmo original: o critério de aceitação de passo (2.21), as estratégias de redução do tamanho do passo, a escolha da sequência $\{\zeta_k\}$, os critérios de parada, bem como todos os demais parâmetros. Tais critérios e parâmetros serão descritos nos próximos parágrafos.

Havendo necessidade de redução do tamanho do passo, o novo valor é obtido através de uma interpolação quadrática, utilizando a matriz identidade como aproximação da matriz Jacobiana. Por exemplo, havendo necessidade de obter um novo valor λ_+ no passo 2 do Algoritmo 2.1, determinamos um polinômio de grau 2 interpolante para a função

$$\begin{aligned} &\varphi : [0, \lambda_+] \to \mathbb{R} \\ &\varphi(\lambda) = f(x_k - \lambda(1/\alpha_k)F(x_k)). \end{aligned} \tag{2.29}$$

Feito isso, encontramos o valor λ_{new} como a solução do problema de minimizar a função (2.29) e definimos o novo valor para o tamanho do passo, adicionando a seguinte salvaguarda:

$$\lambda_+ = \begin{cases} \tau_{\min}\lambda_+, & \text{se } \lambda_{new} < \tau_{\min}\lambda_+ \\ \tau_{\max}\lambda_+, & \text{se } \lambda_{new} > \tau_{\max}\lambda_+ \\ \lambda_{new}, & \text{caso contrário,} \end{cases} \tag{2.30}$$

utilizando os valores $\tau_{\min} = 0.1$ e $\tau_{\max} = 0.5$.

Observação 2.3: Outras estratégias para redução do tamanho do passo foram desenvolvidas. A primeira proposta considerava, após a segunda iteração, um modelo com interpolação em três pontos, eliminando, assim, a necessidade de utilizar aproximações para a matriz Jacobiana de F.

Também foi testada uma estratégia que consistia em criar o modelo na região entre $x_k - \lambda_+(1/\alpha_k)F(x_k)$ e $x_k + \lambda_-(1/\alpha_k)F(x_k)$ e encontrar o minimizador nesta região. Desta forma, era possível prosseguir para a próxima iteração em apenas uma direção (observe que, aqui, o minimizador do modelo quadrático pode assumir um número negativo). A rigor, a estratégia indicava que as duas direções fossem testadas se o minimizador encontrado estivesse demasiadamente próximo de zero.

No entanto, nenhuma dessas estratégias obteve bons desempenhos na prática e, por esse motivo, foram abandonadas.

Já a sequência $\{\zeta_k\}$ foi definida de tal forma que

$$\zeta_k = \frac{\|F(x_0)\|}{(1+k)^2}, \quad \text{para todo } k \in \mathbb{N}.$$

Para assegurar uma limitação para o valor de α_k, utilizamos o intervalo $[10^{-10}, 10^{10}]$. Caso o valor determinado pelo parâmetro espectral (2.6) esteja fora desse intervalo, um novo valor para α_k é definido de modo a ficar dependente da norma de F no ponto x_k:

$$\alpha_k = \begin{cases} 1, & \text{se } \|F(x_k)\| > 1, \\ \|F(x_k)\|, & \text{se } 10^{-5} \leq \|F(x_k)\| \leq 1, \\ 10^{-5}, & \text{se } \|F(x_k)\| < 10^{-5}. \end{cases} \tag{2.31}$$

Ademais, utilizamos $f(x) = \|F(x)\|^2$ como função de mérito, $\alpha_0 = 1$, $\gamma = 10^{-4}$, $M = 10$ e, como critério de parada:

$$\frac{\|F(x_k)\|}{\sqrt{n}} \leq \varepsilon_a + \frac{\varepsilon_r \|F(x_0)\|}{\sqrt{n}}, \tag{2.32}$$

com $\varepsilon_a = 10^{-5}$ e $\varepsilon_r = 10^{-4}$, reiteramos que esse critério de parada, assim como os demais parâmetros utilizados, segue o que foi feito em [13].

Os testes foram realizados utilizando o *software* Matlab 7.0, em uma máquina com processador Intel(R) Core I3-2100 3.10GHz e 4Gb de memória RAM. Cada problema foi resolvido utilizando como valor inicial, o ponto padrão para cada problema indicado em [14] e também para as três dimensões testadas no mesmo trabalho. Essas informações podem ser encontradas nas Tabelas A.1 e A.2.

Os critérios utilizados para interromper a execução do algoritmo e as notações adotadas neste capítulo foram:

EST parada por estagnação, quando atingir o máximo de 100 reduções no tamanho do passo;

EAVF parada por excesso de avaliações de função, quando atingir a quantidade de 10000 avaliações de função;

PPF parada por dificuldades numéricas decorrentes da aritmética de ponto flutuante, ocasionando *overflow* ou *underflow*.

C parada com sucesso, quando a condição (2.32) é verificada.

A Tabela 2.1 resume o desempenho, em termos de robustez, dos algoritmos modificados neste conjunto inicial de 60 problemas.

	EST	**EAVF**	**PPF**	**C**
DFSANE	0%	21.667%	5%	73.333%
DFSANE-O1	0%	23.333%	3.333%	73.333%
DFSANE-O2	0%	21.667%	3.333%	75%
DFSANE-D3	0%	25%	1.667%	73.33%
DFSANE-D4	0%	26.667%	0%	73.333%

Tabela 2.1: Resumo dos resultados - problemas extraídos de [14] - valor inicial padrão

Além disso, foram feitos dois conjuntos de gráficos de perfil de desempenho para avaliar os algoritmos modificados neste mesmo conjunto de testes. O primeiro conjunto de gráficos encontra-se na Figura 2.1 e mostra o desempenho dos algoritmos em que foram adicionadas as estratégias de ordenação propostas neste trabalho (DFSANE-O1 e DFSANE-O2) e o algoritmo DFSANE. Já a Figura 2.2 apresenta o segundo conjunto de gráficos, onde é possível comparar o desempenho dos algoritmos com novas opções de direção de busca (DFSANE-D3 e DFSANE-D4) e o algoritmo original.

Pela Figura 2.1, podemos observar um desempenho melhor em termos de eficiência do algoritmo DFSANE-O2 sobre o algoritmo DFSANE nas três medidas de desempenho analisadas: iterações, número de avaliações de função e tempo de execução. Essa melhora de eficiência aliada ao fato do algoritmo DFSANE-O2 ser ligeiramente mais robusto que o algoritmo original, como pode ser visto pela Tabela 2.1, nos permite estabelecer o algoritmo DFSANE-O2 como vitorioso neste conjunto de testes.

O algoritmo DFSANE-O1, além de não ser mais robusto que os demais, se mostrou relativamente ineficiente. Embora seja ligeiramente mais eficiente quando se considera o gráfico que representa o número de iterações como medida de desempenho, o mesmo não ocorre nas demais medidas analisadas, onde apresentou resultados inferiores. Já era esperado um número maior de avaliações de função para este algoritmo, já que é o único que deve certamente fazer ao menos duas avaliações de função por iteração. No entanto, a expectativa era de que o algoritmo necessitasse, em geral, de um menor número médio de iterações para convergir, além de que fosse mais robusto, forçando, assim, uma melhoria de desempenho nos outros critérios avaliados.

Pela análise da Tabela 2.1 e da Figura 2.2, podemos observar pelos gráficos referentes ao número de avaliações de função e referentes ao tempo de execução uma queda em termos de eficiência nos algoritmos DFSANE-D3 e DFSANE-D4, o que já esperávamos, pois, devido à adição das novas direções

esses algoritmos tendem a testar mais pontos por iterações. Novamente, a expectativa era que essa perda de eficiência viesse acompanhada de algoritmos mais robustos e, no entanto, os resultados mostram que, os algoritmos `DFSANE-D4` e `DFSANE-D3` não obtiveram melhoras em termos de robustez.

Entretanto, ao trabalhar com as dimensões e os valores iniciais padrões encontrados em [14] foi possível constatar que:

1. em alguns problemas, é sugerido utilizar dimensões relativamente baixas, quando a intenção é trabalhar com problemas de grande porte. Por exemplo, para o problema 7 (*Badly scaled augmented Powell's function*) são sugeridas as dimensões $n = 9,\ 99,\ 399$;

2. o valor inicial sugerido, em alguns casos, é muito próximo da solução. Para o problema 2 (*Exponential function 2*), por exemplo, o valor inicial sugerido é $x_0 = (\frac{1}{n^2}, \frac{1}{n^2}, \ldots \frac{1}{n^2})^\top$ e, para as três dimensões testadas pelos autores ($n = 500, 1000, 2000$), a norma do valor inicial é menor do que $\dfrac{\varepsilon_a\sqrt{n}}{1-\varepsilon_r}$ (conforme pode ser visto pela Tabela 2.2), indicando que o ponto inicial já está satisfazendo o critério de parada adotado no artigo.

n	$\|F(x_0)\|$	$\dfrac{\varepsilon_a\sqrt{n}}{1-\varepsilon_r}$
500	$4.00003470677632 \times 10^{-6}$	$2.23629160666046 \times 10^{-4}$
1000	$1.00000091732796 \times 10^{-6}$	$3.16259391956034 \times 10^{-4}$
2000	$2.50000083660249 \times 10^{-7}$	$4.47258321332091 \times 10^{-4}$

Tabela 2.2: Norma de valores iniciais e valor do critério de parada para o problema *Exponential function 2*

Isso nos levou a refazer os testes com algumas modificações. Os valores para n foram modificados e todos os problemas foram resolvidos com as dimensões $n = 100,\ 500,\ 1000,\ 2000$ e 5000. Também foram modificados os valores iniciais e todos os problemas foram resolvidos com 20 valores iniciais diferentes, gerados aleatoriamente na vizinhança do vetor inicial originalmente proposto.

Os novos vetores iniciais foram divididos em duas categorias, os 10 primeiros vetores foram gerados obedecendo distribuição uniforme e os 10 últimos foram gerados satisfazendo uma distribuição normal. Considerando $x_0 = (x_1, x_2, \ldots, x_n)^\top$ o valor inicial proposto em [14], para gerar os vetores com distribuição uniforme, definimos que a i-ésima componente do novo vetor inicial deveria ser gerada aleatoriamente no intervalo $[a_i, b_i]$, onde $a_i = x_i + \min\{-5, -5|x_i|\}$ e $b_i = x_i + \max\{5, 5|x_i|\}$. Já para gerar os vetores com distribuição normal, a i-ésima componente do novo vetor foi gerada com média igual a x_i e desvio padrão igual a $\max\{5, 5|x_i|\}$.

Os resultados para esses novos testes podem ser visualizados nos gráficos da Figura 2.3 para fins de comparação do algoritmo original, `DFSANE`, com aqueles com novas estratégias de ordenação e

na Figura 2.4 onde se encontram os gráficos que permitem a análise de desempenho dos algoritmos com novas direções. Além disso, preparamos a Figura 2.5 com o gráfico das curvas de desempenho do algoritmo original, dos algoritmos modificados que tiveram melhores desempenhos: `DFSANE-O2` e `DFSANE-D3` e, também, de um novo algoritmo que combina essas duas estratégias, ao qual chamamos `DFSANE-O2D3`. Por fim, apresentamos também a Tabela 2.3, que indica a porcentagem de convergência dos algoritmos testados nesse novo conjunto de testes.

	EST	**EAVF**	**PPF**	**C**
`DFSANE`	5.9%	39.7%	5.5%	48.9%
`DFSANE-O1`	10.7%	40.5%	9.4%	39.4%
`DFSANE-O2`	5.9%	40.0%	7.0%	47.1%
`DFSANE-D3`	5.9%	33.5%	5.2%	55.4%
`DFSANE-D4`	5.1%	38.7%	5.1%	51.1%
`DFSANE-O2D3`	6.0%	35.5%	5.9%	52.6%

Tabela 2.3: Resumo dos resultados - problemas extraídos de [14] - valor inicial aleatório

Os novos resultados permitiram confirmar o bom desempenho algoritmo `DFSANE-D3` se comparado com o algoritmo original. Embora esse algoritmo ainda não tenha se mostrado tão eficiente quanto `DFSANE`, se analisadas as medidas de desempenho: número de avaliações de função e tempo de execução, podemos notar que seus gráficos rapidamente ultrapassam os gráficos do algoritmo `DFSANE`, sendo que a ultrapassagem ocorre para algum valor de τ menor que 1.5. Além disso, o fato de ele ter resolvido um número de problemas consideravelmente maior é bastante animador. O Algoritmo `DFSANE-D4` também foi mais robusto que `DFSANE`. No entanto, como foi menos robusto e menos eficiente que o algoritmo que utiliza a direção d_3, decidimos que nos próximos testes iremos abandonar esse algoritmo e focalizar no Algoritmo `DFSANE-D3`.

Os algoritmos com ordenação tiveram desempenhos piores que o do algoritmo original, tanto em eficiência quanto em robustez. Como o desempenho de `DFSANE-O2` foi parecido com o do algoritmo original, decidimos, por ora, não abandonar essa estratégia. Por fim, a estratégia `DFSANE-O2D3`, que combina as duas melhores estratégias, teve desempenho parecido com a estratégia `DFSANE-D3`, ainda que um pouco inferior, e também não será descartada por enquanto.

Como podemos observar pela Tabela 2.3, os algoritmos apresentados ainda deixam um espaço para que sejam melhorados em termos de robustez. Apesar de estarmos trabalhando com problemas difíceis, seria interessante aumentar o número de problemas resolvidos. Tentaremos isso no próximo capítulo, quando iremos propor um algoritmo híbrido que combina a estratégia `DFSANE` (e as versões aqui desenvolvidas) com o método de Newton inexato.

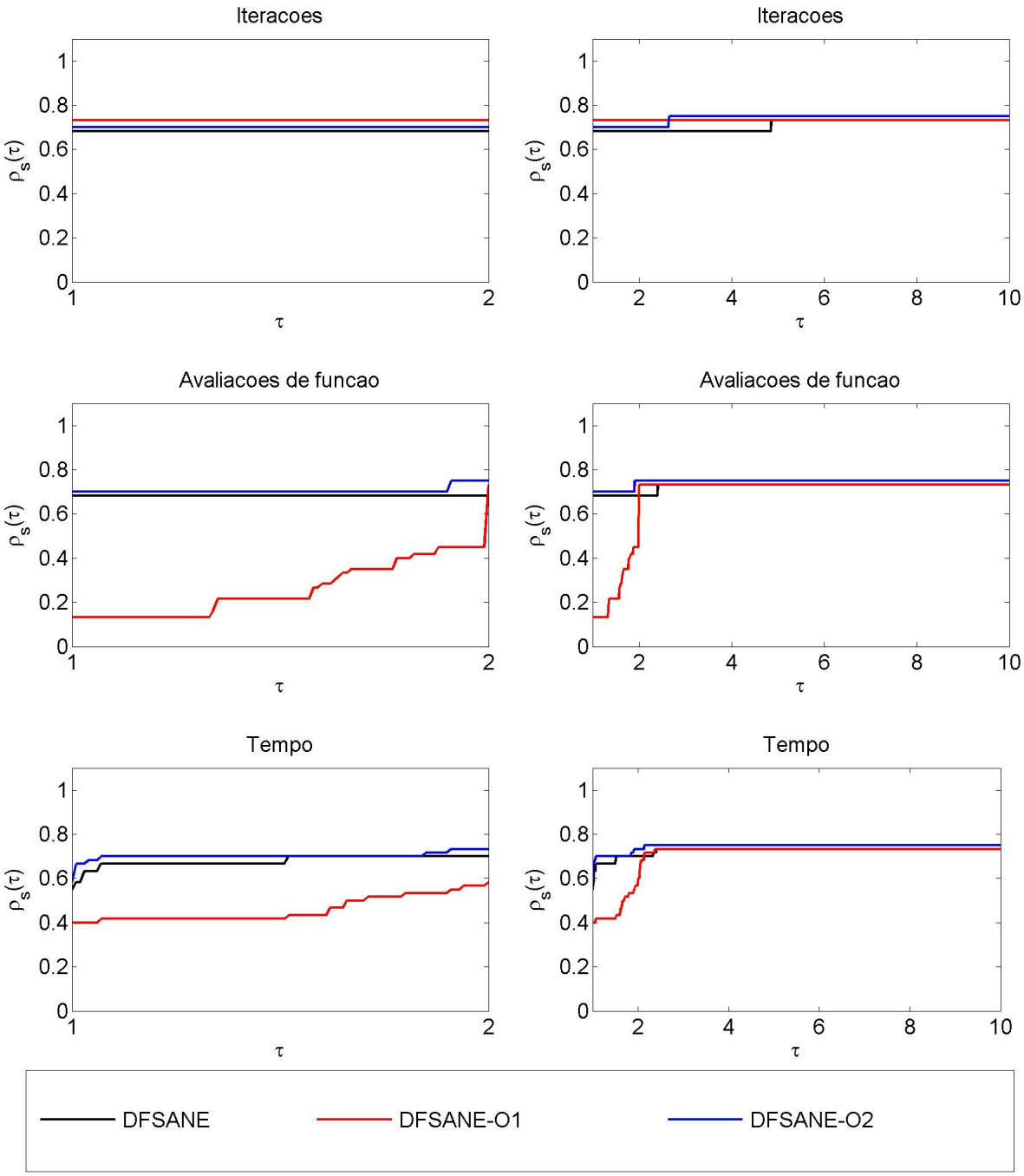

Figura 2.1: Desempenho dos algoritmos `DFSANE`, `DFSANE-O1` e `DFSANE-O2`

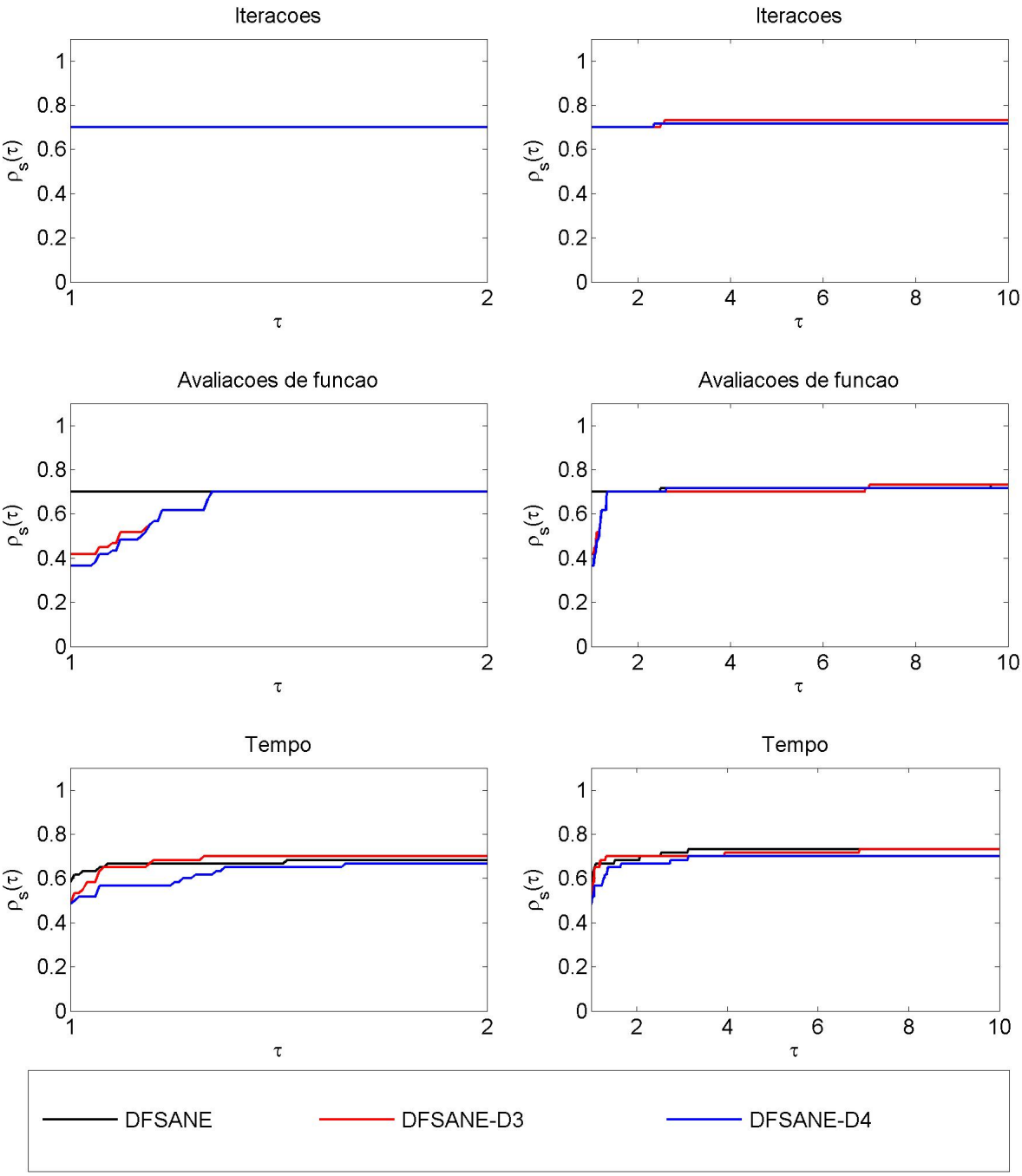

Figura 2.2: Desempenho dos algoritmos `DFSANE`, `DFSANE-D3` e `DFSANE-D4`

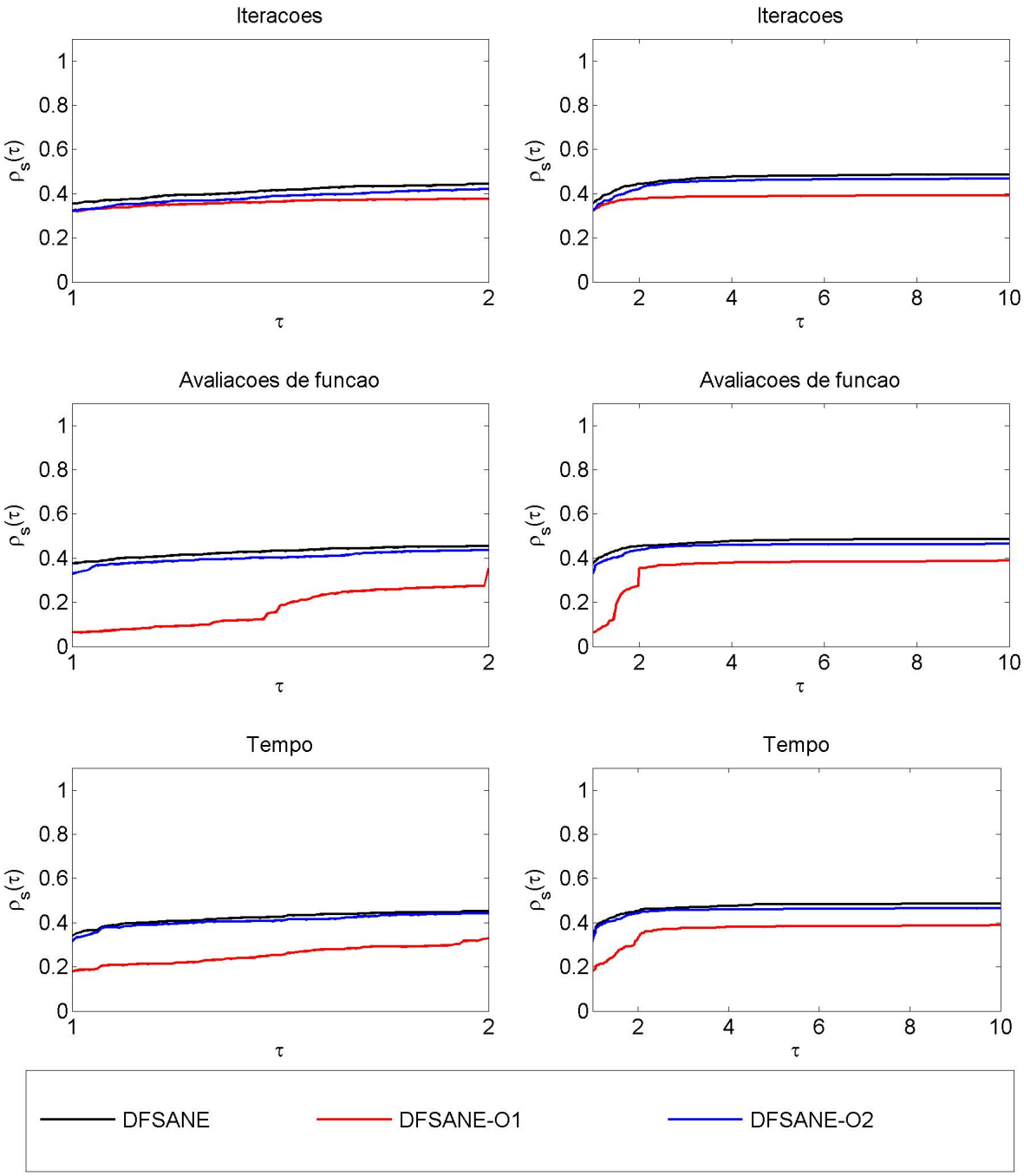

Figura 2.3: Desempenho dos algoritmos `DFSANE`, `DFSANE-O1` e `DFSANE-O2` - vetores iniciais aleatórios

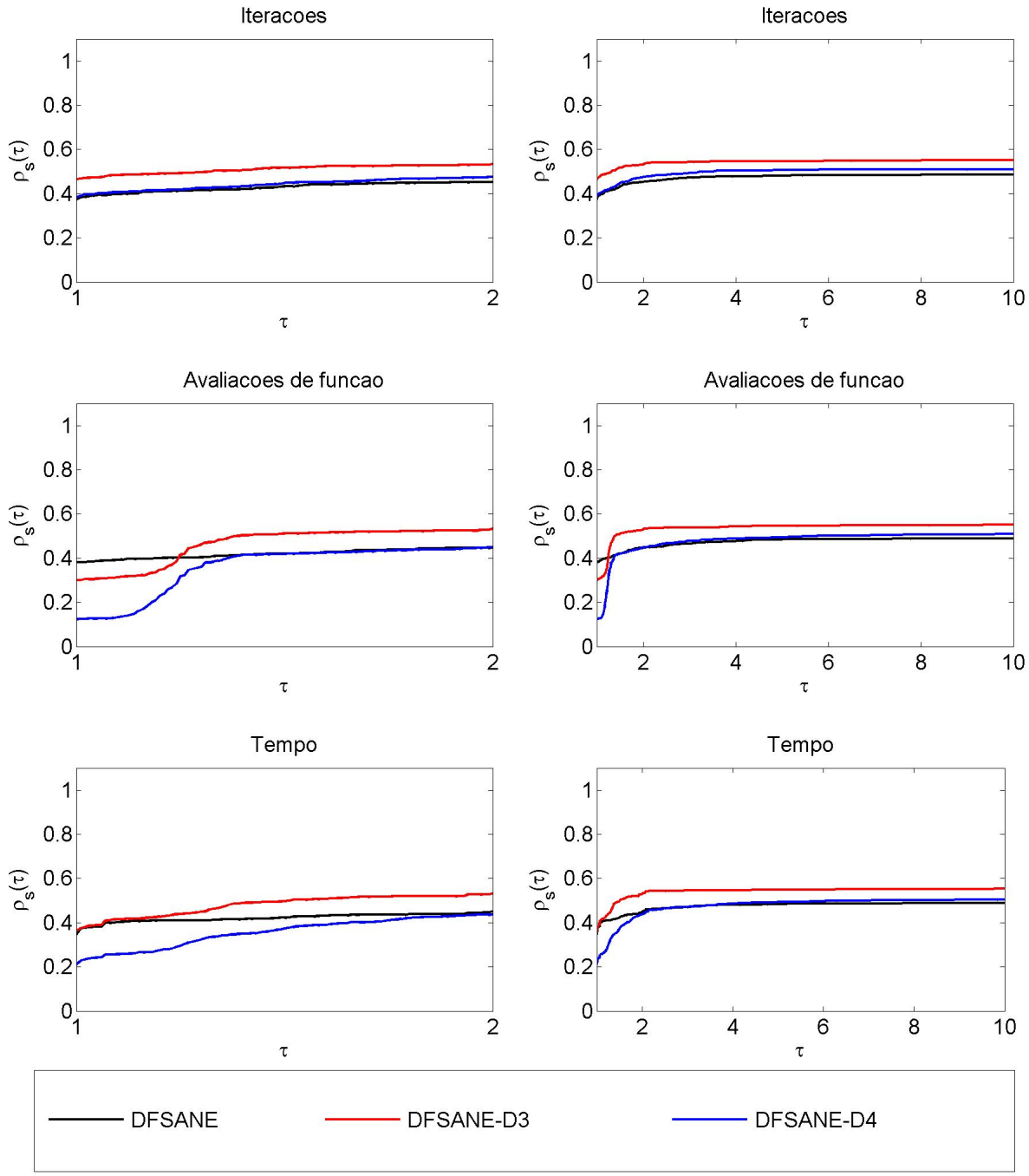

Figura 2.4: Desempenho dos algoritmos `DFSANE`, `DFSANE-D3` e `DFSANE-D4` - vetores iniciais aleatórios

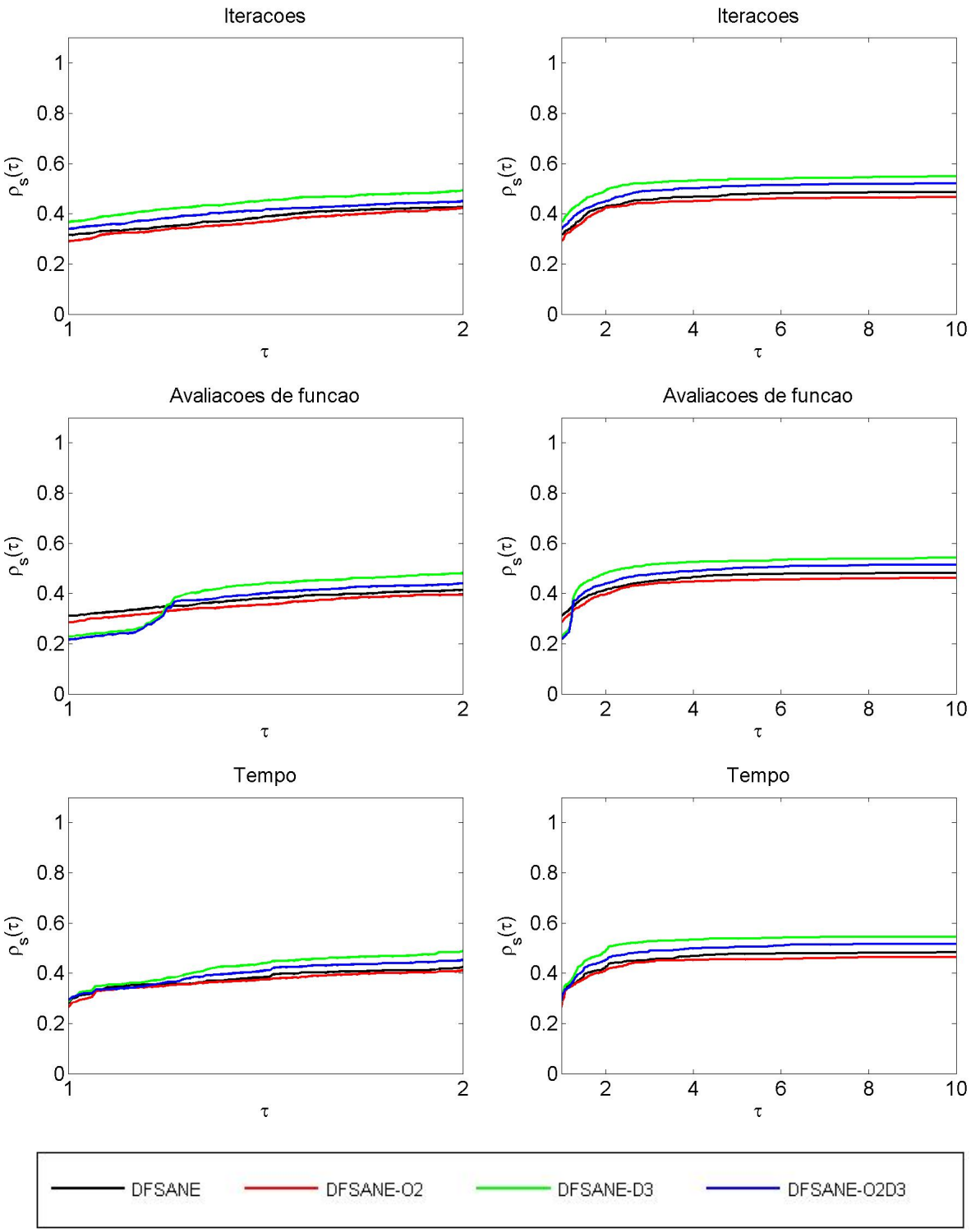

Figura 2.5: Desempenho dos algoritmos `DFSANE`, `DFSANE-O2`, `DFSANE-D3` e `DFSANE-O2D3` - vetores iniciais aleatórios

Capítulo

3 Método híbrido para resolução de sistemas não lineares

Os testes preliminares, apresentados na Seção 2.6, mostraram que o desempenho do método do resíduo espectral deixa a desejar quando se trata da robustez. Neste capítulo, iremos desenvolver uma estratégia híbrida onde métodos do tipo espectral serviriam, numa primeira fase, como acelerador do método de Newton inexato, que corresponderia à segunda fase. O método híbrido desenvolvido neste capítulo é um método de convergência não-monótona e livre do uso de derivadas. A motivação é propor um método mais robusto, porém conservando as características do resíduo espectral.

Nos últimos anos, foram propostos trabalhos para resolução de sistemas não lineares através de métodos híbridos, onde duas estratégias de resolução são combinadas para obter um melhor desempenho. Grippo e Sciandrone publicaram trabalhos sobre o tema, utilizando como primeira fase uma versão do método do resíduo espectral (ver [29]) ou uma versão *derivative-free* do método de Newton inexato (ver [30]), ambos sucedidos por uma segunda fase que utiliza uma estratégia de busca direta em caso de fracassos nas respectivas estratégias iniciais. No entanto, devido ao alto número de avaliações de função que tendem a necessitar por iteração, estratégias de busca direta não são aconselháveis para problemas de grande porte, daí a opção de utilizá-las somente no caso em que não restam outros recursos para a resolução do problema.

Iniciamos o capítulo com uma seção teórica sobre o método Newton-GMRES e, após isso, apresentamos o método proposto e resultados de convergência. O capítulo é finalizado com relatos dos testes numéricos realizados com os novos algoritmos.

3.1. Método Newton-GMRES

Embora carregue o nome de Isaac Newton (1642-1727), o método de Newton para resolução de sistemas não lineares foi proposto, na verdade, por Simpson (1710-1761) (ver [54]), e corresponde a uma generalização do método conhecido como Newton-Raphson para encontrar zeros de uma função real.

Supondo a função F do sistema não-linear (1.1) continuamente diferenciável, cada iteração do método de Newton irá utilizar como direção de busca a solução do sistema linear:

$$J(x_k)d = -F(x_k), \tag{3.1}$$

onde $J(x_k)$ é a matriz Jacobiana de F no ponto x_k.

Uma das principais vantagens do método de Newton é a sua boa taxa de convergência local. Sob hipóteses adequadas, entre elas a de que F é continuamente diferenciável, o método tem garantida uma *taxa de convergência q-superlinear*. Isto quer dizer que, se a sequência $\{x_k\}$ gerada pelo método converge para x^*, então:

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0. \tag{3.2}$$

Além disso, sob a hipótese de que F é Lipschitz continuamente diferenciável, a taxa de convergência torna-se *q-quadrática*. Neste caso, para um valor de k suficientemente grande, a sequência gerada satisfaz:

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \leq C, \tag{3.3}$$

onde C é uma constante positiva. O Teorema 3.1 formaliza esses resultados.

Teorema 3.1 *Considere F uma função continuamente diferenciável em um conjunto convexo e aberto $\mathcal{A} \subseteq \mathbb{R}^n$, um ponto $x^* \in \mathcal{A}$ de forma que $F(x^*) = 0$ e $J(x^*)$ é não-singular. Seja a sequência $\{x_k\}$ gerada por:*

$$x_{k+1} = x_k + d_k,$$

onde d_k é a solução do sistema linear (3.1). Então, existe $\varepsilon_1 > 0$ tal que, se a aproximação inicial x_0 satisfaz $\|x_0 - x^*\| < \varepsilon_1$, a sequência está bem definida e converge para x^* q-superlinearmente.

Caso F seja Lipschitz continuamente diferenciável numa vizinhança de x^*, existe $\varepsilon_2 > 0$ tal que, se $\|x_0 - x^*\| < \varepsilon_2$, a sequência converge para x^* q-quadraticamente.

Prova: Ver [19, 42]. ∎

A implementação do método de Newton, no entanto, pressupõe que, a cada iteração, a matriz Jacobiana seja calculada e a solução do sistema linear (3.1) seja obtida. Por esta razão, o método torna-se computacionalmente caro para problemas de grande porte e chega a ser impraticável em alguns casos.

Dembo, Eisenstat e Steihaug [18] propuseram, em 1982, o método de Newton inexato, no qual, a cada iteração k, uma direção de busca d_k é escolhida de modo a satisfazer o seguinte critério:

$$\|J(x_k)d_k + F(x_k)\| \leq \eta_k \|F(x_k)\|, \tag{3.4}$$

onde o parâmetro $\eta_k \in [0, 1)$ é denominado termo forçante.

Com essa opção, o trabalho computacional do método de Newton é reduzido, já que o sistema linear (3.1) é resolvido aproximadamente. A contrapartida, neste caso é, uma taxa de convergência inferior. A taxa de convergência do método de Newton inexato depende da sequência de termos forçantes adotada e de hipóteses de Lipschitz continuidade sobre a Jacobiana numa vizinhança da solução.

Teorema 3.2 *Considere F uma função continuamente diferenciável em um conjunto convexo e aberto $\mathcal{A} \subseteq \mathbb{R}^n$, $x^* \in \mathcal{A}$ tal que $F(x^*) = 0$ e $J(x^*)$ é não-singular e a sequência $\{x_k\}$ gerada por:*

$$x_{k+1} = x_k + d_k,$$

onde d_k satisfaz a desigualdade (3.4). Então, existe $\varepsilon_1 > 0$ tal que, tomando x_0 de forma que $\|x_0 - x^*\| < \varepsilon_1$, a sequência está bem definida e:

1. *Se $0 \leq \eta_k \leq \eta < 1$, e η é suficientemente pequeno então $\{x_k\}$ converge para x^* q-linearmente com taxa C, ou seja, existe $C \in (0, 1)$ tal que:*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq C; \tag{3.5}$$

2. *Se $\lim_{k\to\infty} \eta_k = 0$ então a taxa de convergência é q-superlinear;*
3. *Se adicionarmos ainda as hipóteses de que a Jacobiana de F é Lipschitz contínua numa vizinhança de x^* e de que $\eta_k = \mathcal{O}(\|F(x_k)\|)$, a convergência terá taxa q-quadrática.*

Prova: Ver [42], Teorema 11.3. ∎

Como pode ser observado pelo Teorema 3.2, a taxa de convergência do método de Newton inexato depende do valor do termo forçante. Vários trabalhos na área, ver [21, 25], propõem diferentes sequências de parâmetros $\{\eta_k\}$ visando melhorias na taxa de convergência do método.

Além disso, a determinação de um método que resolva aproximadamente o sistema linear (3.1), a fim de encontrar uma direção que satisfaça a condição (3.4), é tema de estudo que originou distintos algoritmos do tipo Newton inexato. Entre as propostas mais utilizadas para a resolução do sistema linear encontra-se a classe de métodos de projeções sobre subespaços de Krylov [8, 34]:

Definição 3.1 *Dada uma matriz A : $n \times n$, não singular, e um vetor $v \in \mathbb{R}^n$, a sequência: $v, Av, A^2v, A^3v, ...$ é chamada sequência de Krylov. O subespaço gerado pelos vetores $v, Av, A^2v, A^3v, ...A^{\ell-1}v$ é denominado subespaço de Krylov de ordem ℓ, e denotado por $\mathcal{K}_\ell(A, v)$.*

Neste trabalho optamos por empregar o método GMRES (*Generalized Minimal Residual*) [47], proposto por Saad e Schultz em 1986 e conhecido pelo bom desempenho na resolução de problemas de grande porte e, também, pela possibilidade de lidar com sistemas lineares não necessariamente simétricos.

A utilização do método GMRES ainda traz a vantagem adicional de requerer o uso das matrizes Jacobianas somente em produtos matriz-vetor, que podem ser aproximados através de expansões em série de Taylor:

$$J(x_k)w \approx \frac{F(x_k + \nu w) - F(x_k)}{\nu}, \quad \nu \in \mathbb{R},\ w \in \mathbb{R}^n. \tag{3.6}$$

Esse processo é conhecido como *matrix-free* e permite que uma solução para a desigualdade (3.4) seja encontrada sem a necessidade de cálculo da matriz Jacobiana.

Considerando a sua aplicação na resolução do sistema linear (3.1) e utilizando um valor inicial d_0, a ideia básica do método GMRES é, a cada iteração ℓ, minimizar a norma-2 do resíduo $r = F(x_k) + J(x_k)d$ sobre o subespaço de Krylov determinado pela matriz $J(x_k)$ e o resíduo inicial $(r_0 = F(x_k) + J(x_k)d_0)$, $\mathcal{K}_\ell(J(x_k), r_0)$:

$$\begin{aligned} &\min \|F(x_k) + J(x_k)d\|^2 \\ &\text{sujeito a } d \in d_0 + \mathcal{K}_\ell(J(x_k), r_0). \end{aligned} \tag{3.7}$$

Saad e Schultz [47] demonstram um resultado que nos permite afirmar que, garantida a não-singularidade de $J(x_k)$, o método GMRES, teoricamente, encontra a solução de um sistema linear $n \times n$, no máximo, após n iterações. No entanto, ao trabalharmos com um método de Newton inexato, nem sempre precisaremos realizar todas as n etapas para obter a solução do sistema (3.1), já que não estamos interessados em resolver o sistema linear (3.1) e sim buscando uma solução aproximada que satisfaça a condição (3.4). Neste sentido, o método GMRES se torna um método iterativo e irá parar quando encontrar uma direção d_k que satisfaça essa condição.

É convencionado chamar as iterações do algoritmo interno, necessárias para obtenção do passo d_k, de *iterações internas*. As direções do método de Newton inexato, isto é, as iterações do tipo $x_{k+1} = x_k + d_k$ dotadas de estratégia de globalização, são chamadas de *iterações externas*.

É conveniente transformar a base $\{r_0, J(x_k)r_0, J(x_k)^2r_0, \ldots\}$ do subespaço de Krylov $\mathcal{K}_\ell(J(x_k), r_0)$ em uma base ortogonal e, para isso, é utilizado o processo de Arnoldi [34]:

Algoritmo 3.1 *Processo de Arnoldi para Ortonormalização (adaptado para Newton-GMRES)*

1. $v_1 = r_0/(\|r_0\|)$;
2. *Para* $j = 1, 2, \ldots, \ell$
   (a) $\bar{v} = J(x_k)v_j$;
   (b) *para* $i = 1, 2, \ldots, j$
   $\hookrightarrow\ h_{i,j} = \bar{v}^\top v_i$;
   (c) $\bar{v} = \bar{v} - \sum_{i=1}^{j} h_{i,j}v_i$;
   (d) $h_{j+1,j} = \|\bar{v}\|$;
   (e) $v_{j+1} = \bar{v}/h_{j+1,j}$.

Supondo que o Algoritmo 3.1 não apresentou falha de execução, após a ℓ-ésima iteração, teremos obtido os vetores ortonormais $v_1, v_2, \ldots, v_{\ell+1}$. Definimos então as matrizes $V_\ell : n \times \ell$ cujas colunas são os vetores $v_1, v_2, \ldots, v_\ell$ e $V_{\ell+1} : n \times (\ell + 1)$ que é a matriz V_ℓ acrescida do vetor $v_{\ell+1}$. Também podemos definir a matriz $\bar{H}_\ell : (\ell + 1) \times \ell$, Hessenberg superior, cujo elemento da i-ésima linha e j-ésima coluna é dado por:

$$\bar{H}_\ell(i,j) = \begin{cases} 0, & \text{se } i > j+1 \\ h_{i,j}, & \text{caso contrário.} \end{cases}$$

Dessa forma, é possível estabelecer a seguinte relação:

$$J(x_k)V_\ell = V_{\ell+1}\bar{H}_\ell. \tag{3.8}$$

Observação 3.1: A possibilidade do Algoritmo 3.1 apresentar falhas de execução, durante o processo de Arnoldi, existe e irá ocorrer quando, na ℓ-ésima iteração, $h_{\ell+1,\ell} = 0$ ou $\hat{v} = J(x_k)v_\ell = 0$. Conforme Kelley [34], isso irá ocorrer se, e somente se, d_ℓ for a solução exata do sistema (3.1); é o que chamam na literatura de *lucky breakdown*. Neste caso, o algoritmo segue utilizando a direção de Newton encontrada.

Considerando que as colunas de V_ℓ formam uma base ortogonal para o subespaço $\mathcal{K}_\ell(J(x_k), r_0)$, para todo $d \in \mathcal{K}_\ell(J(x_k), r_0)$, existe $y \in \mathbb{R}^\ell$ tal que $d = d_0 + V_\ell y$. Utilizando a relação (3.8) e como, pelo Algoritmo 3.1, $v_1 = r_0/\|r_0\|$, podemos escrever:

$$\|F(x_k) + J(x_k)d\| = \|F(x_k) + J(x_k)(d_0 + V_\ell y)\| = \|r_0 + J(x_k)V_\ell y\| = \|r_0 + V_{\ell+1}\bar{H}_\ell y\|, \tag{3.9}$$

e, assim, definindo $\beta = \|r_0\|$, podemos reescrever o problema de quadrados mínimos (3.7):

$$\begin{aligned} &\min \|\beta e_1 + \bar{H}_\ell y\|^2 \\ &\text{sujeito a } y \in \mathbb{R}^\ell. \end{aligned} \tag{3.10}$$

O método de Newton inexato resultante do acoplamento de uma versão do método GMRES sem derivadas, assumindo $d_0 = 0$ a cada iteração externa, está detalhado pelo Algoritmo 3.2. Neste algoritmo, o método GMRES utiliza o produto da matriz Jacobiana por um vetor aproximado através de diferenças avançadas e, portanto, recebe o nome de FDGMRES (*Forward Difference GMRES*).

Algoritmo 3.2 *FDGMRES*

Parâmetros de entrada: F, x_k, σ e η_k.

1. *Faça* $r = -F(x_k)$, $\rho = \|F(x_k)\|$, $\beta = \rho$, $\ell = 0$ *e* $v_1 = r/\rho$.
2. *Enquanto* $\rho > \eta_k \|F(x_k)\|$ *e* $\ell < n$
   (a) *Faça* $\ell = \ell + 1$.
   (b) *Calcule* $\bar{v} = \dfrac{F(x_k + \sigma v_\ell) - F(x_k)}{\sigma}$.
   (c) *Defina* $\bar{h}_{i,\ell} = v_i^\top \bar{v}$, *para* $i = 1, 2, \ldots, \ell$.
   (d) *Faça* $\bar{v} = \bar{v} - \sum_{i=1}^{\ell} \bar{h}_{i,\ell} v_i$.
   (e) *Se* $\|\bar{v}\| = 0$, *pare o algoritmo. Caso contrário, defina* $\bar{h}_{\ell+1,\ell} = \|\bar{v}\|$ *e, daí,* $v_{\ell+1} = \bar{v}/\bar{h}_{\ell+1,\ell}$.
   (f) *Defina a matriz* $\bar{H}_\ell = \{\bar{h}_{i,j}\}$ *com* $1 \leq i \leq \ell + 1$, *e* $1 \leq j \leq \ell$.
   (g) *Calcule* y_ℓ, *o minimizador do Problema (3.10) e defina* $\rho = \|\beta e_1 + \bar{H}_\ell y_\ell\|$.
3. *Faça* $V = [v_1, \ldots, v_\ell]$ *e* $d_k = V y_\ell$.

A maneira como será obtido o vetor y_ℓ no passo 2g do Algoritmo 3.2 pode variar entre os trabalhos que optam por usar o método Newton-GMRES. Embora alguns métodos usem a versão clássica, proposta por Saad e Schultz [47], outros trabalhos preferem utilizar uma modificação de Walker e Zhou [52] denominada *Simpler GMRES*.

Conforme pode ser visto em [47], Saad e Schultz aproveitam a característica Hessenberg superior da matriz $\bar{H}_\ell$ e obtem sua fatoração QR. Isso é especialmente vantajoso porque através de uma rotação de Givens é possível obter a fatoração QR de $\bar{H}_\ell$ a partir da fatoração QR de $\bar{H}_{\ell-1}$ com um trabalho computacional de ordem ℓ apenas.

Definindo a fatoração encontrada na ℓ-ésima iteração como $\bar{H}_\ell = Q_\ell^\top R_\ell$ e, dado que a matriz Q_ℓ é ortogonal, podemos reescrever o problema (3.10):

$$\begin{aligned} &\min \|\beta Q_\ell e_1 + R_\ell y\|^2 \\ &\text{sujeito a } y \in \mathbb{R}^\ell. \end{aligned} \tag{3.11}$$

O Teorema 3.3 apresenta resultados que indicam outras vantagens práticas proporcionadas pela resolução do sistema (3.1) através da resolução do problema de quadrados mínimos modificado (3.11).

Teorema 3.3 *Considere $R_\ell = Q_\ell \bar{H}_\ell$ como definida anteriormente e $g = \beta Q_\ell e_1$. Denotemos por $\hat{R}_\ell$ a matriz triangular superior $\ell \times \ell$, obtida eliminando-se a última linha de R_ℓ, e $\hat{g}_\ell$ o vetor de dimensão ℓ encontrado eliminando-se o elemento $|g_{\ell+1}|$ (a última linha) de g. Podemos dizer que:*

1. *o posto de $J(x_k)V_\ell$ é igual ao posto de $\hat{R}_\ell$. Em particular, se $r_{\ell,\ell} = 0$ então $J(x_k)$ é singular;*
2. *o vetor y^* que é o minimizador de $\|\beta e_1 + \bar{H}_\ell y\|$ é dado por $y^* = \hat{R}_\ell^{-1}\hat{g}$;*
3. *a norma-2 do resíduo no passo ℓ é dada por $|g_{\ell+1}|$.*

Prova: Ver [46], Proposição 6.9. ∎

Conforme o item 3 do Teorema 3.3, comprovada a não-singularidade da matriz $J(x_k)$, a solução aproximada para o sistema linear (3.1) não necessita ser obtida a cada iteração do GMRES e nem o resíduo precisa ser calculado explicitamente. Somente quando tivermos o valor do módulo do último elemento do vetor $g = \beta Q_\ell e_1$ suficientemente pequeno de modo que satisfaça ao critério de parada é que calculamos o vetor d_k, utilizando, para isto, o último valor y_ℓ encontrado.

3.1.1 GMRES com recomeços

Durante a utilização do método GMRES para resolução de problemas de grande porte, à medida que o número de iterações vai aumentando, crescem também os requerimentos de memória e o custo computacional para realizar cada iteração. Assim, para a resolução de alguns problemas de grande porte, o método torna-se ineficiente.

Uma alternativa para evitar esse tipo de problema é o uso de uma estratégia que reinicializa o método GMRES após um ciclo de m iterações, onde m é um número natural pré-definido. Após o fim desse ciclo, se a última aproximação (d_m) encontrada não for satisfatória, o vetor d_m é utilizado como valor inicial para um novo ciclo de m iterações. Quando utiliza a estratégia com recomeços, o método GMRES é, em geral, designado como GMRES(m).

O mesmo procedimento pode ser adotado quando se trabalha com o método FDGMRES. No entanto, ao utilizar o método FDGMRES dotado com a estratégia de recomeços, algumas mudanças precisam ser feitas no Algoritmo 3.2, o Algoritmo 3.3 explicita essas mudanças. Além da variável m, que indica a número máximo de iterações em cada ciclo do FDGMRES, deve ser fornecida também uma variável $nc_{\max} \in \mathbb{N}$ que indica o número máximo de ciclos que o FDGMRES(m) deve realizar antes de declarar falha de execução.

Algoritmo 3.3 *FDGMRES(m)*

1. *Faça* $r_k = -F(x_k)$, $\rho_k = \|F(x_k)\|, v_1 = (1/\rho_k)r_k$, $\beta_k = \rho_k$, $\ell = 0$, $nc = 0$ *e* $\bar{d} = 0$.
2. *Enquanto* $\rho_k > \eta_k\|F(x_k)\|$ *e* $nc < nc_{\max}$
   (a) *Faça* $\ell = \ell + 1$.
   (b) *Calcule* $\bar{v} = \dfrac{F(x_k + \sigma v_\ell) - F(x_k)}{\sigma}$.
   (c) *Defina* $\bar{h}_{i,\ell} = v_i^\top \bar{v}$, *para* $i = 1, 2, \ldots, \ell$.
   (d) *Faça* $\bar{v} = \bar{v} - \sum_{i=1}^{\ell} \bar{h}_{i,\ell} v_i$.
   (e) *Se* $\|\bar{v}\| = 0$, *pare o algoritmo. Caso contrário, defina* $\bar{h}_{\ell+1,\ell} = \|\bar{v}\|$ *e, daí,* $v_{\ell+1} = \bar{v}/\bar{h}_{\ell+1,\ell}$.
   (f) *Defina a matriz* $\bar{H}_\ell = \{\bar{h}_{i,j}\}$ *com* $1 \le i \le \ell + 1$, *e* $1 \le j \le \ell$.
   (g) *Obtenha* y_ℓ, *o minimizador do Problema (3.10) e defina* $\rho_k = \|\beta_k e_1 + \bar{H}_\ell y_\ell\|$.
   (h) *Se* $\ell = m$ *e* $\rho_k > \eta_k\|F(x_k)\|$
      i. $V = [v_1, \ldots, v_\ell]$ *e* $\bar{d} = \bar{d} + V y_\ell$.
      ii. $r_k = F(x_k) - \dfrac{F(x_k + \sigma\bar{d}) - F(x_k)}{\sigma}$ *e* $\beta_k = \|r_k\|$.
      iii. $v_1 = \dfrac{r_k}{\beta_k}$, $\ell = 0$ *e* $nc = nc + 1$.
3. *Faça* $V = [v_1, \ldots, v_\ell]$ *e* $d_k = \bar{d} + V y_\ell$.

Devemos salientar que, embora a norma-2 do resíduo seja não crescente, conforme pode ser visto em [46], este valor pode sofrer estagnação em casos em que a matriz do sistema linear a ser resolvido não seja positiva-definida e, sendo assim, não há garantia teórica de que o Algoritmo 3.3 convergirá para a solução do sistema linear em questão. No entanto, alguns trabalhos, por exemplo [26, 30], mostraram que a adoção do método com recomeços obtém bons resultados na prática.

3.2. Método proposto

Inicialmente, vamos apresentar o algoritmo conceitual do método híbrido proposto neste trabalho. Este algoritmo generaliza as duas fases que compõem o método: a fase I, que utiliza alguma variação do método DFSANE e a fase II, que consiste no método de Newton inexato. O modelo de algoritmo descrito é flexível na forma como será resolvida cada etapa. Assim, na etapa I, pode ser utilizado o algoritmo DFSANE original ou alguma modificação proposta neste trabalho. E, na etapa II, pode-se estabelecer de que forma o método GMRES será executado, se na forma descrita na seção anterior

ou com base em algum algoritmo que utilize a modificação *Simpler GMRES* [52] como, por exemplo, NITSOL [43]. Um diagrama ilustrando de forma simplificada o funcionamento desse algoritmo pode ser encontrado na Figura 3.1.

Além disso, são deixados em aberto dois critérios de aceitação de ponto, $CA1$ e $CA2$. O critério $CA1$ é utilizado para pontos advindos da direção de busca encontrada pelo método DFSANE e o critério $CA2$ para pontos provenientes da direção encontrada pelo método de Newton inexato. Evidentemente, uma mesma estratégia pode ser utilizada nas duas etapas.

Os algoritmos que seguirem o modelo conceitual aqui proposto mudam da fase I para a fase II após um número $NBL_{\max}$ de reduções em cada uma das duas direções comumente testadas pelo método DFSANE, e em mais uma terceira direção, se o método adotado assim o designar. Se após esse número de reduções, não for obtido nenhum ponto que satisfaça o critério de aceitação escolhido para esta etapa, o método de Newton inexato é acionado. O algoritmo termina se for encontrado um ponto x_k tal que $F(x_k) = 0$.

Algoritmo 3.4 ***HIB1***

Parâmetros de entrada: *x_0, $NBL_{\max}$, duas condições de aceitação de pontos $CA1$ e $CA2$, θ_1, θ_2, $\theta_3 \in (0,1)$, $0 < \tau_{\min} < \tau_{\max} < 1$ e uma sequência de pontos η_k tal que $\eta_k \in [0,1)$ para todo k.*

1. *Defina $k = 0$*
2. *Se $NBL = NBL_{max}$ vá para o passo 6.*
3. *Encontre a direção de busca d_k através do método* `DFSANE` *(ou uma de suas variações). Defina $\lambda_+ = \lambda_- = 1$ e $NBL = 0$.*
4. *Se $x_+ = x_k + \lambda_+ d_k$ ou $x_- = x_k - \lambda_- d_k$ satisfazem $CA1$ defina, respectivamente, $\lambda = \lambda_+$ ou $\lambda = \lambda_-$ e vá para o passo 11.*
5. *Se $NBL = NBL_{max}$ vá para o passo 6. Caso contrário:*
   (a) *Escolha $\lambda_+ \in [\tau_{\min}\lambda_+, \tau_{\max}\lambda_+]$ e $\lambda_- \in [\tau_{\min}\lambda_-, \tau_{\max}\lambda_-]$;*
   (b) *faça $NBL = NBL + 1$ e volte para o passo 4.*
6. *Faça $\eta = \eta_k$ e determine d_k pelo método* `FDGMRES`*. Defina $\lambda = 1$.*
7. *Se $x_k + \lambda d_k$ satisfaz $CA2$ vá para o passo 11.*
8. *$t = 0$*
9. *Compute um novo valor para λ utilizando um algoritmo de busca linear.*

10. *Se $\lambda = 0$ então coloque $\sigma = \theta_1\sigma$, $\eta = \theta_2\eta$. Encontre uma nova direção d_k usando o método* `FDGMRES`. *Defina $\lambda = 1$, $\mu = \theta_3\mu$, $t = t + 1$ e vá para o passo 9.*

11. *Defina $\lambda_k = \lambda$, $\tilde{\sigma}_k = \sigma$, $\tilde{\eta}_k = \eta$, $\tilde{\mu}_k = \mu$, $x_{k+1} = x_k + \lambda_k d_k$ e faça $k = k + 1$.*

12. *Se $F(x_k) = 0$, pare o algoritmo e declare SUCESSO. Caso contrário, vá para o passo 2*

Observação 3.2: O algoritmo de busca linear utilizado no Passo 9 será explicitado na próxima subseção (Algoritmo 3.5).

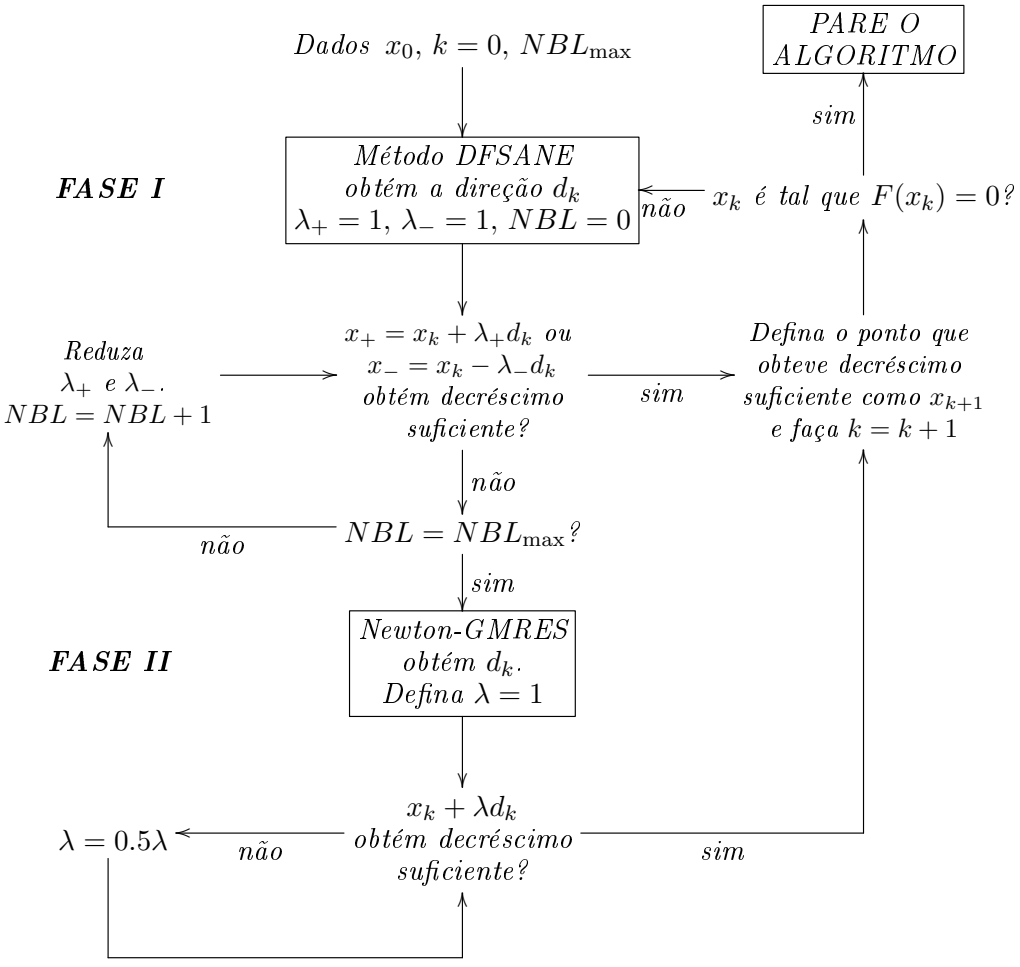

Figura 3.1: Funcionamento do Algoritmo 3.4 (`HIB1`)

3.2.1 Análise de convergência

Nesta subseção, iremos apresentar resultados de convergência para o Algoritmo HIB1 em que os critérios de aceitação CA1 e CA2 são definidos conforme a proposta de La Cruz, Martínez e Raydan [13], representada na desigualdade (2.21).

Durante toda esta seção, vamos considerar como função de mérito $f(x) = \|F(x)\|^2$. Além disso, definimos

$$W_k = \max_{0 \leq j \leq \min\{k, M-1\}} f(x_{k-j}) \tag{3.12}$$

e, também, uma sequência de índices $\{\nu(k)\}$ tal que

$$f(x_{\nu(k)}) = W_k. \tag{3.13}$$

A demonstração do resultado de convergência segue, essencialmente, as ideias apresentadas por Grippo e Sciandrone em [30], sendo assim, vamos, primeiramente, analisar a relação entre os critérios de aceitação para novos pontos encontrados neste trabalho com o critério (2.21), encontrado em [13].

A primeira condição de [30] é utilizada durante a execução da primeira fase do Algoritmo NM1, que consiste no método Newton-GMRES com diferenças finitas para evitar o cálculo explícito dos produtos entre matrizes Jacobianas J_k e vetores. A este algoritmo foi acoplada a estratégia *watchdog* [11] para aceleração. A condição exige que o passo $\lambda_k \in (0, 1]$ cumpra:

$$f(x_k + \lambda_k d_k) \leq (1 - \gamma\lambda_k) W_k, \tag{3.14}$$

onde $\gamma \in (0, 1)$.

Observação 3.3: A rigor, diferentemente do que é feito neste trabalho, em [30] a função de mérito adotada é $f(x) = \big\|F(x)\big\|$. Essa diferença fará com que sejam necessários alguns ajustes quando tivermos de utilizar os resultados encontrados em [30] e os faremos no momento conveniente. Por ora, a análise será feita sem a distinção da função de mérito a ser utilizada.

Definindo a sequência $\{\zeta_k\}$ com $\zeta_k > 0$ para todo k e $\sum_{i=1}^{\infty} \zeta_i = \zeta < \infty$, vamos ter que:

$$(1 - \gamma\lambda_k) W_k < (1 - \gamma\lambda_k) W_k + \zeta_k \leq W_k + \zeta_k - \gamma\lambda_k f(x_k) \leq W_k + \zeta_k - \gamma\lambda_k^2 f(x_k). \tag{3.15}$$

A última expressão da direita da relação (3.15) corresponde à expressão da direita da condição (2.21). Assim sendo, satisfazer a condição (3.14) implica em satisfazer a condição (2.21) proposta por La Cruz, Martínez e Raydan [13]. No entanto, a recíproca não é verdadeira, o que indica que

a condição (2.21) é mais flexível e poderá aceitar pontos que não seriam aceitos se utilizados os critérios propostos por Grippo e Sciandrone [30].

A segunda condição proposta em [30] é utilizada somente na etapa de busca direta do Algoritmo NM2 e dada por:

$$f(x_k + \lambda_k d_k) \leq (1 - \gamma\lambda_k^2)W_k. \tag{3.16}$$

Desenvolvendo de uma maneira semelhante ao que foi feito em (3.15), concluímos que a satisfação da condição (3.16) também implica na satisfação da condição (2.21), proposta em [13].

Os resultados de convergência apresentados em [30] são baseados no fato de que todo ponto encontrado pela sequência gerada pelo Algoritmo NM1 ou pelo Algoritmo NM2 satisfaz:

$$f(x_{k+1}) \leq W_k - \sigma(W_k), \tag{3.17}$$

em que $\sigma : \mathbb{R}_+ \to \mathbb{R}_+$ é chamada função forçante e deve garantir que:

$$\lim_{k\to\infty} \sigma(W_k) = 0 \Rightarrow \lim_{k\to\infty} W_k = 0. \tag{3.18}$$

Em nosso caso, no lugar de (3.17), vamos utilizar uma condição do tipo

$$f(x_{k+1}) \leq W_k + \zeta_k - cf(x_k), \tag{3.19}$$

onde $c > 0$, cujo lado direito não permite que seja definida uma função forçante sobre $f(x_k)$ (e tampouco sobre W_k), visto que $-(\zeta_k - cf(x_k))$ pode assumir valores negativos (a menos que $\zeta_k < cf(x_k)$ para todo k). Ainda assim, é possível estabelecer um resultado análogo àquele encontrado em [30] e que será utilizado na demonstração do teorema de convergência para o método híbrido aqui desenvolvido.

Lema 3.1 *Considere* $f : \mathbb{R}^n \to \mathbb{R}_+$, $M \in \mathbb{N}$, $W_k = \max_{0\leq j\leq \min\{k,M-1\}} f(x_{k-j})$, $\bar{c} \in \mathbb{R}$, $\bar{c} > 0$ *e* $\{\zeta_k\}$ *uma sequência em* $\mathbb{R}$ *tal que* $\zeta_k > 0$ *para todo* k *e* $\sum_{i=1}^{\infty} \zeta_i = \zeta < \infty$. *Tem-se então que*

$$\lim_{k\to\infty} \bar{c}f(x_k) - \zeta_k = 0 \Rightarrow \lim_{k\to\infty} W_k = 0. \tag{3.20}$$

Prova: Se

$$\lim_{k\to\infty} [\bar{c}f(x_k) - \zeta_k] = 0, \tag{3.21}$$

e dado que, por hipótese, $\lim_{k\to\infty} -\zeta_k = 0$, tem-se $\lim_{k\to\infty} \bar{c}f(x_k) = 0$ e, consequentemente,

$$\lim_{k\to\infty} f(x_k) = 0. \tag{3.22}$$

Vamos ver agora que,

$$\lim_{k\to\infty} f(x_k) = 0 \Rightarrow \lim_{k\to\infty} W_k = 0. \tag{3.23}$$

Por (3.22), dado $\varepsilon > 0$, existe $\bar{k} \in \mathbb{N}$ tal que, para todo $k > \bar{k}$, $f(x_k) < \varepsilon$. Por (3.13), para todo $k > \bar{k} + M$ tem-se que $W_k = f(x_{\nu(k)})$ com $\nu(k) > \bar{k}$ e, portanto, $W_k < \varepsilon$, o que comprova o que queríamos. ■

A Proposição 3.1 é uma adaptação de resultados encontrados nos trabalhos [28, 29, 30] e nos dará as condições necessárias para demonstrar os resultados da Proposição 3.2 que, por sua vez, será empregada para a demonstração do teorema de convergência do algoritmo híbrido desenvolvido.

Proposição 3.1 *Considere $\{x_k\} \subset \mathbb{R}^n$ uma sequência de pontos tal que:*

$$f(x_{k+1}) \leq W_k + \zeta_k, \text{ para todo } k, \tag{3.24}$$

com $\zeta_k > 0$ para todo k e $\sum_{i=1}^{\infty} \zeta_i = \zeta < \infty$. Então

1. *$x_k \in \bar{\mathcal{L}}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0) + \zeta\}$ para todo k.*
2. *Se, além disso, é possível estabelecer um índice $\bar{k} \in \mathbb{N}$ que garante*

$$f(x_{k+1}) \leq W_k \text{ para todo } k \geq \bar{k}, \tag{3.25}$$

então a sequência $\{W_k\}_{k>\bar{k}}$ é monotonamente não-crescente.

Prova: Considerando que $\min\{k+1, M-1\} \leq \min\{k, M-1\}+1$, pela definição de $\nu(k)$ (conforme a equação (3.13)), temos:

$$\begin{aligned} W_{k+1} = f(x_{\nu(k+1)}) &= \max_{0\leq j\leq \min\{k+1,M-1\}} f(x_{k+1-j}) \\ &\leq \max_{0\leq j\leq \min\{k,M-1\}+1} f(x_{k+1-j}) \\ &= \max\{\max_{0\leq j\leq \min\{k,M-1\}+1} f(x_{k+1-j}), f(x_{k+1})\} \\ &= \max\{f(x_{\nu(k)}), f(x_{k+1})\}. \end{aligned} \tag{3.26}$$

Mas, $f(x_{k+1}) \leq W_k + \zeta_k$ por hipótese e, além disso, $f(x_{\nu(k)}) = W_k \leq W_k + \zeta_k$. Logo, pela desigualdade (3.26), tem-se que $W_{k+1} \leq W_k + \zeta_k$ para todo k. Um argumento indutivo permite estabelecer que:

$$W_{k+1} = f(x_{\nu(k+1)}) \leq f(x_{\nu(0)}) + \sum_{i=1}^{k} \zeta_i \leq f(x_0) + \zeta, \tag{3.27}$$

e, como $f(x_{k+1}) \leq W_{k+1}$, tem-se que $x_k \in \bar{\mathcal{L}}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0) + \zeta\}$ para todo k , provando, assim, o item 1 desta proposição.

Para provar o segundo item, observemos que, por hipótese, existe um índice $\bar{k}$ suficientemente grande tal que $f(x_{k+1}) \leq f(x_{\nu(k)})$ para todo $k \geq \bar{k}$. Logo, pela desigualdade (3.26), tem-se que a sequência $\{W_k\}_{k>\bar{k}}$ é monotonamente não-crescente. ■

Proposição 3.2 *Seja $f : \mathbb{R}^n \to \mathbb{R}_+$ uma dada função e $\{x_k\} \subset \mathbb{R}^n$ uma sequência de pontos tais que:*

$$f(x_{k+1}) \leq W_k + \zeta_k - cf(x_k), \tag{3.28}$$

onde W_k é definido como em (3.12), $c > 0$ $\zeta_k > 0$ para todo k e $\sum_{i=1}^{\infty} \zeta_i = \zeta < \infty$ e, além disso, existe um índice $\bar{k}$ tal que para todo $k > \bar{k}$ se verifica

$$\zeta_k - \bar{c} f(x_k) < 0, \ \ com\ \bar{c} > 0 \tag{3.29}$$

Então $x_k \in \bar{\mathcal{L}}_0$ para todo k e, ainda,

$$\lim_{k\to\infty} f(x_k) = \lim_{k\to\infty} W_k = 0. \tag{3.30}$$

Prova: A condição (3.28) implica que a desigualdade (3.24) é satisfeita para todo k, logo, pela Proposição 3.1, $x_k \in \bar{\mathcal{L}}_0$ para todo k e, além disso, utilizando a mesma proposição tem-se que a desigualdade (3.29) garante que a sequência $\{W_k\}_{k>\bar{k}}$ é monotonamente não-crescente e, como $W_k \geq 0$, deve haver um limite $W_* \geq 0$ para esta sequência, conforme $k \to \infty$.

Suponhamos, por absurdo, que $W^* \neq 0$, então existe $k_1 \in \mathbb{N}$ tal que, para todo $k > k_1$ tem-se $W_k > t$, para algum $t > 0$. E, pelo Lema 3.1, deve existir um índice k_2 tal que $|-\zeta_k + cf(x_k)| > t_2 > 0$ para todo $k > k_2$.

Podemos supor, sem perda de generalidade, que $k_2 \geq \bar{k}$. Assim sendo, tem-se, por hipótese, que

$$f(x_{k+1}) \leq W_k - t_2,\ k > k_2 \geq \bar{k}. \tag{3.31}$$

Considerando $k \geq k_2 + M + 1$, tem-se que $\nu(k) - 1 \geq k - M - 1 \geq k_2$ e, consequentemente, usando a definição (3.13) e a equação (3.31), temos que:

$$f(x_{\nu(k)}) \leq f(x_{\nu(\nu(k)-1)}) - t_2. \tag{3.32}$$

Tomando limite em ambos os lados, e lembrando que $f(x_{\nu(k)}) \to W_*$, temos que $t_2 = 0$, o que é uma contradição. ■

A fim de fixar parâmetros e procedimentos que possibilitem a demonstração dos resultados de convergência, vamos formalizar o algoritmo de busca linear com o critério (2.21) nos moldes do algoritmo definido em [30].

Algoritmo 3.5 *Busca linear não-monótona*

Parâmetros: $\gamma \in (0,1)$, $a \in (0,1]$, $0 < \xi_{\min} < \xi_{\max} < 1$, $\mu \in (0,1)$, $\zeta_k > 0$ *para todo* k *e* $\sum_{i=1}^{\infty} \zeta_i = \zeta < \infty$.

1. $\alpha = a$ *e* $j = 0$
2. *Enquanto* $f(x_k + \alpha d_k) > W_k + \zeta_k - \gamma\alpha^2 f(x_k)$, *faça:*
   (a) *Se* $\alpha < \mu a$ *então faça* $\lambda = 0$, $\delta = \alpha$ *e termine.*
   (b) *Escolha* $\xi \in [\xi_{\min}, \xi_{\max}]$ *e faça* $\alpha = \alpha\xi$, $j = j + 1$.
3. *Faça* $\lambda = \alpha$ *e termine.*

A Proposição 3.1 de [30] estabelece que a busca linear proposta naquele trabalho está bem definida. A Proposição 3.3, a seguir, é uma adaptação daquele resultado e permite constatar que nossa estratégia de busca linear também está bem definida.

Proposição 3.3 *Suponha que* $F(x_k) \neq 0$ *e considere um valor* $\mu \in (0,1)$ *dado, associado a* x_k. *Então o Algoritmo 3.5 determina, em um número finito de iterações, um escalar* $\lambda \in [0, a]$ *tal que:*

$$f(x_k + \lambda d_k) \leq W_k + \zeta_k - \gamma\lambda^2 f(x_k) \tag{3.33}$$

e, além disso, uma das seguintes condições é assegurada:

1. $\lambda = 0$ *e*

$$\|F(x_k + \delta d_k)\|^2 > W_k + \zeta_k - \gamma\delta^2\|F(x_k)\|^2 \geq (1 - \gamma\delta^2)\|F(x_k)\|^2 + \zeta_k, \tag{3.34}$$

   com $\delta < \mu a$; *ou*

2. $\lambda \geq \xi_{\min}\mu a$.

Prova: Dado que o tamanho de passo α é reduzido, a partir do valor a e a cada iteração, por um escalar $\xi \leq \xi_{\max} < 1$, o Algoritmo 3.5 ou termina no Passo 2 com $\lambda = 0$ (o que implica a satisfação do critério (3.33)) ou no Passo 3, onde encontra um valor não nulo para λ que satisfaz o critério de aceitação (3.33).

Em conformidade com os Passos 1 e 2 do Algoritmo 3.5 e, considerando que $\|F(x_k)\|^2 \leq W_k$ e $\gamma\delta < \gamma\mu a < 1$, caso o algoritmo termine no Passo 2 tem-se que $\lambda = 0$ e, além disso, a condição (3.34) é satisfeita.

Caso o algoritmo tenha sido encerrado no Passo 3, tem-se que, ou o tamanho inicial do passo é aceito sem reduções (neste caso $\lambda = a$), ou é encontrado um tamanho de passo λ tal que $\lambda/\xi > \mu a$ é aceito. Em ambas as situações é satisfeita a condição do item 2. ■

Através de uma simples adaptação do Lema 8.2.1 do livro de Kelley [34], Grippo e Sciandrone [30] estabeleceram o resultado da Proposição 3.4, que irá garantir que a busca linear ao longo de uma direção suficientemente próxima da direção de Newton irá terminar com um valor diferente de zero para λ desde que o ponto em questão satisfaça às seguintes condições:

Hipótese 3.1 *Existem valores positivos m_J e L_J tais que J é Lipschitz contínua com constante L_J no conjunto convexo $\Omega \subset \mathbb{R}^n$ e, além disso, $\|J^{-1}(y)\| \leq m_J$ para todo $y \in \Omega$.*

Proposição 3.4 *Considere $x \in \mathbb{R}^n$ tal que $F(x) \neq 0$ e, além disso, satisfaz as condições da Hipótese 3.1 para algum conjunto do tipo $\Omega = \{y \in \mathbb{R}^n \mid \|x - y\| \leq r\}$, com $r > 0$. Se $d \in \mathbb{R}^n$ é um vetor que satisfaz a condição Newton inexato:*

$$\|J(x)d + F(x)\| \leq \eta \|F(x)\| \tag{3.35}$$

com $\eta \leq \bar{\eta} < (1 - \gamma)$ e $\gamma \in (0, 1)$, então, temos que:

$$\|F(x + \lambda d)\| \leq (1 - \gamma\lambda)\|F(x)\| \tag{3.36}$$

para $\lambda \in [0, \lambda(x)]$, onde

$$\lambda(x) = \min\left(\frac{r}{m_J(1 + \bar{\eta})\|F(x)\|}, \frac{2(1 - \gamma - \bar{\eta})}{(1 + \bar{\eta})^2 m_J^2 L_J \|F(x)\|}\right). \tag{3.37}$$

Prova: Ver Lema 8.2.1 de [34] ■

Cabe notar que a Proposição 3.4 irá garantir um resultado análogo para a busca linear que estamos propondo, pois se um valor para λ satisfaz a condição (3.36), se definirmos como função de mérito $f(x) = \|F(x)\|^2$ e, considerando que tanto o valor de λ quanto o de $(1 - \lambda\gamma)$ pertencem ao intervalo $[0, 1]$, temos que:

$$\begin{aligned} \|F(x_k + \lambda d)\|^2 &\leq (1 - \lambda\gamma)^2 \|F(x_k)\|^2 \leq (1 - \lambda\gamma)\|F(x_k)\|^2 \\ &\leq \|F(x_k)\|^2 - \lambda^2\gamma\|F(x_k)\|^2 < W_k + \zeta_k - \gamma\lambda^2\|F(x_k)\|^2. \end{aligned} \tag{3.38}$$

O último resultado necessário à demonstração do resultado de convergência do algoritmo desenvolvido irá determinar condições para que o método FDGMRES forneça a direção requerida pelo método de Newton inexato. Como veremos, essas condições dependem das características do problema, mas estabelecem que, para valores suficientemente pequenos do parâmetro σ, utilizado nas

fórmulas de diferenças finitas, é possível encontrar a direção que procuramos. A demonstração do resultado pode ser encontrada em [31] e segue, essencialmente, os resultados apresentados em [34] (Proposição 6.2.1).

Proposição 3.5 *Dado $x_k \in \mathbb{R}^n$, suponha que $F(x_k) \neq 0$ e que x_k satisfaz a Hipótese 3.1 em um dado conjunto convexo Ω_k tal que $x_k \in \Omega_k$, com $L_J = L_k$ e $m_J = c_k$. Considere:*

$$\hat{\sigma}_k = \frac{1}{2n^{1/2}L_k c_k} \tag{3.39}$$

e

$$C_k = 4n^{1/2}L_k c_k. \tag{3.40}$$

Então, para cada $\sigma \in (0, \hat{\sigma}]$ e, para cada $\eta_k \in (0,1)$, o procedimento FDGMRES determina uma direção d_k satisfazendo

$$\|J(x_k)d_k + F(x_k)\| \leq (\eta_k + C_k\sigma)\|F(x_k)\|. \tag{3.41}$$

Prova: Ver [31]. ∎

Embora a Proposição 3.5 garanta a existência de um passo d_k que irá cumprir a condição Newton inexato (3.35) para qualquer valor de $\eta > 0$, cabe destacar que, na prática, dificilmente será possível obter o valor da constante de Lipschitz L_J e do parâmetro m_J, e, se o valor de σ na equação (3.41) não for suficientemente pequeno, existe a possibilidade de que a direção d_k requerida não seja encontrada. Além disso, também é possível que, comparativamente com o valor de η quando utilizado o método GMRES clássico, o parâmetro η deva ser mais restrito se estivermos trabalhando com o método FDGMRES e desejarmos obter a mesma direção que obteríamos pelo algoritmo que utiliza efetivamente a matriz Jacobiana. Esse fato, no entanto, não modifica a taxa de convergência do método Newton-FDGMRES, já que, se $\eta_k + \sigma_k \to 0$, a convergência superlinear (ou quadrática) característica do método de Newton inexato é preservada.

O resultado a seguir estabelece a convergência do Algoritmo 3.4 (HIB1).

Teorema 3.4 *Definindo $f(x_k) = \|F(x_k)\|^2$, seja $\{x_k\}$ a sequência gerada pelo Algoritmo HIB1 em que os critérios $CA1$ e $CA2$ são ambos dados por:*

$$f(x_k + \lambda_k d_k) \leq W_k + \zeta_k - \gamma\lambda_k^2 f(x_k) \tag{3.42}$$

em que $\zeta_k > 0$ para todo k, $\sum_{k=1}^{\infty}\zeta_k = \bar{\zeta} < \infty$ e, além disso, satisfaz:

$$\zeta_k - \bar{c}f(x_k) < 0 \text{ com } \bar{c} > 0 \tag{3.43}$$

para todo k maior ou igual a algum $\bar{k} \in \mathbb{N}$.

Assuma que existe $r > 0$ tal que, para todo $x \in \bar{\mathcal{L}}_0$, a bola fechada $\bar{B}(x, r)$ está contida em um conjunto aberto e convexo Ω que satisfaz a Hipótese 3.1. Então, o Algoritmo HIB1 está bem definido e, ou termina em algum ponto x_k tal que $F(x_k) = 0$, ou gera uma sequência infinita $\{x_k\}$ tal que:

$$\lim_{k\to\infty} F(x_k) = 0. \tag{3.44}$$

Prova: Inicialmente, notemos que a satisfação da condição (3.42) e das demais hipóteses do Teorema garantem que podemos utilizar o primeiro resultado da Proposição 3.1, o que assegura que $x_k \in \bar{\mathcal{L}}_0$ para todo $k \in \mathbb{N}$.

Vamos provar agora que não existe a possibilidade de que a busca linear do algoritmo pare em algum ponto x_k de forma que $\lambda_k = 0$. Suponhamos, por absurdo, que isto ocorra. Pela configuração do algoritmo HIB1, isso não aconteceria no Passo 5, pois, para todo k tem-se que $\lambda_+ \geq \tau_{\min}^{NBL_{\max}} > 0$ e $\lambda_- \geq \tau_{\min}^{NBL_{\max}} > 0$. Sendo assim, existe um elemento x_k da sequência de pontos gerados pelo algoritmo onde $\lambda = 0$ em todas as iterações internas da busca linear realizada no Passo 8 (isto é, em todas as iterações onde são reduzidos os valores de σ, η e μ). Denotando por $\{t\}$ a sequência de índices utilizados para contar o número de vezes que a busca linear é chamada na iteração k, tem-se que as sequências $\{\sigma(t)\}$, $\{\eta(t)\}$ e $\{\mu(t)\}$ convergem a zero quando t tende ao infinito.

Como $x_k \in \bar{\mathcal{L}}_0 \subset \Omega$, segue da Proposição 3.5 que se tivermos:

(i) $\sigma(t) \leq \frac{1}{2n^{1/2}L_J m_J}$ e

(ii) $\eta(t) + C\sigma(t) < 1 - \gamma$, (com $C = 4n^{1/2}L_J m_J$),

então o método Newton-FDGMRES irá encontrar uma direção d_k tal que:

$$\|J(x_k)d_k + F(x_k)\| \leq (\eta(t) + C\sigma(t))\|F(x_k)\| < (1 - \gamma)\|F(x_k)\|. \tag{3.45}$$

A Proposição 3.4, cujas hipóteses estão plenamente satisfeitas já que $\bar{B}(x, r) \subset \Omega$, permite estabelecer que a condição de decréscimo será satisfeita no intervalo $[0, \bar{\lambda}(x_k)]$. Veremos agora que, para j suficientemente grande $\alpha_j \leq \bar{\lambda}(x_k)$ (aqui, o valor α_j é o tamanho de passo na j-ésima tentativa de busca linear na iteração k).

De fato, como $a \in (0, 1]$ e $\xi \in [\xi_{\min}, \xi_{\max}]$, tem-se que $\xi_{\min}^j \leq \alpha_j \leq \xi_{\max}^j$ e, portanto, é possível escolher um inteiro j_* que satisfaz

$$j_* \geq \max\left\{0, \frac{\log(\bar{\lambda}_k)}{\log(\xi_{\max})}\right\} \tag{3.46}$$

e, daí, é possível estabelecer que $0 < \alpha(j_*) \leq \bar{\lambda}(x_k)$. Como $\mu(t) \to 0$ quando $t \to \infty$, temos, para valores de t suficientemente grandes, que $\mu(t) < \xi_{\min}^{j_*} \leq \alpha(j_*)$, de forma que o algoritmo de busca

linear irá terminar com um valor positivo para λ_k, o que contradiz a hipótese de que $\lambda_k = 0$. Assim, a menos que $F(x_k) = 0$, é possível obter o ponto x_{k+1} a partir de x_k, o que prova que o algoritmo está bem definido.

Vamos agora averiguar que se $0 < \lambda_k \leq 1$ é computado pelo algoritmo, então a sequência $\{\lambda_k\}$ está limitada inferiormente por alguma constante $c > 0$. Novamente a demonstração será realizada por contradição. Suponhamos então, por absurdo, que existe um subconjunto infinito $K_1 \subset \mathbb{N}$ tal que

$$\lim_{k \in K_1,\, k \to \infty} \lambda_k = 0. \tag{3.47}$$

Para k suficientemente grande, tem-se que os pontos da sequência $\{x_k\}$ com $k \in K_1$ são gerados utilizando a direção obtida pelo método FDGMRES, já que os valores de λ_+ e λ_- são limitados inferiormente por um valor maior do que zero, como já vimos, $\lambda_+ \geq \tau_{\min}^{NBL_{\max}} > 0$ e $\lambda_- \geq \tau_{\min}^{NBL_{\max}} > 0$. Assim sendo, em conformidade com os procedimentos do algoritmo de busca linear do Passo 9, uma condição necessária para que a sequência satisfazendo (3.47) exista é que $\tilde{\mu}_k \to 0$ quando $k \in K_1, k \to \infty$. Ademais, quando isso ocorrer, tem-se também que $\tilde{\sigma}_k \to 0$ e $\tilde{\eta}_k \to 0$.

Pela Proposição 3.5, tem-se que quando $\tilde{\sigma}_k \leq \frac{1}{2n^{1/2}L_k m_k}$ e $\tilde{\eta}_k + C\tilde{\sigma}_k \leq \bar{\eta}$, o algoritmo FDGMRES retorna uma direção d_k tal que

$$\|J(x_k)d_k + F(x_k)\| \leq (\tilde{\eta}_k + C\tilde{\sigma}_k)\|F(x_k)\| < \bar{\eta}\|F(x_k)\|, \tag{3.48}$$

onde $\bar{\eta} < (1 - \gamma)$. E, dado que $\bar{B}(x, r) \subset \Omega$, podemos, também neste caso, aplicar a Proposição 3.4 que irá garantir um valor $\bar{\lambda}(x_k) > 0$ tal que, para todo $\lambda \in [0, \bar{\lambda}(x_k)]$ tem-se

$$\|F(x_k + \lambda d_k)\| \leq (1 - \gamma\lambda)\|F(x_k)\| \tag{3.49}$$

o que implica que λ também satisfaz

$$\|F(x_k + \lambda d_k)\|^2 \leq W_k + \zeta_k - \gamma\lambda^2\|F(x_k)\|^2. \tag{3.50}$$

Agora, como $x_k \in \bar{\mathcal{L}}_0$, tem-se que $f(x_k) \leq f(x_0) + \zeta$ e, $\|F(x_k)\|^2 \leq \|F(x_0)\|^2 + \zeta$, logo se definirmos $b = \sqrt{2\max\{\|F(x_0)\|^2, \zeta\}}$ tem-se que $\|F(x_k)\| \leq b$ para todo $k \in K_1$, e segue da expressão (3.37) que

$$\bar{\lambda}(x_k) \geq \min\left(\frac{r}{m_k(1+\bar{\eta})b}, \frac{2(1-\gamma-\bar{\eta})}{(1+\bar{\eta})^2 m_k^2 L_k b}\right) = \delta, \text{ para todo } k \in K_1. \tag{3.51}$$

Mas, por outro lado, se a hipótese de que $\lambda_k \to 0$ para $k \in K_1$ fosse verdadeira, teríamos necessariamente que $\lambda_k < 1$ para k suficientemente grande. Podemos supor, sem perda de generalidade que

isso irá ocorrer para $k \geq k_1$. Neste caso, as instruções do algoritmo de busca linear irão garantir que $\lambda_k \geq \xi_{\min}\bar{\lambda}(x_k)$ e, daí, para todo $k \in K_1$, $k \geq k_1$ tem-se

$$\lambda_k \geq \min\{1, \xi_{\min}\bar{\lambda}(x_k)\} \geq \min\{1, \xi_{\min}\delta\}, \tag{3.52}$$

o que contradiz a hipótese e nos leva a concluir que a sequência $\{\lambda_k\}$ está limitada inferiormente por uma constante $c \in (0, 1]$.

E, consequentemente, para todo k tem-se que

$$\|F(x_k + \lambda d_k)\|^2 \leq W_k + \zeta_k - c\|F(x_k)\|^2. \tag{3.53}$$

Pela condição (3.43) tem-se que, para k suficientemente grande, $\zeta_k < cf(x_k)$ e as hipóteses da Proposição 3.2 podem ser satisfeitas para $f(x_k) = \|F(x_k)\|^2$, donde segue que

$$\lim_{k\to\infty} \|F(x_k)\| = 0. \tag{3.54}$$ ■

Antes de finalizar esta seção cabe ressaltar que é possível cumprir concomitantemente as condições (3.43) e $\sum_{k=1}^{\infty}\zeta_k = \bar{\zeta} < \infty$ com $\zeta_k > 0$ para todo k. Para isso basta definir, por exemplo,

$$\zeta_k = \frac{\min\{f(x_0), f(x_k)\}}{(k+1)^{1.1}}. \tag{3.55}$$

3.3. Testes numéricos

Os testes numéricos que iremos apresentar nesta seção têm por objetivo comparar a estratégia híbrida proposta no Algoritmo `HIB1` com os algoritmos `DFSANE` original e uma versão própria do algoritmo `NM1` proposto em [30].

A estratégia `NM1` consiste num método híbrido e não monótono que utiliza a direção encontrada pelo método de Newton inexato, usando uma versão *matrix-free* do GMRES, conforme a equação (3.6). O método GMRES utilizado baseia-se na versão sem derivadas do trabalho de Pernice e Walker [43], que por sua vez, é fundamentado na estratégia *Simpler GMRES* [52].

A cada iteração do método `NM1`, a estratégia de globalização é dividida em duas etapas. Na primeira etapa é utilizada a estratégia *watchdog* [11] e o algoritmo tenta realizar um máximo de N iterações

do método de Newton inexato, sem qualquer tipo de estratégia de globalização, a fim de encontrar um ponto z que cumpra a condição:

$$\|F(z)\| \leq 0.9 \max_{0 \leq j \leq \min\{k, M-1\}} \|F(x_{k-j})\|. \tag{3.56}$$

Se após N iterações desta fase, nenhum ponto obtiver o decréscimo desejado, o algoritmo aciona sua segunda etapa, onde retorna ao ponto x_k e realiza busca linear na direção requerida pelo método de Newton inexato (previamente encontrada na fase anterior). Para esta nova fase, a condição de aceitação é modificada:

$$\|F(x_k + \lambda d)\|^2 \leq (1 - \gamma\lambda) \max_{0 \leq j \leq \min\{k, M-1\}} \|F(x_{k-j})\|^2, \tag{3.57}$$

com $\gamma = 10^{-4}$.

Para realizar os testes comparativos aqui apresentados, foram avaliadas duas estratégias distintas testadas no trabalho [30]. Na primeira, é definido o parâmetro $N = 0$, o que representa o uso do método de Newton inexato sem a estratégia *watchdog*. A segunda estratégia adota o valor $N = 20$, e realiza 20 tentativas de obtenção de um ponto satisfatório via *watchdog* antes de iniciar a busca linear na direção encontrada pelo FDGMRES no ponto x_k. A opção com valor de $N = 20$ foi a que obteve melhor desempenho nos testes realizados em [30].

Entretanto, em nosso trabalho não estamos preocupados em comparar as distintas maneiras de obter a direção requerida pelo método de Newton inexato através do GMRES, mas sim avaliar quais estratégias podemos acoplar ao método Newton-GMRES a fim de obter melhorias de desempenho. Sendo assim, para evitar que possíveis diferenças nos desempenhos do método sejam causadas pela escolha de uma direção diferente, implementamos uma versão do algoritmo `NM1` na qual a estratégia GMRES está em conformidade com aquela apresentada em [34] e que será utilizada também no algoritmo híbrido que descrevemos neste capítulo.

Ao implementar a estratégia conforme proposto em [34] utilizamos as configurações iniciais padrões, incluindo assim, uma estratégia de reortogonalização para corrigir possíveis perdas de ortogonalidade da matriz V, causadas por instabilidade numérica. No caso, sempre que, após obter um novo vetor v_{i+1} através do processo de Arnoldi, for detectado que $\|J(x_k)v_i\| + 0.001\|v_{i+1}\| = \|J(x_k)v_i\|$, o algoritmo irá proceder da seguinte maneira:

1. Para $j = 1, 2, \ldots, k$
   (a) Defina $h_{tmp} = v_{i+1}^\top v_j$
   (b) Faça $h_{j,i} = h_{j,i} + h_{tmp}$ e $v_{i+1} = v_{i+1} - h_{tmp} v_j$
2. Redefina $h_{i+1,i} = \|v_{i+1}\|$ e $v_{i+1} = v_{i+1}/h_{i+1,i}$.

Nesta fase de testes, utilizamos o algoritmo híbrido acoplado com o algoritmo DFSANE original, em duas versões, a primeira sem busca linear ($NBL = 0$), a qual chamaremos de `H1E` (algoritmo híbrido com um teste em cada direção do método do resíduo espectral), e a segunda com até 5 buscas lineares ($NBL = 5$) antes de prosseguir para o método de Newton inexato, essa última versão será denominada por `H6E`. Nas duas versões, o método de Newton inexato é utilizado na versão sem *watchdog*, isto é, definindo o parâmetro $N = 0$.

Os algoritmos híbridos serão comparados com uma versão padrão do DFSANE (ao qual chamaremos `DFSANE`) e também com duas versões do método de Newton inexato, a primeira estratégia usa o acoplamento *watchdog* e foi chamada de `NIWD` e a estratégia que corresponde ao algoritmo `NM1` sem a utilização da estratégia *watchdog* foi denominada `NI`.

Foi mantido o critério de parada utilizado nos testes realizados na Seção 2.6, definido por (2.32). O critério para aceitação de novos pontos já foi apresentado na seção 2.6 e está descrito pela desigualdade (2.21). No entanto, a sequência ζ_k foi modificada para atender as hipóteses do Teorema 3.4 e passa a ser definida como na equação (3.55).

Os parâmetros de entrada foram divididos em três grupos: aqueles necessários para a execução do método DFSANE, aqueles necessários para a execução do método de Newton inexato e, por fim, aqueles necessários para o uso conjunto das duas estratégias.

Para o primeiro grupo foram repetidos os parâmetros adotados nos testes da Seção 2.6. No segundo grupo, repetimos, sempre que possível, os parâmetros utilizados em [30]. Ressaltamos que esses parâmetros foram utilizados também nos algoritmos `NIWD` e `NI`. Como optamos por utilizar o algoritmo FDGMRES(m), definimos o número máximo de iterações em cada ciclo do método GMRES como $m = 30$ e o número máximo de ciclos $nc_{\max} = 30$. Em conformidade com o algoritmo `NM1`, o algoritmo `HIB` foi implementado de modo que declare falha por excesso de buscas lineares quando a variável λ for menor que $\lambda_{\min} = 10^{-12}$. Por fim, diferentemente do algoritmo `NM1`, onde foi utilizada uma sequência de termos forçantes constantes, tanto no nosso algoritmo, quanto no algoritmo baseado em `NM1`, utilizamos a proposta de Eisenstat e Walker em [21]:

$$\eta_k = \gamma \left(\frac{\|F(x_k)\|}{\|F(x_{k-1})\|} \right)^{\alpha}, \tag{3.58}$$

com $\gamma = 1$ e $\alpha = 0.5(1 + \sqrt{5})$. Para este parâmetro adotamos ainda o intervalo de salvaguarda $[10^{-6}, 10^{-2}]$.

A redução do tamanho de passo foi realizada de duas maneiras distintas, conforme a etapa. O objetivo é fidelizar a estratégia de redução de passo com os algoritmos originais usados para comparação: `DFSANE` e `NM1`. Para o caso de reduções do tamanho do passo na fase I foi adotada a estratégia já descrita no capítulo anterior (ver equações (2.29) e (2.30)). Para a etapa correspondente ao método

de Newton inexato, a atualização do tamanho do passo foi feita através de bissecção, o que já está descrito no Algoritmo 3.4.

Ao Algoritmo 3.4 é preciso acrescentar alguns critérios de parada, prevendo possíveis falhas de execução. Na prática, há quatro situações em que o procedimento irá parar declarando falha de execução. A seguir, descrevemos os critérios utilizados para a parada do algoritmo, acompanhados das notações que iremos adotar no restante deste texto:

EII por excesso de iterações internas, quando não conseguir obter uma direção satisfatória após 30 ciclos de 30 iterações do GMRES(m);

EST por estagnação, quando o tamanho do passo for inferior a 10^{-12};

EAVF por excesso de avaliações de função, quando atingir a quantidade de 10000 avaliações de função;

PPF por dificuldades numéricas decorrentes da aritmética de ponto flutuante, ocasionando *overflow* ou *underflow*.

Lembramos que o algoritmo declara sucesso quando a condição (2.32) é verificada, neste caso denotaremos a finalização do Algoritmo com **C**.

Os algoritmos híbridos, bem como os algoritmos `NIWD` e `NI` serão interrompidos quando ocorrer qualquer uma dessas cinco situações. O algoritmo `DFSANE` irá parar em qualquer uma destas situações exceto, obviamente, falha por excesso de iterações internas.

Inicialmente, realizamos os testes com os mesmos problemas utilizados para os testes da seção 2.6, propostos em [14] e que podem ser encontrados nas tabelas A.1 e A.2. Os testes foram realizados com x_0 padrão indicado em [14] para cada problema e para as dimensões 100, 500, 1000, 2000 e 5000. Novamente, utilizamos o *software* Matlab 7.0, em uma máquina com processador Intel(R) Core I3-2100 3.10GHz e 4Gb de memória RAM.

Embora, como já foi dito na seção 2.6, esses problemas apresentassem valores iniciais inadequados, decidimos realizar estes testes preliminares para validar o funcionamento do algoritmo híbrido, HIB1, aqui proposto. A Tabela 3.1 contém, para cada algoritmo, a porcentagem de ocorrência de cada critério de parada para este conjunto de testes.

Os gráficos da Figura 3.2 apresentam o perfil de desempenho para esse teste, de onde podemos tirar as primeiras conclusões referentes à eficiência dos métodos (neste capítulo, os gráficos de perfil de desempenho seguiram o mesmo padrão utilizado no capítulo 2). É possível observar uma relação inversamente proporcional entre o uso da direção `DFSANE` e o número de iterações, ou seja, quanto mais um algoritmo tende a utilizar a estratégia `DFSANE` antes de prosseguir para o método de Newton

inexato, mais iterações esse algoritmo tende a realizar. Esse resultado já era esperado, visto que o método `DFSANE` é caraterizado por realizar muitas iterações. Como esse fato é compensado pelo baixo custo computacional (se comparado aos métodos Newton e Newton inexato) para a obtenção da direção de busca, consideramos que o número de iterações não é um fator determinante para avaliar a eficiência nesta nova comparação.

	EII	**EST**	**EAVF**	**PPF**	**C**
`H1E`	19%	0%	0%	0%	81%
`DFSANE`	0%	2%	25%	0%	73%
`H6E`	15%	3%	1%	0%	81%
`NIWD`	4%	1%	0%	5%	90%
`NI`	4%	1%	0%	5%	90%

Tabela 3.1: Resumo dos resultados - problemas extraídos de [14] - valor inicial padrão

Além disso, como esse trabalho trata somente de estratégias que não fazem o uso de derivadas, as avaliações da função F são responsáveis por uma grande porcentagem dos cálculos realizados, o que nos levou a utilizar o número de avaliações de função como principal medida de desempenho. O tempo de execução será adotado como medida secundária de nossa análise.

Antes de começar a análise dos gráficos da Figura 3.2, alertamos para um desempenho parecido dos algoritmos `DFSANE` e `H6E`, o que faz com que as curvas correspondentes a esses algoritmos praticamente se sobreponham nos gráficos referentes às medidas avaliações de função e tempo, especialmente para valores de τ próximos de 1.

Feitas essas observações, podemos destacar a eficiência dos algoritmos que utilizam a direção do resíduo espectral (exclusivamente ou numa primeira fase), a qual é paga com a perda de robustez. Entre os algoritmos que usam essa estratégia, aqueles que tendem a utilizá-la com mais frequência (`H6E` e `DFSANE`) tiveram um desempenho superior aos demais, quando medido em termos de avaliações de função e perdem, com uma diferença pequena, para os algoritmos `NI` e `NIWD` quando avaliamos o tempo. Ressaltamos ainda que esse desempenho inferior é causado, possivelmente, por conta do menor número de problemas em que esses algoritmos tiveram sucesso. Sendo assim, num primeiro momento podemos valorizar somente a eficiência do algoritmo híbrido. No entanto, no próximo conjunto de testes, criamos uma situação mais difícil, com pontos diferentes do padrão e o algoritmo `H6E` se mostrará, também, robusto.

Assim como fizemos na Seção 2.6, realizamos um novo conjunto de testes utilizando os mesmos problemas, mas, agora, com as dimensões $n = 100$, 500, 1000, 2000 e 5000 e pontos inicias gerados aleatoriamente na vizinhança do ponto inicial originalmente proposto. Para análise dos resultados,

reunimos os dados na Tabela 3.2 que segue o mesmo padrão da Tabela 3.1 e na Figura 3.3, onde são mostrados os perfis de desempenho.

	EII	**EST**	**EAVF**	**PPF**	**C**
H1E	54.5%	0%	0%	6.4%	39.1%
DFSANE	0%	1.5%	44.7%	3%	50.8%
H6E	39.9%	0.4%	3%	3.4%	53.2%
NIWD	57.2%	0%	0%	3.6%	39.2%
NI	57.2%	0%	0%	3.6%	39.2%

Tabela 3.2: Resumo dos resultados - problemas extraídos de [14] - valor inicial aleatório

Nesse novo teste, podemos averiguar a eficiência e a robustez do algoritmo híbrido. O algoritmo H6E foi o mais robusto nesse conjunto de testes. Quando comparamos o desempenho em termos de eficiência, H6E fica em segundo lugar, com desempenho inferior somente ao algoritmo DFSANE. Além disso, podemos observar que tanto o algoritmo H6E quanto o algoritmo DFSANE diminuíram a diferença em número de iterações com relação aos algoritmos NIWD e NI.

Nesta etapa, testamos ainda o algoritmo híbrido com as modificações que propusemos no algoritmo DFSANE no capítulo anterior. Foram modificados os algoritmos H1E e H6E, de forma que, para cada um deles, foram criadas três novas versões:

1. adição da direção d_3, de maneira similar ao que foi feito no algoritmo DFSANE-D3. Em correspondência à notação utilizada neste capítulo e no capítulo anterior, vamos denominar os novos algoritmos de H1E-D3 e H6E-D3;

2. ordenação das direções a serem testadas de acordo com o sucesso da última iteração, conforme foi feito no Algoritmo DFSANE-01, originando, assim, os algoritmos H1E-02 e H6E-02;

3. utilização das duas estratégias anteriores conforme o algoritmo DFSANE-02D3, o que irá definir os algoritmos H1E-02D3 e H6E-02D3.

Os resultados referentes a esses novos algoritmos encontram-se na Tabela 3.3, na qual, para auxiliar a comparação, repetimos os dados dos algoritmos H1E e H6E e, também, refizemos os testes com os algoritmos DFSANE-D3, DFSANE-02 e DFSANE-02D3, utilizando, agora, os parâmetros definidos nesta seção. As Figuras 3.4 e 3.5 apresentam os gráficos com a análise de desempenho para esse teste, na Figura 3.4 encontram-se os resultados para as versões modificadas do algoritmo H1E e a Figura 3.5 apresenta os resultados para as novas versões do algoritmo H6E.

Em termos de eficiência, as modificações no algoritmo H6E proporcionaram um desempenho ligeiramente inferior quando é feita a comparação com o algoritmo sem modificações, principalmente para aqueles algoritmos em que a terceira direção é utilizada. Já para as modificações no algoritmo H1E, podemos observar que há um baixo impacto acarretado pelo uso da estratégia de ordenação, já que os

algoritmos H1E e H1E-O2 tiveram desempenho similar, assim como H1E-D3 e H1E-O2D3. Observamos também que a adição da terceira direção tornou o algoritmo ligeiramente menos eficiente.

As observações do parágrafo anterior, se assemelham com o que fora notado na seção de testes do capítulo anterior, utilizando as mesmas modificações no algoritmo DFSANE.

Além disso, pela Tabela 3.3, podemos observar que os algoritmos que utilizam uma terceira direção são mais robustos do que os demais, além de que o uso de uma nova estratégia de reordenação faz com que a robustez diminua. Isso também já fora observado nos testes anteriores, de forma que podemos concluir que, de maneira geral, as modificações que trazem melhorias para o algoritmo DFSANE tendem a preservar tais melhorias quando o algoritmo é inserido num contexto híbrido.

	EII	**EST**	**EAVF**	**PPF**	**C**
DFSANE	0%	1.5%	44.7%	3%	50.8%
DFSANE-O2	0%	1.5%	45.2%	3%	50.3%
DFSANE-D3	0%	1.5%	44.3%	3%	51.2%
DFSANE-O2D3	0%	1.5%	45.2%	3%	50.3%
H1E	54.5%	0%	0%	6.4%	39.1%
H1E-O2	54.5%	0%	0%	6.4%	39.1%
H1E-D3	53.8%	0%	0%	6.2%	40.0%
H1E-O2D3	53.8%	0%	0%	6.2%	40.0%
H6E	39.9%	0.4%	3%	3.4%	53.3%
H6E-O2	40.2%	0.3%	2.8%	3.5%	53.2%
H6E-D3	38.1%	0.2%	4.8%	3.4%	53.5%
H6E-O2D3	39.2%	0.1%	5.7%	3.4%	54.2%

Tabela 3.3: Resumo dos resultados - problemas extraídos de [14] - valor inicial aleatório - DFSANE modificado

Devido ao baixo número de problemas em que houve convergência, se comparados com o teste anterior, e também à própria escolha dos pontos iniciais, consideramos esse teste mais difícil que o teste anterior. Sendo assim, ao que parece, o algoritmo híbrido aqui desenvolvido tem um desempenho superior em problemas mais difíceis. Para confirmar essa tendência, trabalhamos com os primeiros 21 problemas propostos em [39]. Inicialmente realizamos os testes utilizando tanto as dimensões quanto os pontos iniciais sugeridos naquele trabalho, os resultados podem ser encontrados na Tabela 3.4. Nessa tabela, para cada algoritmo testado e para cada problema, são apresentados três dados: **final**, que indica a finalização do problema e segue o padrão de siglas já adotado nas Tabelas 3.1 e 3.2, **Iter** onde pode ser encontrado o número de iterações e **Avalf** que indica o número de avaliações de função. A Figura 3.6 apresenta os gráficos com os perfis de desempenho para este teste. Alertamos que, nesta figura e nos gráficos referentes ao tempo de execução, a curva correspondente ao Algoritmo NIWD sobrepõe a curva referente ao Algoritmo NI em sua totalidade, de maneira que não é possível visualizar esta última no gráfico. O mesmo ocorre, mas somente em alguns trechos, nos gráficos referentes às outras medidas de desempenho.

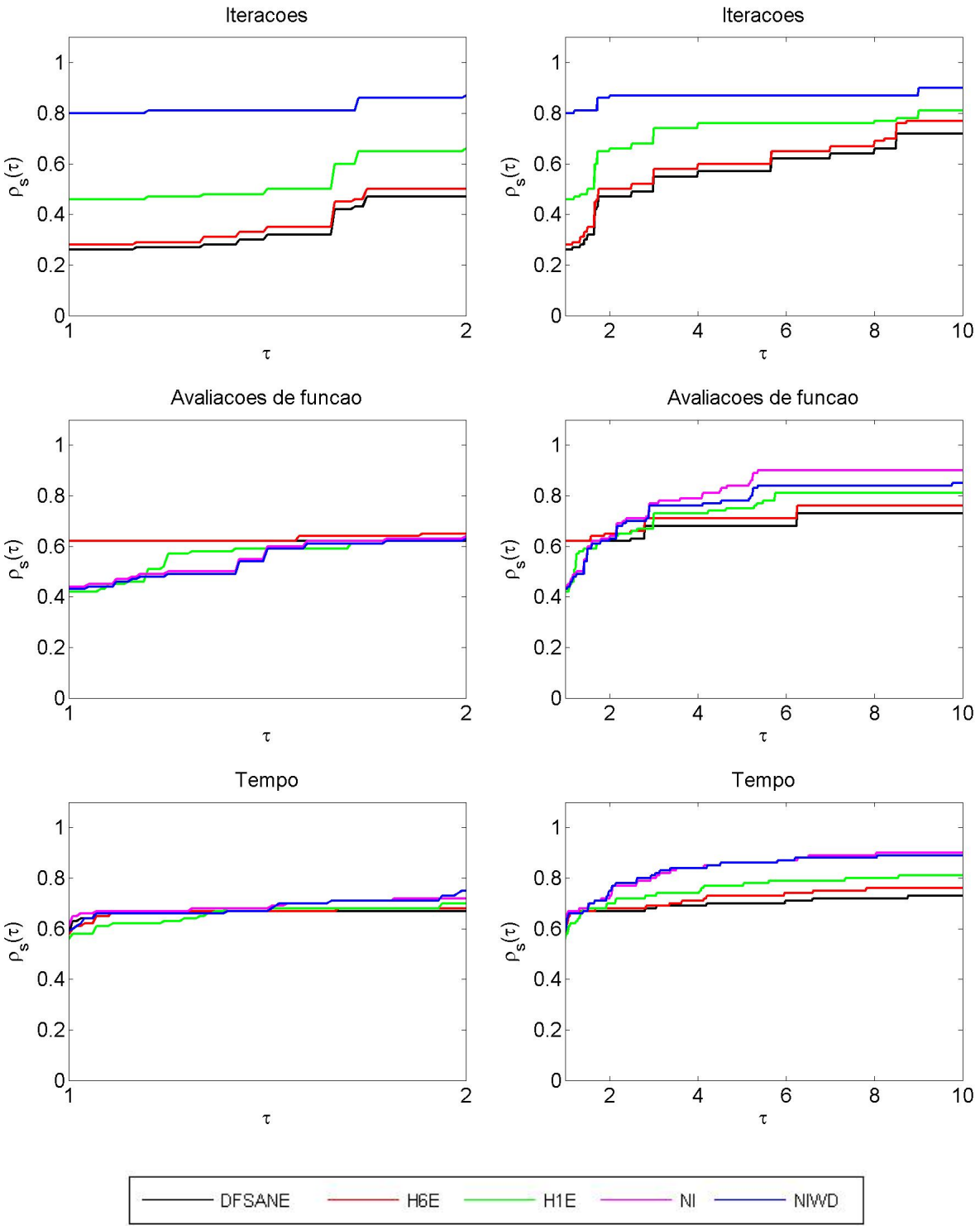

Figura 3.2: Desempenho dos algoritmos `H1E`, `DFSANE`, `H6E`, `NIWD` e `NI` - problemas extraídos de [14] - valor inicial padrão

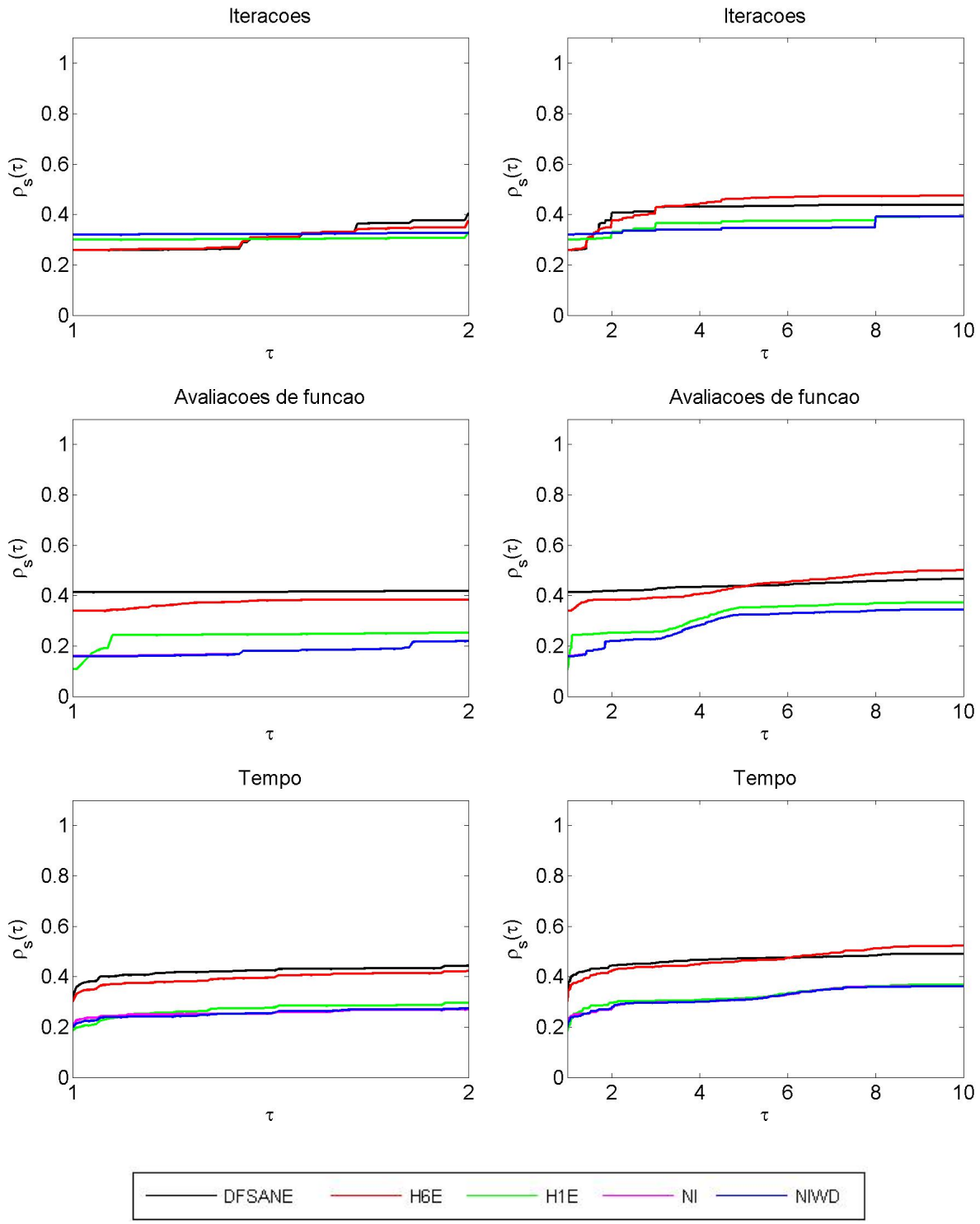

Figura 3.3: Desempenho dos algoritmos `H1E`, `DFSANE`, `H6E`, `NIWD` e `NI` - problemas extraídos de [14] - valor inicial aleatório

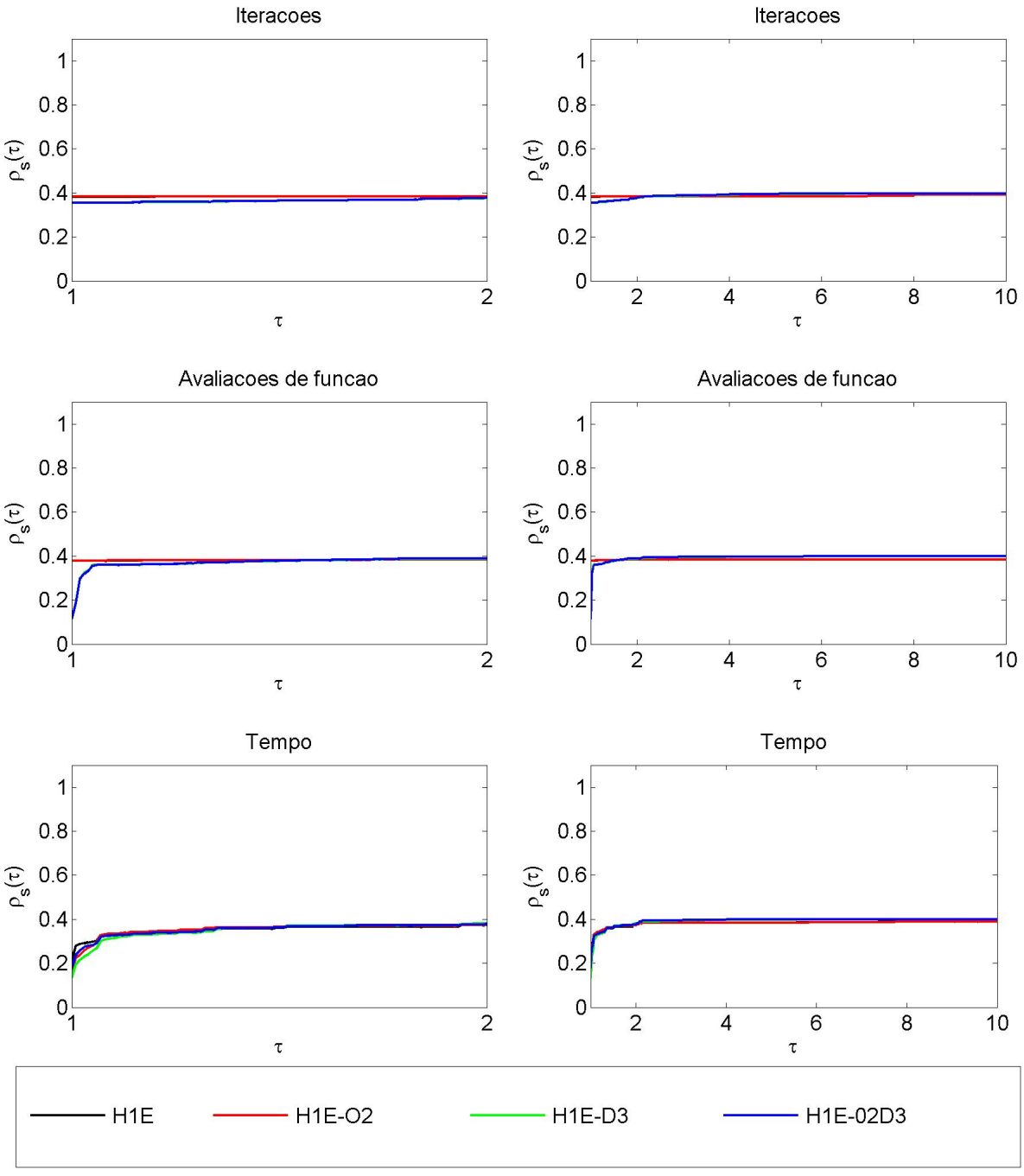

Figura 3.4: Desempenho das versões modificadas do Algoritmo `H1E` - problemas extraídos de [14] - valor inicial aleatório

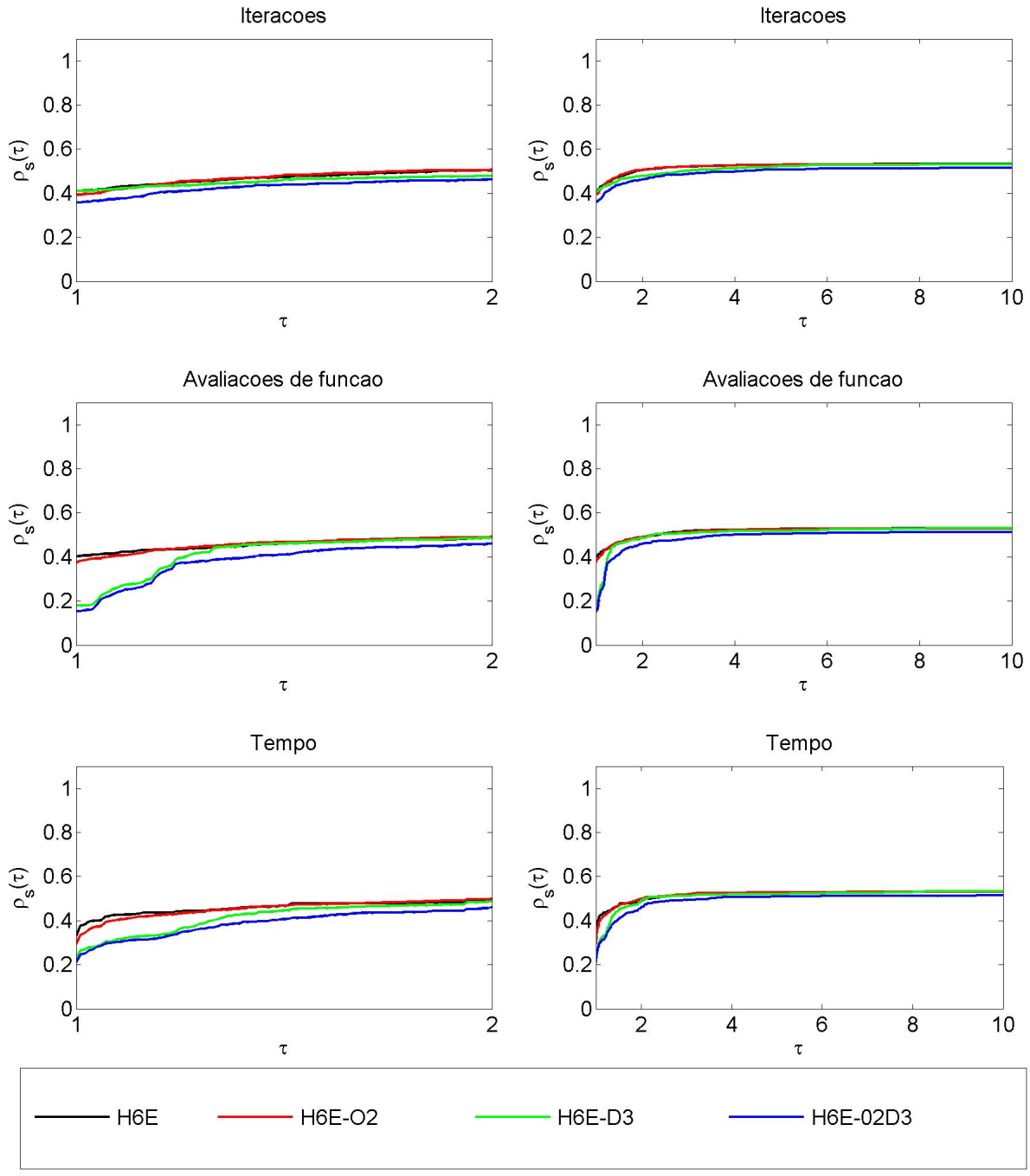

Figura 3.5: Desempenho das versões modificadas do Algoritmo `H6E` - problemas extraídos de [14] - valor inicial aleatório

	H1E			DFSANE			H6E			NIWD			NI		
p	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
1	EII	3	71	EAVF	1186	10002	EII	135	653	EII	3	69	EII	3	69
2	EII	2	61	EAVF	1112	10002	EII	576	4789	EII	2	59	EII	2	59
3	C	3	4	C	3	4	C	3	4	C	1	45	C	1	6
4	C	4	18	C	7	10	C	7	10	C	4	16	C	4	16
5	EII	0	34	EAVF	849	10002	EII	20	194	EII	0	32	EII	0	32
6	C	6	31	EAVF	1131	10001	EAVF	479	10013	C	6	29	C	6	29
7	C	7	20	C	12	19	C	12	19	C	7	18	C	7	18
8	C	11	72	C	22	33	C	22	33	C	16	80	C	11	70
9	C	13	79	C	21	28	C	21	28	C	13	77	C	13	77
10	EII	1	35	EAVF	910	10001	EII	160	1269	C	3	20	C	3	20
11	C	7	27	EAVF	875	10001	C	193	1051	C	7	25	C	7	25
12	C	10	82	EAVF	770	10002	C	64	531	C	10	80	C	10	80
13	C	8	39	C	25	31	C	25	31	C	8	37	C	8	37
14	C	17	19	C	17	19	C	17	19	C	2	18	C	2	18
15	C	3	21	C	7	10	C	7	10	C	3	19	C	3	19
16	C	0	5	C	1	10	C	1	10	C	0	3	C	0	3
17	C	6	36	C	15	24	C	15	24	C	6	34	C	6	34
18	C	1	2	C	1	2	C	1	2	C	0	3	C	0	3
19	C	0	1	C	0	1	C	0	1	C	0	1	C	0	1
20	EII	0	34	EAVF	813	10001	EII	3	69	EII	0	32	EII	0	32
21	EII	7	93	EAVF	2485	10002	EAVF	2485	10002	EII	2	84	EII	2	84

Tabela 3.4: Resumo dos resultados - problemas extraídos de [39] - valor inicial padrão

Analisando a Figura 3.6, podemos notar, novamente, que o algoritmo híbrido `H6E` teve um bom desempenho em termos de eficiência nas medidas que definimos como prioritárias, sendo o melhor, empatado com `DFSANE`, quando a medida adotada é o número de avaliações de função e superior a todos os algoritmos quando avaliamos o tempo de execução. No entanto, conforme pode ser conferido na Tabela 3.4, o algoritmo não mostrou-se tão robusto, já que convergiu em 14 problemas contra a convergência em 16 problemas obtida pelos algoritmos `NIWD` e `NI` e em 15 problemas obtida por `H1E`. O algoritmo `DFSANE` foi o menos robusto neste teste, convergindo em somente 12 problemas.

Durante a realização desses testes, observamos que as estratégias de globalização são pouco acionadas, além de que, no problema 19 (*Discrete boundary value problem*) a convergência é imediata. Sendo assim, utilizamos os problemas em que o algoritmo híbrido `H1E` obteve convergência e criamos um novo conjunto de testes de maneira que pudéssemos estudar o comportamento dos algoritmos à medida que os pontos fossem se afastando da solução. Assim, foram utilizados os 18 primeiros problemas exceto o primeiro, o segundo, o quinto e o décimo. Novamente, todos os problemas foram testados com dimensão $n = 5000$.[1]

Para obter um ponto inicial mais distante da solução, conforme podemos acompanhar pela Figura 3.7, armazenamos o ponto em que o algoritmo `H1E` parou com sucesso no teste anterior e criamos a direção $d_t = x_0 - \bar{x}$, onde x_0 é o valor inicial proposto em [39] e $\bar{x}$ o ponto encontrado como solução pelo algoritmo `H1E`. Feito isso, um parâmetro ω irá determinar o quão mais longe o novo ponto inicial estará da solução do que o ponto inicial anterior. O novo ponto inicial será definido como:

$$x_{0novo} = x_0 + \omega d_t. \tag{3.59}$$

Note que para $\omega = 0$ temos o ponto inicial padrão. A Tabela 3.5 fornece a finalização de cada algoritmo conforme aumentamos o valor de ω.

Gerando os novos pontos desta maneira, todos os pontos iniciais utilizados pertencem à mesma reta. Para eliminar possíveis vícios que isso acarretaria no conjunto de testes, aumentamos o número de pontos iniciais, sendo que agora não estarão todos na mesma reta.

Os novos pontos foram escolhidos de maneira que, além de mais afastados da solução, também estivessem fora da direção determinada por d_t. Para que isso ocorresse, definimos uma direção aleatória d_{aleat} e um parâmetro υ, de forma que os novos pontos iniciais foram gerados conforme a equação:

$$x_{0novo} = x_0 + \omega d_t + \upsilon d_{aleat}. \tag{3.60}$$

[1] O problema 19 não foi testado pois, devido à sua convergência imediata, não é possível computar a direção d_t, necessária para obtenção de pontos mais distantes.

P	ω	H1E	DFSANE	H6E	NIWD	NI
3	1	C	C	C	C	C
	20	EST	C	C	C	C
	200	C	C	C	C	C
4	1	C	C	C	C	C
	20	C	C	C	C	C
	200	EST	C	EST	EST	EST
6	1	C	EAVF	EST	C	C
	20	C	C	C	C	C
	200	C	C	C	C	C
7	1	C	C	C	C	C
	20	C	C	C	C	C
	200	C	C	C	C	C
8	1	C	C	C	C	C
	20	C	C	C	C	C
	200	C	C	C	C	C
9	1	C	C	C	C	C
	20	C	C	C	C	C
	200	C	C	C	C	C
11	1	C	C	C	C	C
	20	C	EAVF	C	C	C
	200	C	EAVF	C	C	C
12	1	C	EAVF	C	C	C
	20	C	EAVF	C	C	C
	200	C	EAVF	C	C	C
13	1	PPF	EAVF	PPF	C	C
	20	EAVF	EAVF	EAVF	C	C
	200	PPF	PPF	PPF	PPF	PPF
14	1	C	C	C	C	C
	20	EST	EAVF	EST	EST	EST
	200	EST	EAVF	EST	EST	EST
15	1	C	C	C	C	C
	20	C	C	C	C	C
	200	C	C	C	C	C
16	1	C	EAVF	C	C	C
	20	C	C	C	C	C
	200	C	C	C	C	C
17	1	C	C	C	C	C
	20	C	C	C	C	C
	200	C	EAVF	C	C	C
18	1	C	C	C	C	C
	20	C	C	C	C	C
	200	C	C	C	C	C

Tabela 3.5: Resultados dos testes com problemas de [39] para distintos valores de ω

Para que a direção aleatória esteja suficientemente afastada da direção d_t exigimos, também, que o valor do cosseno do ângulo entre a direção d_{aleat} e a direção d_t não seja superior a 0.95, o que é necessário para garantir que à medida que vamos aumentando o valor de υ, a direção determinada por $x_{0novo} - x_0$ vá se afastando da direção d_t. A Figura 3.8 esclarece a obtenção do novo ponto inicial.

Os resultados finais obtidos pelos algoritmos em cada um dos testes produzidos com essa nova formulação para o ponto inicial podem ser encontrados no apêndice deste trabalho, Tabelas A.5, A.6, A.7 e A.8. A Tabela 3.6 resume os dados dos dois últimos conjuntos de testes (afastamento usando uma direção e duas direções) e a Figura 3.9 apresenta os perfis de desempenho nesses testes, com os quais iremos chegar às nossas conclusões a respeito da eficiência dos métodos.

	EII	**EST**	**EAVF**	**PPF**	**C**
`H1E`	35%	1,11%	2,22%	4,44%	57,22%
`DFSANE`	0%	1,67%	41,67%	3,33%	53,33%
`H6E`	29,44%	1,11%	3,33%	4,44%	61,67%
`NIWD`	34,44%	1,11%	1,11%	3,33%	60%
`NI`	34,44%	1,11%	1,11%	3,33%	60%

Tabela 3.6: Resumo dos resultados - problemas extraídos de [39] - pontos iniciais se afastando da solução

O algoritmo híbrido `H6E` foi o mais robusto de todos os demais nesse novo conjunto de testes e, além disso, teve um bom desempenho em termos de eficiência nas medidas prioritárias, ainda que tenha um desempenho parecido com os demais, exceto `DFSANE`, quando $\tau = 1$, pode-se observar que a curva correspondente a esse Algoritmo, logo vai se afastando das demais tanto nos gráficos correspondentes ao número de avaliação de funções quanto nos gráficos correspondentes ao tempo de execução. O Algoritmo `H1E` teve um desempenho um pouco inferior a `H6E` em termos de eficiência, mas não foi tão robusto.

O algoritmo `DFSANE` foi o destaque negativo deste teste, sendo o menos robusto e menos eficiente.

Por fim, pudemos confirmar a expectativa de que um algoritmo híbrido apresentaria melhores resultados em problemas difíceis, possivelmente por conta do maior número de direções testadas. Essa ideia ganha crédito se observarmos que há casos em que a convergência ocorre no algoritmo `H6E` e somente em um dos algoritmos baseados em um único método. Por exemplo, no Problema 8 utilizando $\upsilon = 1$ e $\omega = 1$ ou no Problema 9 utilizando $\upsilon = 20$ e $\omega = 20$, temos somente convergência de `H6E` e `DFSANE`. Exemplos para quando ocorre convergência somente em `H6E` e `NI` (e `NIWD`) são o Problema 11 utilizando $\upsilon = 20$ e $\omega = 1$ e o Problema 16 utilizando $\upsilon = 1$ e $\omega = 1$.

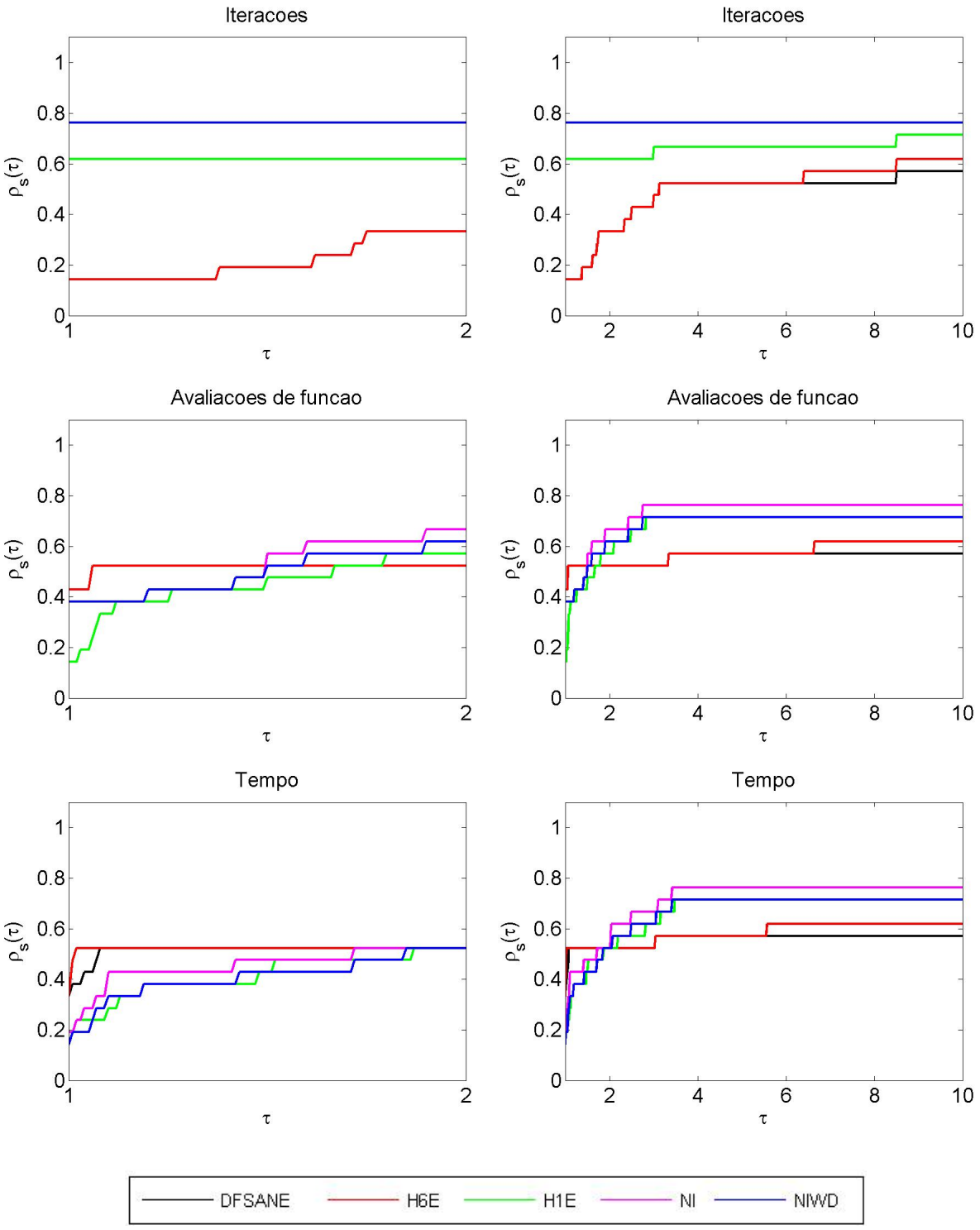

Figura 3.6: Desempenho dos algoritmos `H1E`, `DFSANE`, `H6E`, `NIWD` e `NI` - problemas extraídos de [39] - valor inicial padrão

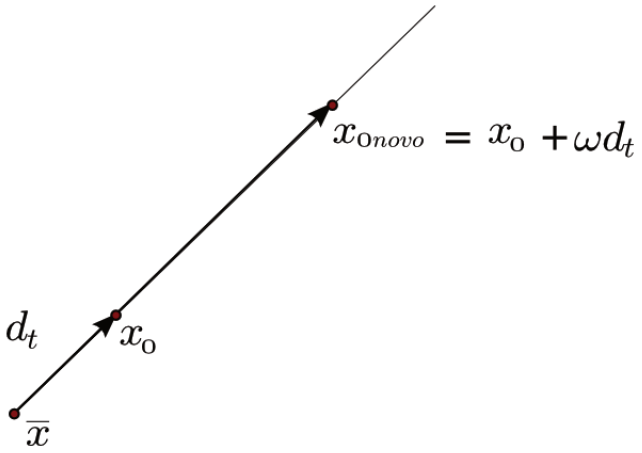

Figura 3.7: Obtenção de novos pontos iniciais

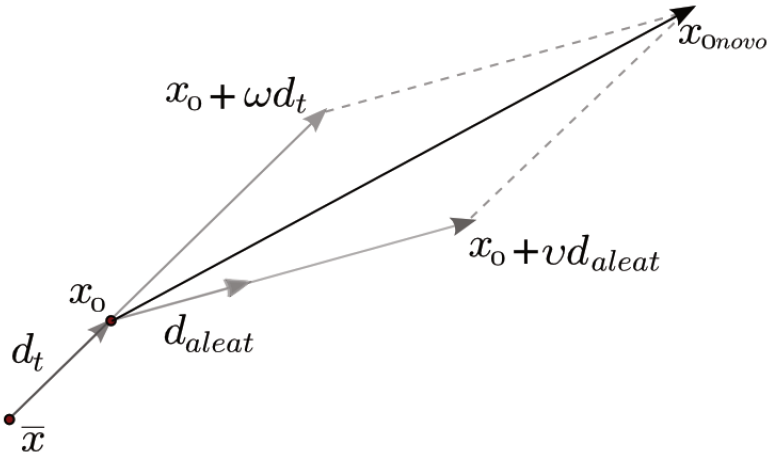

Figura 3.8: Obtenção de novos pontos iniciais utilizando a direção aleatória

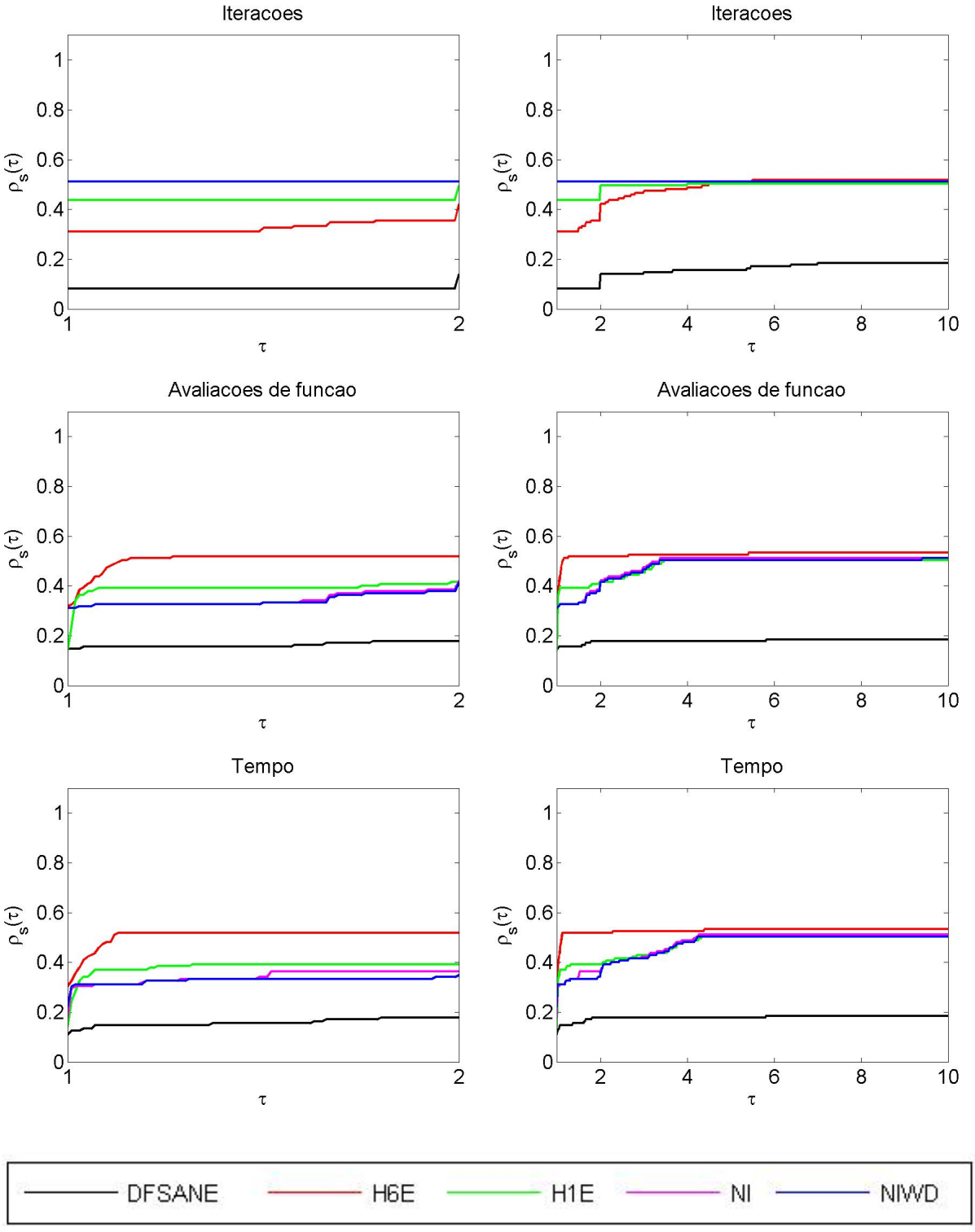

Figura 3.9: Perfis de desempenho - problemas extraídos de [39] - pontos iniciais se afastando da solução

Capítulo

4 Busca direta

Métodos de busca direta são métodos iterativos para minimização sem derivadas e, basicamente, consistem em avaliar o valor da função objetivo em um número finito de pontos e, a partir desta análise, decide-se em quais dos pontos é viável seguir insistindo na próxima iteração, e quais pontos devem ser abandonados.

Devido à sua facilidade de implementação e paralelização e, principalmente, aos resultados teóricos que garantem sua convergência global [50], esta classe de métodos tem sido objeto de um renovado interesse da comunidade científica [12, 35].

No entanto, por conta de sua baixa taxa de convergência local, métodos de busca direta não são recomendáveis para problemas suaves de grande porte, já que existem métodos mais eficientes para a resolução deste tipo de problemas, o que nem sempre acontece com problemas não suaves. Ainda assim, trabalhos recentes [29, 30] têm avaliado o acoplamento de uma fase de busca direta ao final do algoritmo. Na prática, essa fase somente é acionada quando as tentativas anteriores estiverem esgotadas, situação em que seria declarada falha de execução.

Neste capítulo, vamos propor um algoritmo em que uma estratégia de busca direta não monótona será acionada no Passo 9 do Algoritmo 3.4, quando o tamanho do passo λ torna-se demasiadamente pequeno sem, no entanto, obter um ponto que satisfaça a condição de aceitação. Nesta nova abordagem, o passo 10 é retirado. A intenção de acrescentar essa nova etapa é obter melhorias em termos de robustez nos testes numéricos, já que, na prática, essa nova etapa será acionada somente quando o algoritmo estiver prestes a declarar falha de execução.

Nessa nova fase, propomos o uso de uma estratégia de busca direta em que a sequência de pontos não necessariamente fará com que a função de mérito $f(x)$ seja não crescente, como é feito geralmente em métodos clássicos de busca direta. Além disso, desenvolvemos um novo conjunto de direções de sondagem, o qual é construído a partir de um conjunto ortogonal que contém a direção $d = -(1/\alpha_k)F(x_k)$ ou a direção $d = (1/\alpha_k)F(x_k)$.

Iniciamos este capítulo com uma introdução teórica sucedida pela apresentação do Algoritmo híbrido que propomos e dos resultados de convergência teórica. Para finalizar o capítulo, apresentamos os resultados dos testes teóricos que comprovam que a adição de uma nova fase de busca direta ao Algoritmo 3.4 proporciona melhorias em termos de robustez.

4.1. Métodos de Busca Direta

Em 1961, Hooke e Jeeves [33] definiram o termo busca direta para designar métodos iterativos para otimização sem derivadas em que se compara, a cada iteração, o valor da função objetivo em um número finito de pontos e, em seguida, decide-se em quais pontos podemos ter boas perspectivas para o prosseguimento do método, e quais pontos devem ser descartados.

Embora os primeiros trabalhos que propõem o uso de busca direta datem das décadas de 50 e 60 ([7, 22, 49]), somente depois do trabalho de Torczon [50] em 1989, onde são demonstrados resultados de convergência global, é que surge um relevante interesse da comunidade científica nesse tipo de método.

Métodos de busca direta são métodos do tipo *derivative-free* e não utilizam qualquer tipo de aproximação explícita ou implícita para derivadas. Além disso, têm a vantagem de serem fáceis de implementar e paralelizar e permitem trabalhar com funções não suaves. No entanto, são métodos de convergência lenta. Ainda que o valor da função objetivo decresça muito rapidamente nas primeiras iterações, ele tende a estagnar e convergir lentamente a partir de um certo ponto (cf. [15], Seção 2.5).

Os métodos de busca direta são separados em duas classes:

Busca direta direcional: São métodos em que um conjunto de direções de sondagem com características adequadas é definido e a busca é realizada nos pontos provenientes da adição de cada uma dessas direções ao ponto atual.

O método mais simples de busca direta direcional é o da chamada busca coordenada, que utiliza como direções de busca os vetores canônicos de $\mathbb{R}^n$ e seus respectivos simétricos.

Busca direta simplética: São métodos que baseiam sua procura em simpléticos[1] ou operações sobre simpléticos: reflexões, expansões e contrações.

[1]Simplex é a generalização da noção de triângulo ou tetraedro para uma dimensão qualquer. Especificamente, definindo o fecho convexo de um dado conjunto S como a coleção de todas as combinações convexas de elementos de S, dizemos que um politopo é o fecho convexo de um número finito de pontos $x_1, \ldots, x_p, x_{p+1}$ em $\mathbb{R}^n$. Assim sendo, podemos definir o simplex como o caso mais simples de politopo. Num simplex, os pontos $x_1, \ldots, x_n, x_{n+1}$ (agora chamados de vértices do simplex) são tais que $x_2 - x_1, \ldots, x_n - x_1$, $x_{n+1} - x_1$ são linearmente independentes [6].

O método mais conhecido dessa classe é o método Nelder-Mead [41], que é também o método de busca direta mais citado na literatura.

Nosso interesse é trabalhar com a resolução de sistemas não lineares. Assim sendo, iremos considerar como função objetivo a função de mérito definida nos mesmo moldes daquelas encontradas nos capítulos 2 e 3. Iremos focalizar nosso estudo nos métodos que utilizam busca direta direcional. Para uma análise mais detalhada sobre busca direta simplética, deve-se consultar, por exemplo, o capítulo 8 do livro de Conn, Scheinberg e Vicente [12].

Neste trabalho, a intenção é utilizar uma estratégia de busca direta direcional, seguindo a estrutura proposta por Vicente [51] para algoritmos que têm como objetivo a resolução de problemas de minimização irrestrita. A estrutura proposta generaliza aquela encontrada em [3] e está representada no Algoritmo 4.1. O objetivo deste algoritmo padrão é gerar uma sequência infinita de pontos, que convergem para um ponto estacionário da função objetivo, a menos que um ponto deste tipo seja encontrado em uma certa iteração, o que fará com que o algoritmo seja interrompido. Para encontrar um ponto x_{k+1} que obtém um dado decréscimo, cada iteração é dividida em duas etapas:

Etapa de busca: etapa não obrigatória, adotada em geral para melhorar a velocidade de convergência. Deve ser finita de modo a não interferir na análise de convergência do método.

Etapa de sondagem: nesta etapa, é definido um conjunto finito de direções, chamadas *direções de sondagem* e são avaliados os valores da função objetivo nos pontos resultantes da soma dessas direções ao ponto atual. É esta etapa que garante a convergência global do algoritmo, no entanto, para isso, o conjunto de direções de sondagem deve ter características especiais, das quais falaremos adiante.

Devido à sua flexibilidade, a etapa de busca não é considerada para a análise de convergência do algoritmo, que depende exclusivamente da etapa de sondagem. Para iniciar a análise das condições em que podemos obter a convegência teórica para o algoritmo de busca direta, vamos definir o conceito de conjunto gerador positivo, conforme estabelecido em [17].

Definição 4.1 *Um* conjunto gerador positivo *é um conjunto que gera positivamente o espaço* $\mathbb{R}^n$. *Isto quer dizer que, se* $\{d_1, d_2, \ldots, d_r\}$ *é um conjunto gerador positivo, então para todo elemento* $d \in \mathbb{R}^n$ *existem valores reais não-negativos* $a_1, a_2, \ldots, a_r$ *tais que* $d = a_1 d_1 + a_2 d_2 + \ldots + a_r d_r$.

Com isso, já podemos, também, definir bases positivas:

Definição 4.2 *Um conjunto de vetores* $\{d_1, d_2, \ldots, d_r\}$ *é dito ser* positivamente dependente *se existe um índice* i *tal que* d_i *pertence ao espaço gerado positivamente por*

$$\{d_1, d_2, \ldots, d_{i-1}, d_{i+1}, \ldots, d_r\}, \tag{4.1}$$

ou seja, se um de seus vetores pertence ao cone convexo gerado positivamente pelos demais vetores. Caso contrário, o conjunto é dito ser positivamente independente.

Uma base positiva *para $\mathbb{R}^n$ é um conjunto gerador positivo que tem a propriedade de ser positivamente independente.*

Uma base positiva com $n+1$ elementos é chamada base positiva minimal *e uma base positiva com $2n$ elementos é chamada* base positiva maximal.

Observação 4.1: Conforme pode ser conferido em [17], o número máximo e mínimo de elementos de uma base positiva para o espaço $\mathbb{R}^n$ é, respectivamente, $2n$ e $n+1$, daí as denominações "base positiva maximal" e "base positiva minimal".

Considerando o espaço $\mathbb{R}^n$, dois exemplos básicos de conjuntos geradores positivos são os conjuntos: $\{e_1, e_2, \ldots, e_n, -e\}$ (base positiva minimal) e $\{e_1, e_2, \ldots, e_n, -e_1, -e_2, \ldots, -e_n\}$ (base positiva maximal), onde e_i é o i-ésimo vetor da base canônica do espaço $\mathbb{R}^n$ e $e = (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$.

A Proposição 4.1 apresenta um resultado, devido a Lewis e Torczon [36], que estabelece uma maneira eficiente de gerar bases positivas no $\mathbb{R}^n$, a partir de outras bases positivas.

Proposição 4.1 *Se $\{d_1, \ldots, d_r\}$ é uma base positiva para o espaço $\mathbb{R}^n$ e $C \in \mathbb{R}^{n\times n}$ é uma matriz não singular, então $\{Cd_1, \ldots, Cd_r\}$ é uma base positiva para $\mathbb{R}^n$.*

Prova: Ver [36]. ∎

Dados o ponto atual x_k, uma base positiva D_k e um parâmetro λ_k, a etapa de sondagem consiste em avaliar a função objetivo[2] nos pontos determinados pelo conjunto:

$$P_k = \{x_k + \lambda_k d : d \in D_k\}. \tag{4.2}$$

Os pontos de P_k são chamados de *pontos de sondagem* e as direções do conjunto de sondagem são chamadas de *direções de sondagem*. Se não for possível encontrar um valor que cause decréscimo suficiente na função objetivo, a iteração é declarada mal sucedida e o valor do tamanho do passo λ_k é reduzido.

Antes de prosseguir, é necessário antecipar alguns esclarecimentos sobre o decréscimo necessário na função objetivo para a aceitação de novos pontos. Classicamente, existem duas maneiras de demonstrar a convergência em algoritmos de busca direta. A primeira delas utiliza um número

[2]Lembramos que no nosso caso estamos trabalhando com resolução de sistemas não lineares e, sendo assim, utilizaremos a função de mérito $f(x) = \left\|F(x)\right\|^2$ como função objetivo a ser minimizada. Por ora, como as principais referências tratam de problemas de minimização, faremos nossa análise através deste viés.

finito de bases positivas e todo ponto testado deve pertencer a uma malha inteira ou racional pré definida (o conceito de malha inteira/racional será visto posteriormente). Neste caso, não será necessário estabelecer nenhum critério de aceitação mais exigente do que o simples decréscimo. Por outro lado, se optarmos pela utilização de infinitas bases positivas sem qualquer tipo de estrutura de malha, é necessário estabelecer uma condição de decréscimo suficiente baseado em uma função forçante, que iremos definir no momento conveniente.

O modelo algorítmico descrito a seguir considera uma função $\varphi : \mathbb{R} \to \mathbb{R}$ que será utilizada no critério de aceitação para novos pontos. Assim, se desejarmos trabalhar com um número finito de bases positivas, definimos $\varphi \equiv 0$. Caso contrário, como veremos adiante, na Hipótese 4.3, φ deve ter certas características que garantam um critério adequado para aceitação de novos pontos.

Algoritmo 4.1 *Busca direta direcional*

Escolha $x_0 \in \mathbb{R}^n$ com $f(x_0) < \infty$ e os escalares $\lambda_0 > 0$, $0 < \tau_{\min} < \tau_{\max} < 1$ e $\beta \geq 1$. Defina $\mathcal{D}$ um conjunto de bases positivas e $k = 0$.

1. ***Passo de busca:*** *Tente encontrar x tal que $f(x) < f(x_k) - \varphi(\lambda_k)$ através da avaliação de f em um número finito de pontos. Caso encontre, defina $x_{k+1} = x$, declare sucesso e vá para o passo 3.*

2. ***Passo de sondagem:*** *Escolha uma base positiva D_k do conjunto $\mathcal{D}$, ordene o conjunto de sondagem $P_k = \{x_k + \lambda_k d : d \in D_k\}$. Avalie a função f nos pontos de sondagem. Se é encontrado um ponto $x_k + \lambda_k d_k$, com $d_k \in D_k$, tal que $f(x_k + \lambda_k d_k) < f(x_k) - \varphi(\lambda_k)$, defina $x_{k+1} = x_k + \lambda_k d_k$. Caso contrário, defina $x_{k+1} = x_k$ e declare fracasso.*

3. ***Atualização do tamanho do passo/parâmetro da malha:*** *Se a iteração obteve sucesso, defina $\lambda_{k+1} \in [\lambda_k, \beta\lambda_k]$. Caso contrário, defina $\lambda_{k+1} \in [\tau_{\min}\lambda_k, \tau_{\max}\lambda_k]$.*

A intenção de usar uma base positiva se deve ao resultado da Proposição 4.2, original de [17] e que garante a existência de ao menos uma direção de descida dentro de qualquer conjunto gerador positivo, desde que f seja continuamente diferenciável e que o ponto atual não seja um ponto estacionário.

Proposição 4.2 *O conjunto $\{d_1, d_2, \ldots, d_r\}$ gera positivamente $\mathbb{R}^n$ se, e somente se, para qualquer $v \neq 0 \in \mathbb{R}^n$ existe um elemento $d_i \in \{d_1, d_2, \ldots, d_r\}$ tal que $v^\top d_i > 0$*

Prova: Ver [17] ou [12], Teorema 2.3. ∎

É evidente, então, que dentro de um conjunto gerador positivo $\{d_1, d_2, \ldots, d_r\}$ deve existir ao menos um índice i tal que $-\nabla f(x_k)^\top d_i > 0$ e, portanto, uma direção de descida para f a partir de x_k.

A etapa de sondagem pode ser do tipo completa ou oportunística:

Sondagem completa: consiste em avaliar a função objetivo em todos os pontos de sondagem e escolher aquele que obtiver o maior decréscimo na função objetivo, satisfazendo, ainda, o critério de aceitação.

Sondagem oportunística: a etapa de sondagem é encerrada quando o ponto avaliado satisfaz o critério de aceitação pré-determinado. Optando por esse tipo de sondagem, pode-se ainda definir um esquema de ordenação que avalia primeiramente as direções mais propícias a serem direções de descida.

Neste trabalho optamos pelo uso da estratégia oportunística, já que, ao trabalharmos com problemas de grande porte, estratégias de busca completa têm uma tendência maior de se tornarem computacionalmente inviáveis. De qualquer maneira, a opção por uma ou outra estratégia não interfere na análise de convergência desta classe de algoritmos. Pode-se, inclusive, observar que o Algoritmo 4.1 é apto para abordar qualquer uma das duas escolhas.

4.1.1 Convergência para problemas de minimização irrestrita

Assim como nos demais métodos de otimização não-linear, a existência de uma direção de descida não é suficiente para garantir a convergência global do Algoritmo 4.1. No caso em que a função objetivo escolhida é continuamente diferenciável, é possível, desde que estabelecidas as condições corretas, obter resultados que garantam que uma sequência gerada irá convergir globalmente para um ponto estacionário da função objetivo.

A análise de convergência será realizada em duas etapas. Primeiro garantimos a existência de uma subsequência de tamanhos de passo que convirja para zero, o que é usual em demonstrações de algoritmos *derivative-free* e indica a existência de um ponto de acumulação. Após isso, será demonstrado que o algoritmo gera uma sequência de pontos que converge para um ponto crítico. A análise aqui realizada baseia-se naquela encontrada no capítulo 7 do livro de Conn, Scheinberg e Vicente [12] e as demonstrações dos resultados aqui expostos podem ser encontradas no mesmo trabalho, a menos que haja alguma indicação do contrário.

Para iniciar a análise, vamos impor que todos os pontos da sequência gerada pelo método pertençam a um conjunto compacto. Considerando que, neste caso, a sequência $\{f(x_k)\}$ é monotonamente não crescente[3] e f é contínua, basta impor a seguinte condição:

[3]Nesta seção, estamos tratando somente de métodos clássicos de busca direta, nos quais, em geral, utiliza-se critérios para aceitação de novos pontos que irão gerar uma sequência monótona. No nosso caso, como veremos na próxima

Hipótese 4.1 *O conjunto de nível*

$$L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\} \tag{4.3}$$

é compacto.

A partir dessa suposição, é possível trabalhar nas demais hipóteses que irão garantir a existência de ao menos uma subsequência de tamanhos de passo convergindo a zero, ou seja, que existe uma subsequência de índices $\{k_i\}$, tal que $\lim_{i\to\infty} \lambda_{k_i} = 0$. Essa propriedade está assegurada pelo Teorema 4.1, desde que, além da Hipótese 4.1, também seja garantido o cumprimento da Hipótese 4.2 a seguir.

Hipótese 4.2 *Se existe um valor $\lambda > 0$ tal que $\lambda_k > \lambda$, para todo $k \in \mathbb{N}$, então o Algoritmo 4.1 irá gerar um número finito de pontos.*

Para que fique esclarecida a importância da Hipótese 4.2, optamos por transferir para esse texto o Teorema 4.1, que por sua vez, engloba os resultados do Teorema 7.1 do livro [12] e, também, o Corolário 7.2.

Teorema 4.1 *Se as Hipóteses 4.1 e 4.2 são satisfeitas, então existem um ponto x^* e uma subsequência $\{k_i\}_{i=1}^{\infty}$ de iterações de insucessos tais que:*

$$\lim_{i\to\infty} \lambda_{k_i} = 0 \qquad e \qquad \lim_{i\to\infty} x_{k_i} = x^*. \tag{4.4}$$

Prova: Vamos demonstrar inicialmente que a sequência $\{\lambda_k\}$ é tal que:

$$\lim_{k\to+\infty} \inf \lambda_k = 0 \tag{4.5}$$

Assumindo por contradição que existe um valor $\lambda > 0$ tal que $\lambda_k > \lambda$ para todo k, então, pela Hipótese 4.2, o algoritmo irá visitar um número finito de pontos. Como o algoritmo se move para um novo ponto somente quando há sucesso na iteração, podemos concluir que existe um índice $\bar{k}$ tal que $x_k = x_{\bar{k}}$ para todo $k \geq \bar{k}$. Devido à maneira como o tamanho do passo λ_k é atualizado após cada iteração de insucesso, temos que

$$\lim_{k\to\infty} \lambda_k = 0, \tag{4.6}$$

o que contradiz a hipótese que assumimos e garante a veracidade da equação (4.5).

seção, vamos trabalhar com critérios que permitem a geração de sequências não necessariamente monótonas. Essa abordagem exigiu que realizássemos a demonstração de resultados específicos, o que pode ser conferido na subseção 4.3.

Assim sendo, deve existir uma subsequência de índices K_1 de iterações de fracassos tal que $\lim_{k\to\infty,\, k\in K_1} \lambda_{k+1} = 0$. Dado que $\lambda_k \leq (1/\tau_{\min})\lambda_{k+1}$, teremos que $\lim_{k\to\infty,\, k\in K_1} \lambda_k = 0$.

Considerando agora que, pela Hipótese 4.1, a sequência $\{x_k\}_{K_1}$ é limitada, então deve admitir uma subsequência convergente. Logo, existem um conjunto de índices $K_2 \subset K_1$ e um ponto $x^* \in \mathbb{R}^n$ tais que

$$\lim_{k\in K_2} x_k = x^*. \tag{4.7}$$

Como $K_2 \subset K_1$, a prova se completa colocando $\{k_i\}_{i=1}^{\infty} = K_2$. ■

Se por um lado o cumprimento da Hipótese 4.1 depende exclusivamente da função de mérito a ser considerada, é necessária uma melhor explanação sobre como a Hipótese 4.2 pode ser garantida na prática. Tradicionalmente, isso tem sido feito de duas maneiras:

1. através da imposição de uma condição de decréscimo suficiente, utilizando uma escolha adequada para a função $\varphi(.)$. Neste caso não há qualquer restrição sobre o número de bases positivas utilizadas;

2. por meio de condições que avaliem somente pontos pertencentes a uma malha racional previamente definida, ou seja, todo ponto avaliado pelo algoritmo deve pertencer ao conjunto:

$$M_k = \{x_k + \lambda_k D u : u \in \mathbb{N}^p\}, \tag{4.8}$$

   para algum $\lambda_k \in \mathbb{Q}$, onde p denota o número de elementos do conjunto de sondagem e D é uma matriz $n \times p$ e possibilita a determinação de diferentes conjuntos geradores positivos, se assim for desejado. A Figura 4.1 representa um exemplo de uma malha racional para uma base maximal. Ao utilizar esta estratégia, faz-se necessário que seja adotado um número finito de bases positivas. Porém, em contrapartida, o uso do simples decréscimo para aceitação de novos pontos é suficiente para garantir a convergência, e podemos trabalhar com a função forçante $\varphi \equiv 0$.

   No entanto, trabalhar com malhas não é a intenção deste trabalho, pois, como veremos adiante, este trabalho propõe, a cada iteração que se fizer necessário, construir um conjunto de direções ortogonais baseado na direção do resíduo espectral e, neste caso, não podemos garantir que seja possível a utilização de um número finito de bases. Sendo assim, não detalharemos como obter os resultados do Teorema 4.2 através dessa abordagem. O leitor interessado pode consultar o capítulo 7 do livro [12].

Para garantir que a Hipótese 4.2 seja satisfeita, são necessárias algumas condições sobre a função $\varphi(.)$ utilizada no critério de aceitação de pontos. Tais condições estão agrupadas na Hipótese 4.3.

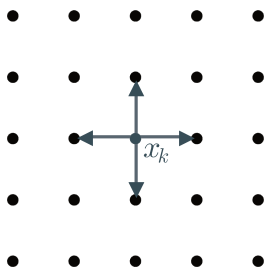

Figura 4.1: Malha racional para uma base maximal

Hipótese 4.3 *A função $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ utilizada no critério de aceitação de pontos pelo Algoritmo 4.1 é contínua, positiva, não decrescente e satisfaz*

$$\lim_{t \to 0^+} \frac{\varphi(t)}{t} = 0. \tag{4.9}$$

Para que fique claro como essa condição implica na satisfação da Hipótese 4.2, vamos reproduzir a demonstração do Teorema abaixo, originalmente encontrada em [12] (Teorema 7.11).

Teorema 4.2 *Se forem satisfeitas as Hipóteses 4.1 e 4.3, então a Hipótese 4.2 também será satisfeita.*

Prova: Lembremos que provar que a Hipótese 4.2 é satisfeita, implica demonstrar que se existe um valor $\lambda > 0$ tal que $\lambda_k > \lambda$ para todo $k \in \mathbb{N}$, então o Algoritmo 4.1 irá gerar um número finito de pontos. Sendo assim, dado que a função φ é monotonamente não decrescente, se $\lambda_k > \lambda$ para todo $k \in \mathbb{N}$, temos que $0 < \varphi(\lambda) \leq \varphi(\lambda_k)$. E, para provar que o algoritmo irá visitar um número finito de pontos, suponhamos, por absurdo, que existe uma subsequência infinita de iterações com sucesso. Da desigualdade $f(x_{k+1}) < f(x_k) - \varphi(\lambda_k)$ (Passo 2 do Algoritmo 4.1) tem-se que $f(x_{k+1}) < f(x_k) - \varphi(\lambda)$. Como, em casos de insucesso, $f(x_{k+1}) = f(x_k)$, a subsequência $\{f(x_k)\}$ deve convergir para $-\infty$ o que contradiz a Hipótese 4.1. ■

Já foi detalhada a possibilidade de assegurar o cumprimento da Hipótese 4.2, o que implica na garantia da existência de uma subsequência de tamanhos de passo convergindo para zero. A partir de agora vamos expor as condições para que a sequência gerada pelo Algoritmo 4.1 tenha como ponto limite um ponto crítico da função de mérito. Num primeiro momento, são impostas condições de Lipschitz continuidade sobre o gradiente de f e, também, condições sobre a geometria do conjunto de bases positivas $\mathcal{D}$.

Hipótese 4.4 *O gradiente ∇f é Lipschitz contínuo num conjunto aberto contendo o conjunto de nível $L(x_0)$, dado pela equação (4.3).*

Antes de descrever a condição sobre a geometria do conjunto $\mathcal{D}$, vamos definir a *medida do cosseno* de um conjunto gerador positivo, conforme proposto por Kolda, Lewis e Torczon [35].

Definição 4.3 *A medida do cosseno de um conjunto gerador positivo D é dada por:*

$$\kappa(D) = \min_{v \in \mathbb{R}^n,\, v \neq 0} \max_{d \in D} \frac{v^\top d}{\|v\| \|d\|}. \tag{4.10}$$

O Algoritmo 4.1 admite a utilização de infinitas bases positivas. Conforme [12, p.22], todo conjunto gerador positivo possui medida do cosseno positiva. No entanto, para demonstrar a existência de uma subsequência que converge para um ponto estacionário é necessário estabelecer uma condição sobre as medidas do cosseno desses conjuntos, a fim de que sejam suficientemente positivas. A demonstração do teorema também irá exigir que os vetores da base positiva tenham suas normas limitadas.

Hipótese 4.5 *Considere $\kappa_{\min} > 0$ e $\lambda_{\max} > 0$ constantes previamente definidas no início do algoritmo. Toda base positiva D_k usada no Algoritmo 4.1 é tal que $\kappa(D_k) > \kappa_{\min}$ e para todo vetor $d \in D_k$, tem-se $\|d\| \leq \lambda_{\max}$.*

Teorema 4.3 *Se forem satisfeitas as Hipóteses 4.1, 4.2, 4.4 e 4.5, então a sequência $\{x_k\}$ gerada pelo Algoritmo 4.1 admite um ponto limite x^* satisfazendo:*

$$\nabla f(x^*) = 0. \tag{4.11}$$

Prova: Ver [12], Teorema 7.3. ∎

Antes de prosseguir, convém listar possibilidades de relaxar algumas das hipóteses necessárias para a demonstração do Teorema 4.3. Conforme pode ser visto em [12], Teorema 7.4, ao utilizar um número finito de bases positivas, além de garantir que a Hipótese 4.5 seja satisfeita automaticamente, podemos substituir a Hipótese 4.4 por uma condição mais fraca, na qual f necessita ser somente continuamente diferenciável (e não necessariamente ter gradiente Lipschitz contínuo) em um aberto contendo o conjunto de nível $L(x_0)$.

Caso tenha sido utilizada a condição de decréscimo suficiente para satisfação da Hipótese 4.2, o trabalho de Kolda, Lewis e Torczon [35] apresenta ainda a possibilidade de demonstrar um resultado equivalente àquele obtido pelo Teorema 4.1, substituindo a Hipótese 4.1, que exige que a sequência gerada esteja dentro de um conjunto compacto, por uma hipótese mais fraca, que irá exigir somente que f seja limitada inferiormente no conjunto de nível dado por (4.3).

Como já adiantamos, nossa intenção é trabalhar com sistemas não lineares, por isso utilizaremos como função objetivo a função de mérito $f(x) = \|F(x)\|^2$ e, também, uma versão não monótona para

a busca linear. Além disso, desenvolvemos um novo conjunto de direções que aproveita os dados da direção do gradiente espectral. Sendo assim, foi necessária uma nova análise de convergência para suprir as modificações que propusemos. Começaremos a nossa análise na próxima seção, relatando a construção da nova base positiva [16].

4.2. Novas bases positivas

Seguindo a estrutura proposta em [12] e relatada no Algoritmo 4.1 para resolução de sistemas não lineares, desenvolvemos um novo método em que a fase de busca corresponde a uma versão finita do Algoritmo 3.4 (`HIB1`) e, para a fase de sondagem apresentaremos uma nova sequência de bases positivas, utilizando os pontos advindos da direção do resíduo espectral [16]. Essa abordagem segue uma das etapas do algoritmo ORTHOMADS [1], uma variante do Algoritmo MADS (*Mesh Adaptive Direct Search*) [4] que utiliza direções ortogonais no conjunto de sondagem.

O método ORTHOMADS é um método determinístico utilizado para resolução de problemas de minimização através de busca direta. ORTHOMADS gera um conjunto de direções de sondagem ortogonais, de forma que a união de todas as direções geradas (em todas as iterações), se normalizadas, gera um conjunto denso na esfera unitária. Esta característica é uma condição necessária para uma boa convergência de métodos de busca direta em problemas de minimização no contexto não-suave. Para gerar esse conjunto, cada iteração de ORTHOMADS realiza os seguintes procedimentos:

1. gera uma sequência de Halton pseudorrandômica [32] que produz um vetor no espaço $[0,1]^n$;
2. o vetor encontrado na etapa anterior é escalado para um tamanho apropriado e suas componentes são arredondadadas para um valor inteiro;
3. uma transformação tipo Householder escalada é aplicada, produzindo uma base B formada por n vetores ortogonais com entradas inteiras;
4. a união da base B e os correspondentes vetores simétricos dos elementos da base formam uma base positiva maximal, utilizada como o conjunto de direções de sondagem.

Estamos interessados em obter, após cada iteração de sucesso, e em caso de falha na etapa de busca, um conjunto D_k com $2n$ direções de sondagem, de forma que a direção do resíduo espectral e seu simétrico estejam contidos neste conjunto. A nossa intenção é gerar um conjunto com n direções ortogonais que contém $-(1/\alpha_k)F(x_k)$ ou $(1/\alpha_k)F(x_k)$ e, daí, determinar o conjunto D_k formado pelos vetores desse conjunto ortogonal e seus respectivos simétricos. Para isso usaremos a ideia de ortogonalização do algoritmo ORTHOMADS (item 3 da lista anterior) e iremos gerar o nosso próprio conjunto de direções de maneira que contenha as direções requeridas.

Não nos preocuparemos em arredondar nosso vetor para transformá-lo num vetor com entradas inteiras, já que pretendemos trabalhar com uma condição de decréscimo suficiente e, portanto, não temos essa necessidade. Também não precisamos nos preocupar com qualquer tipo de densidade na bola unitária, já que não estamos interessados na resolução de problemas não-suaves, e, sendo assim, não geraremos a sequência de Halton pseudorrandômica.

Relembremos que, dado um vetor q qualquer pertencente ao espaço $\mathbb{R}^n$, a transformação de Householder escalada definida por este vetor é dada por:

$$H = \|q\|^2(I - 2vv^\top), \tag{4.12}$$

onde I é a matriz identidade e $v = \|q\|^{-1}q$.

Observação 4.2: Note que a matriz H assim definida, embora tenha colunas/linhas ortogonais, não é uma matriz ortogonal, já que

$$H^\top H = \|q\|^4 I, \tag{4.13}$$

daí a denominação *Householder escalada*.

Dessa forma, para gerar nosso conjunto de direções, iremos determinar um vetor q adequado, de modo que uma das direções espectrais ($d_- = -(1/\alpha_k)F(x_k)$ ou $d_+ = (1/\alpha_k)F(x_k)$), cf. (2.13) e (2.14), seja uma coluna da matriz H. Ao definirmos a matriz H de acordo com a equação (4.12), suas colunas irão formar uma base ortogonal para o $\mathbb{R}^n$. Assim, aplicando o resultado da Proposição 4.1 no conjunto $\{e_1, e_2, \ldots, e_n, -e_1, -e_2, \ldots, -e_n\}$ temos que a união das colunas de H e $-H$ define uma base positiva para o $\mathbb{R}^n$.

Definindo $q = (q_1, q_2, \ldots, q_n)^\top$, conforme (4.12), a expressão geral de H em função de q será:

$$H = \begin{bmatrix} \|q\| - 2q_1^2 & -2q_1q_2 & \cdots & -2q_1q_i & \cdots & -2q_1q_n \\ -2q_1q_2 & \|q\| - 2q_2^2 & \cdots & -2q_2q_i & \cdots & -2q_2q_n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -2q_1q_i & -2q_2q_i & \cdots & \|q\| - 2q_i^2 & \cdots & -2q_iq_n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -2q_1q_n & -2q_2q_n & \cdots & -2q_iq_n & \cdots & \|q\| - 2q_n^2 \end{bmatrix}. \tag{4.14}$$

O vetor $d = (d_1, d_2, \ldots, d_n)^\top$ será escolhido como $-(1/\alpha_k)F(x_k)$ ou $(1/\alpha_k)F(x_k)$, e comporá a i-ésima coluna da matriz H, onde o índice i é tal que $F_i(x_k) \neq 0$. Dessa forma temos:

$$\begin{bmatrix} -2q_1q_i \\ -2q_2q_i \\ \vdots \\ \|q\| - 2q_i^2 \\ \vdots \\ -2q_iq_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_i \\ \vdots \\ d_n \end{bmatrix}. \tag{4.15}$$

Dado que serão utilizadas as colunas de ambas as matrizes H e $-H$ na definição do conjunto D_k, não importa se a direção d a ser considerada é $d = -(1/\alpha_k)F(x_k)$ ou $d = (1/\alpha_k)F(x_k)$. Assim sendo, podemos definir a direção d da seguinte maneira:

$$d = \begin{cases} -(1/\alpha_k)F(x_k), & \text{se } -(1/\alpha_k)F_i(x_k) > 0 \\ (1/\alpha_k)F(x_k), & \text{se } (1/\alpha_k)F_i(x_k) > 0. \end{cases} \tag{4.16}$$

Feito isso, trabalhamos com duas situações:

1. $d_i = \|d\|$, ou seja, $F_j(x_k) = 0$ para todo $j \neq i$;
2. $d_i \neq \|d\|$.

Caso ocorra a primeira situação, basta definir $H = d_iI$ que teremos uma matriz H, cujas colunas são ortogonais, de maneira que a i-ésima coluna seja a direção d como queríamos.

Para o segundo caso, devemos determinar as coordenadas do vetor q gerador da matriz de Householder H. Se definirmos inicialmente:

$$q_i = \sqrt{\frac{1}{2}(\|d\| - d_i)}, \tag{4.17}$$

e, após isso,

$$q_j = \frac{-d_j}{2q_i}, \text{ para todo } j \neq i, \tag{4.18}$$

teremos definido uma solução para o sistema linear (4.15) e, a partir daí, podemos determinar a matriz H e a base positiva que procurávamos.

4.3. Resolvendo sistemas não lineares de grande porte

Assim como foi feito no capítulo anterior, apresentamos um modelo conceitual para resolução de sistemas não lineares onde acrescentamos uma fase de busca direta quando o parâmetro λ_k se tornar demasiadamente pequeno na busca linear do Passo 9 do Algoritmo 3.4.

O novo modelo segue os mesmos procedimentos adotados no modelo descrito no Algoritmo 3.4 até o Passo 9. Neste passo, quando tivermos λ_k menor que um valor $\lambda_{\min}$ pré-determinado, o algoritmo irá iniciar uma fase de busca direta, utilizando o conjunto composto, a cada iteração, pelas colunas de H e $-H$, definidas conforme os procedimentos descritos na seção anterior.

Assim sendo, podemos interpretar que a fase de busca será composta pelos métodos DFSANE e Newton inexato. A fase de sondagem será acionada em situações em que o tamanho do passo λ_k se tornar demasiadamente pequeno, devido ao excesso de buscas lineares na direção fornecida pelo método de Newton inexato.

Observação 4.3: Cabe ressaltar que, na prática, a fase de busca direta somente será acionada quando estivermos em vias de declarar falha de execução por excesso de buscas lineares, já que o uso de métodos de busca direta não é recomendável para problemas de grande porte.

Diante da necessidade de diminuir o tamanho do passo na etapa de sondagem (iterações de insucesso), a próxima iteração não contará com a etapa de busca e, além disso, o conjunto de direções de sondagem utilizado será o mesmo. Em caso de sucesso, o algoritmo conta com a etapa de busca. Além disso, o nosso algoritmo também difere do algoritmo clássico de busca direta (Algoritmo 4.1) no critério de aceitação para novos pontos. No nosso caso, utilizamos um critério que possibilita a geração de uma sequência $\{f(x_k)\}$ não necessariamente monótona, assim como fizemos no capítulo anterior.

Como estamos interessados em trabalhar com problemas suaves de dimensões maiores do que usualmente se trabalha com algoritmos de busca direta, propomos uma nova estratégia de busca nas direções do conjunto de sondagem em que o tamanho do passo é diminuído à medida em que vamos mudando a direção do conjunto de sondagem. Ou seja, ao invés de testar todas as direções e, em caso de insucesso, diminuir o tamanho do passo e iniciar um novo ciclo de sondagem, propomos que o tamanho do passo seja reduzido a cada par de direções opostas testadas. Pretende-se, com isso, evitar um número excessivo de avaliações de função, já que o nosso objetivo é sair rapidamente da etapa de busca direta que, na prática, tende a ser bem mais cara do que as demais etapas. Chamamos essa nossa busca de *busca multidirecional* e a descrevemos no Algoritmo 4.3. A Figura 4.2 ilustra como seria a busca que estamos propondo para problemas em três dimensões, onde os pontos testados seriam somente aqueles que estão escurecidos (os número marcados indicam a ordem em que os pontos seriam testados em caso de sucessivos fracassos).

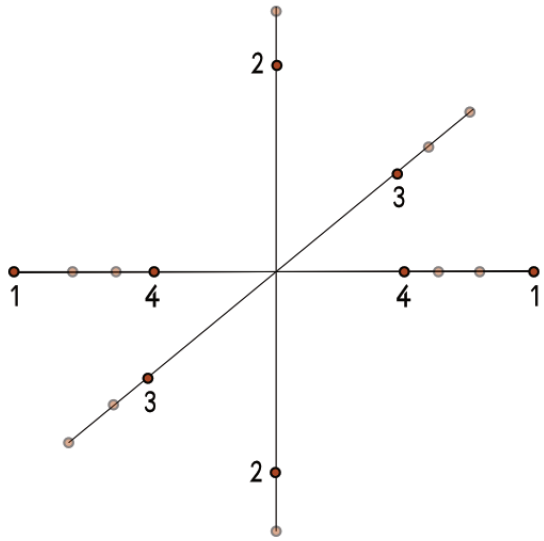

Figura 4.2: Busca multidirecional

Essas modificações que estamos propondo irão alterar a estrutura do Algoritmo 4.1 e, assim sendo, foi necessário realizar uma nova análise de convergência, que se encontra na subseção 4.3.

O Algoritmo 4.2 irá descrever o nosso segundo método híbrido, ao qual chamamos de **HIB2**. O fluxograma apresentado na Figura 4.3 auxilia a compreensão de cada etapa do método.

Algoritmo 4.2 ***HIB2***

Parâmetros de entrada: x_0, $\lambda_{\min}$, $NBL_{\max}$, $\beta \in (0,1)$, $0 < \tau_{\min} < \tau_{\max} < 1$ *e duas condições de aceitação para novos pontos:* $CA1$ *e* $CA2$.

1. *Defina* $k = 0$.
2. *Encontre a direção de busca* d_k *através do método DFSANE. Defina* $\lambda_+ = \lambda_- = 1$ *e* $NBL = 0$.
3. *Se* $x_+ = x_k + \lambda_+ d_k$ *ou* $x_- = x_k - \lambda_- d_k$ *satisfazem* $CA1$ *defina,* $\lambda_k = \lambda_+$, *no primeiro caso, ou* $\lambda_k = \lambda_-$, *no segundo, e vá para o passo 15.*
4. *Se* $NBL = NBL_{max}$ *vá para o passo 5. Caso contrário, escolha* $\lambda_+ \in [\tau_{\min}\lambda_+, \tau_{\max}\lambda_+]$ *e* $\lambda_- \in [\tau_{\min}\lambda_-, \tau_{\max}\lambda_-]$, *faça* $NBL = NBL + 1$ *e volte para o passo 3.*
5. *Determine* d_k *pelo método FDGMRES. Defina* $\lambda = 1$.
6. *Se* $x_k + \lambda d_k$ *satisfaz* $CA2$, *defina* $\lambda_k = \lambda$ *e vá para o passo 15.*
7. *Faça* $\lambda = 0.5\lambda$.

8. *Se $\lambda < \lambda_{\min}$, vá para o passo 9. Caso contrário, vá para o passo 6*

9. *Encontre $i \in \{1, 2, \ldots, n\}$ tal que $F_i(x_k) \neq 0$.*

   - *Se $-(1/\alpha_k)F_i(x_k) > 0$, faça $d = -(1/\alpha_k)F(x_k)$,*
   - *Caso contrário, faça $d = (1/\alpha_k)F(x_k)$.*

10. *Se $d_i = \|d\|$ então defina $H = d_i I$. Caso contrário:*

    (a) *Defina $q_i = \sqrt{\frac{1}{2}(\|d\| - d_i)}$;*

    (b) *Para $j = 1, 2, \ldots, n$ com $j \neq i$, defina $q_j = \frac{-d_j}{2q_i}$;*

    (c) *Calcule $v = \|q\|^{-1} q$;*

    (d) *Calcule $H = [h_1\, h_2 \ldots h_n] = \|q\|^2 (I - 2vv^\top)$;*

11. *Defina $r = mod(i, n) + 1$, $t = 0$ e $\lambda = 1$.*

12. *Obtenha λ utilizando o Algoritmo 4.3.*

13. *Se $\lambda = 0$, faça $\mu = \beta\mu$, $t = t + 1$ e volte para o passo 12.*

14. *Faça $\lambda_k = \lambda$.*

15. *Defina $x_{k+1} = x_k + \lambda_k d_k$ e faça $k = k + 1$.*

16. *Se $F(x_{k+1}) = 0$ ou $\nabla f(x_k) = 0$, pare o algoritmo e declare SUCESSO. Caso contrário, vá para o passo 2.*

Observação 4.4: Na prática, não é preciso encontrar e armazenar a matriz H. Se tivermos $d_i \neq \|d\|$, então, a cada teste de sondagem, basta determinar a coluna que será utilizada como direção de busca, conforme a matriz (4.14) e as equações (4.17) e (4.18). Se estivermos na situação em que $d_i = \|d\|$, é suficiente que o algoritmo faça, quando necessário, $h_r = d_i e_r$.

Apresentamos a seguir o algoritmo que será utilizado para realizar a busca multidirecional no Passo 12. Neste algoritmo tivemos de adicionar o marcador r_{marc} que indica o índice do vetor do conjunto de sondagem em que o algoritmo atinge um tamanho de passo menor que um valor $a\mu$ predeterminado. Neste caso, o algoritmo não reduz mais o valor do tamanho do passo e irá realizar, quando necessário, um novo ciclo completo. Esse procedimento é necessário para demonstrar a convergência teórica do método. Na prática, quando o algoritmo atingir um valor menor que $a\mu$ (que deve ser baixo), irá declarar falha de execução.

Algoritmo 4.3 *Busca multidirecional*

Parâmetros: $\gamma \in (0,1)$, $a \in (0,1]$, $0 < \xi_{\min} < \xi_{\max} < 1$ *e* $\mu \in (0,1)$.

1. $\alpha = a$
2. *Enquanto* $f(x_k + \alpha h_r) > (1 - \gamma\alpha^2)W_k$ *e* $f(x_k - \alpha h_r) > (1 - \gamma\alpha^2)W_k$, *faça:*
   (a) *Se* $\alpha \geq \mu a$ *escolha* $\xi \in [\xi_{\min}, \xi_{\max}]$, *faça* $\alpha = \xi\alpha$, $r_{marc} = r$ *e vá para 2c.*
   (b) *Caso contrário:*
      i. *se* $r \neq r_{marc}$, *vá para 2c ;*
      ii. *se* $r = r_{marc}$, *faça* $\lambda = 0$, $d_k = h_r$, $\delta = \alpha$ *e termine.*
   (c) *Faça* $r = mod(r, n) + 1$
3. *Se* $f(x_k + \alpha h_r) \leq (1 - \gamma\alpha^2)W_k$, *faça* $d_k = h_r$. *Caso contrário, faça* $d_k = -h_r$.
4. *Faça* $\lambda = \alpha$ *e termine.*

O critério de aceitação para novos pontos utilizado na nova busca direta, embora seja não monótono, difere daquele utilizado nas etapas anteriores. Para demonstração de um resultado de convergência global, houve a necessidade de um resultado equivalente àquele da Proposição 3.3, já com a mudança de busca linear para busca multidirecional. A demonstração será omitida aqui pois segue os mesmos procedimentos adotados na demonstração para a Proposição 3.3.

Proposição 4.3 *Suponha que* $F(x_k) \neq 0$ *e considere um valor* $\mu \in (0,1)$ *dado em* x_k. *Então o Algoritmo 4.3 determina, em um número finito de iterações, uma direção* d_k *e um escalar* $\lambda \in [0, a]$ *tais que:*

$$f(x_k + \lambda d_k) \leq (1 - \gamma\lambda^2)W_k. \tag{4.19}$$

Além disso, uma das seguintes condições é assegurada:

1. $\lambda = 0$ *e*
$$\|F(x_k + \delta_s h_s)\|^2 > (1 - \gamma\delta_s^2)W_k \geq (1 - \gamma\delta_s^2)\|F(x_k)\|^2, \tag{4.20}$$
   para $s = 1, 2, \ldots, 2n$ *com* $\delta_s < \mu a$; *ou*
2. $\lambda \geq \xi_{\min}\mu a$.

4.3.1 Análise de convergência

Analisamos nesta subseção a convergência do Algoritmo 4.2. A demonstração segue a linha da demonstração para a Proposição 6.1, apresentada em [30]. Tal qual aquele resultado, provamos que a sequência gerada ou obtém um ponto crítico da função de mérito ou admite uma subsequência que converge para um ponto crítico da função de mérito. Assim como foi feito em [30], consideramos que x é um ponto crítico para a função de mérito $f(x) = \|F(x)\|^2$ se satisfaz uma das condições:

- $\|F(x)\|^2 = 0$
- $\|F(x)\|^2 > 0$ e $\nabla f(x) = 2J(x)^\top F(x) = 0$.

Teorema 4.4 *Considere $\{x_k\}$ a sequência gerada pelo Algoritmo 4.2 HIB2 em que as condições $CA1$ e $CA2$ são ambas definidas como:*

$$f(x_k + \lambda_k d_k) \leq W_k + \zeta_k - \gamma\lambda_k^2 f(x_k) \tag{4.21}$$

onde $\zeta_k > 0$ para todo k, $\sum_{k=1}^{\infty}\zeta_k = \zeta < \infty$ e, além disso satisfaz:

$$\zeta_k - \bar{c}f(x_k) < 0 \text{ com } \bar{c} > 0 \tag{4.22}$$

para todo k maior ou igual a algum $\bar{k} \in \mathbb{N}$. Assuma que o conjunto de nível $\bar{\mathcal{L}}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0) + \zeta\}$ é limitado. Então o algoritmo está bem definido e, ou termina em um ponto x_k que é ponto crítico para a função de mérito, ou gera uma sequência infinita tal que todo ponto limite de $\{x_k\}$ é um ponto crítico da função de mérito.

Prova: Todo ponto x_k gerado pelo algoritmo irá satisfazer:

$$f(x_{k+1}) \leq W_k + \zeta_k, \text{ para todo } k. \tag{4.23}$$

Assim sendo, pela Proposição 3.1, vamos ter que $x_k \in \bar{\mathcal{L}}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0) + \zeta\}$ para todo k.

Vamos provar inicialmente que o Algoritmo 4.2, HIB2, está bem definido. Para isso basta provar que não irá ciclar entre os passos 12 e 13. Suponha, por absurdo, que existe um índice k de maneira que não é possível obter um ponto x_{k+1} pelo algoritmo HIB2. Temos que $F(x_k) \neq 0$ e, além disso, $\nabla f(x_k) \neq 0$, pois, caso contrário, o algoritmo teria parado.

Considerando que h_i é a i-ésima coluna da matriz H, vamos definir $h_{n+i} = -h_i$ para todo $i = 1, 2, \ldots, n$. O algoritmo deve gerar todos os vetores $h_1, \ldots, h_n, h_{n+1}, \ldots, h_{2n}$ da base positiva

maximal, já que o número de chamadas de busca multidirecional t converge para o infinito. E, para cada $s \in \{1, \ldots, 2n\}$ e para cada valor de t, a busca deve terminar sem sucesso, de forma que, pela Proposição 4.3, podemos encontrar escalares $\delta_s(t)$ tais que:

$$\|F(x_k + \delta_s(t)h_s)\|^2 > (1 - \gamma\delta_s(t)^2)\|F(x_k)\|^2, \text{ com } 0 < \delta_s(t) < \mu(t)a \tag{4.24}$$

onde $\mu(t) = \beta^t\mu_0$. Quando $t \to \infty$, tem-se $\mu(t) \to 0$ e, consequentemente, $\delta_s(t) \to 0$, para cada s. Dessa maneira, temos que para valores suficientemente grandes de t o ponto $x_k+\delta_s(t)h_s$ se encontra numa vizinhança de x_k em que a função f é diferenciável. Assim, pelo Teorema do Valor Médio:

$$f(x_k + \delta_s(t)h_s) = f(x_k) + \delta_s(t)\nabla f(x_k + b_s(t)\delta_s(t)h_s)^\top h_s \tag{4.25}$$

para algum valor $b_s(t) \in (0, 1)$. Pela desigualdade (4.24), tem-se, portanto:

$$f(x_k) + \delta_s(t)\nabla f(x_k + b_s(t)\delta_s(t)h_s)^\top h_s > f(x_k) - \gamma\delta_s(t)^2 f(x_k). \tag{4.26}$$

Assim, para valores suficientemente grandes de t temos

$$\nabla f(x_k)^\top h_i \geq 0,\ i = 1, 2, \ldots, 2n \tag{4.27}$$

Agora, pela definição de base positiva tem-se que existem valores $\beta_i^k \geq 0$, $i = 1, \ldots, 2n$ tais que:

$$-\nabla f(x_k) = \sum_{i=1}^{2n} \beta_i^k h_i. \tag{4.28}$$

Consequentemente,

$$-\|\nabla f(x_k)\|^2 = \sum_{i=1}^{2n} \beta_i^k \nabla f(x_k)^\top h_i \geq 0 \tag{4.29}$$

o que implica que $\nabla f(x_k) = 0$ e contradiz a hipótese de que $\nabla f(x_k) \neq 0$. Logo, o algoritmo não irá ciclar e temos assegurado que um novo ponto x_{k+1} será obtido utilizando o valor para λ_k obtido no Passo 3, 6 ou 12.

Para provar que todo ponto limite é ponto crítico, vamos supor, por absurdo, que existe um conjunto infinito de índices $K \subset \mathbb{N}$ tal que a subsequência $\{x_k\}_{k\in K}$ converge para um ponto $\bar{x}$ que não é um ponto crítico. Neste caso, para valores suficientemente grandes de k temos

$$\|F(x_k)\| > \varepsilon, \tag{4.30}$$

para algum $\varepsilon > 0$ e $\nabla f(\bar{x}) \neq 0$. Pela Proposição 3.1, temos que a sequência $\{W_k\}_{k\in K}$ vem a tornar-se monotonamente não crescente para valores grandes de k e, portanto, admite um limite

$W^* \geq 0$. Se $W^* = 0$ há uma contradição com (4.30) e, portanto, W^* deve ser estritamente positivo.

Vamos agora averiguar que é possível obter um valor $c \in (0,1)$ de forma que

$$\|F(x_{k+1})\|^2 \leq W_k + \zeta_k - c\|F(x_k)\|^2. \tag{4.31}$$

Já provamos que o algoritmo está bem definido, desta maneira, para todo k, um ponto x_{k+1} será gerado pelo algoritmo HIB2, a partir de x_k e deve necessariamente ter uma das características abaixo:

(a) x_{k+1} é obtido através da utilização do método DFSANE no Passo 3 e, neste caso, a condição (3.33) é satisfeita com um tamanho de passo $\lambda_k \geq \tau_{\min}^{NBL_{\max}}$ e podemos escrever:

$$f(x_{k+1}) \leq W_k + \zeta_k - \gamma(\tau_{\min}^{NBL_{\max}})^2 f(x_k); \tag{4.32}$$

(b) x_{k+1} é obtido através do método Newton-FDGMRES no Passo 6. Neste caso, deve haver um valor $\lambda_k \geq \lambda_{\min}$ que satisfaz a condição (3.33) e, assim,

$$f(x_{k+1}) \leq W_k + \zeta_k - \gamma\lambda_{\min}^2 f(x_k); \tag{4.33}$$

(c) x_{k+1} é obtido através do método de busca direta no Passo 12.

Sendo assim, a única possibilidade de que a sequência gerada pelo algoritmo não garanta a existência de um valor para c que assegure a desigualdade (4.31) reside nos pontos do caso (c). Definindo t_k como o número de chamadas consecutivas do Algoritmo de busca multidirecional 4.3, a partir de x_k, temos duas possibilidades:

(c_1) existe um valor $\bar{t}$ tal que $t_k \leq \bar{t}$ para todo $k \in K$;

(c_2) existe uma subsequência infinita $\{x_k\}_{K_1}$, com $K_1 \subset K$ tal que todo ponto x_k satisfazendo $k \in K_1$ é obtido no passo 12 e, além disso,

$$\lim_{k\to\infty,\, k\in K_1} t_k = \infty. \tag{4.34}$$

Vamos provar que $\{t_k\}_{k\in K}$ é limitada superiormente e que, consequentemente, (c_2) não irá ocorrer.

A Proposição 4.3 garante que, para cada h_s, podemos encontrar δ_s^k

$$\|F(x_k + \delta_s^k h_s)\|^2 > W_k - \gamma(\delta_s^k)^2\|F(x_k)\|^2 \geq (1 - \gamma(\delta_s^k)^2)\|F(x_k)\|^2. \tag{4.35}$$

Assim sendo, utilizando os mesmos argumentos da prova de que o algoritmo está bem definido (ver equações (4.25), (4.26), (4.27), (4.28)) vamos ter que:

$$f(x_k + \delta_s^k h_s) = f(x_k) + \delta_s^k \nabla f(x_k + b_s(k)\delta_s^k h_s)^\top h_s \tag{4.36}$$

para algum valor $b_s(k) \in (0,1)$. E, assim:

$$f(x_k) + \delta_s^k \nabla f(x_k + b_s(k)\delta_s^k h_s)^\top h_s > f(x_k) - \gamma(\delta_s^k)^2 f(x_k). \tag{4.37}$$

Tomando o limite para $k \to \infty$, $k \in K_1$ em (4.37):

$$\nabla f(\bar{x})^\top h_s \geq 0,\ s = 1, 2, \ldots, 2n, \tag{4.38}$$

e, dado que, pela definição de base positiva,

$$-\nabla f(\bar{x}) = \sum_{i=1}^{2n} \beta_i^k h_i,\ \text{com}\ \beta_i^k \geq 0,\ i = 1, 2, \ldots, 2n \tag{4.39}$$

vamos ter

$$-\|\nabla f(\bar{x})\|^2 = \sum_{i=1}^{2n} \beta_i^k \nabla f(\bar{x})^\top h_i \geq 0, \tag{4.40}$$

o que implica que $\nabla f(\bar{x}) = 0$ e contradiz nossa hipótese de que $\bar{x}$ não é um ponto estacionário. Sendo assim, a subsequência $\{t_k\}_{k \in K_1}$ deve ser limitada superiormente e, consequentemente, o caso (c_2) nunca irá ocorrer, donde todo ponto obtido pelo passo de busca direta é do caso (c_1).

Logo, se x_k é obtido pelo passo de busca direta, pelas instruções do Algoritmo HIB2 deve valer:

$$f(x_{k+1}) \leq (1 - \gamma\bar{\lambda}^2) W_k, \tag{4.41}$$

onde $\bar{\lambda} = a\xi_{\min}^{\bar{t}}$. E, portanto,

$$f(x_{k+1}) \leq W_k + \zeta_k - \gamma\bar{\lambda}^2 f(x_k). \tag{4.42}$$

Definindo agora $c = \min\{\tau_{\min}^{NBL_{\max}}, \lambda_{\min}, \bar{\lambda}\}$, concluímos, por (4.32), (4.33) e (4.42), que a condição (4.31) é satisfeita para todo valor de $k \in K$. Assim sendo, por (4.22) e pela desigualdade (4.42), as hipóteses da Proposição 3.2 estão satisfeitas para $k \geq \bar{k}$ e, consequentemente,

$$\lim_{k \to \infty,\, k \in K} f(x_{k+1}) = \lim_{k \to \infty,\, k \in K} W_k = 0, \tag{4.43}$$

contradizendo a hipótese de que $W^* > 0$ e completando nossa demonstração. ■

Para avaliar o impacto da adição da etapa de busca direta na robustez dos algoritmos, adicionamos esta nova etapa aos algoritmos `DFSANE`, `H1E` e `H6E` definidos o capítulo anterior, criando, assim, três novos algoritmos, que foram denominados, respectivamente, `DFSANE-BD`, `HBD1E` e `HBD6E`.

A resolução de problemas através de algoritmos com estratégias de busca direta tende a ficar exponencialmente mais cara à medida que a dimensão do problema aumentar e, sendo assim, não é recomendável para problemas de grande porte. Por isso, conforme já adiantamos, a fase de busca direta só será acionada em situações em que o Algoritmo 3.4 estiver em vias de declarar falhas de execução. Além disso, nos concentramos na resolução de problemas com dimensão $n = 100$ e os valores iniciais propostos na literatura.

Os problemas de La Cruz e Raydan [14] já haviam sido testados com esta dimensão no capítulo anterior. Para esse novo teste, foram selecionados os problemas em que qualquer um entre os algoritmos `DFSANE`, `H1E` e `H6E` tenha declarado falha de execução ou por estagnação na busca linear ou por excesso de iterações internas, o que não permitiu que uma direção fosse encontrada para prosseguir com o método Newton inexato. Problemas em que a falha de execução se deu por excesso de avaliações de função não foram considerados a princípio, já que a simples adição da fase de busca direta não irá proporcionar qualquer melhoria na robustez. Esse tipo de falha será tratada posteriormente. Sendo assim, os problemas 13, 14, 17, 18 e 19 foram avaliados neste teste.

No caso dos problemas encontrados em [38], ainda não havíamos realizado testes com dimensão $n = 100$. Sendo assim, testamos os problemas 1 a 21 para esse conjunto de testes com a dimensão adequada e, a partir daí, agrupamos num novo conjunto de testes, todos e aqueles problemas que resultaram nas falhas dos tipos já indicado no parágrafo anterior. Esse novo conjunto conta com os problemas 1, 2, 5, 6, 10, 19 e 20 para análise dos novos algoritmos.

Ao avaliar os resultados para posterior escolha dos problemas testes, foi possível constatar que sempre que o Algoritmo `DFSANE` falhava, o motivo era excesso de avaliações de função. Isso impossibilitaria o teste do Algoritmo `DFSANE-BD`, já que a estratégia de busca direta nunca seria acionada. Foi feita uma investigação sobre esse fenômeno, da qual falaremos mais adiante.

As duas primeiras fases dos algoritmos `HBD1E` e `HBD6E` e a primeira fase do algoritmo `DFSANE-BD` repetem os parâmetros utilizados nos testes computacionais que realizamos no Capítulo 3. Neste sentido, ressaltamos que a fase de busca direta só é acionada quando o tamanho do passo estiver menor que 10^{-12} (ou seja, quando houver estagnação) ou após 30 ciclos de 30 iterações internas do método FDGMRES, quando uma direção que satisfaça a condição Newton inexata não é encontrada.

Na fase de busca direta, a condição para aceitação de novos pontos, para condizer com o resultado do Teorema 4.4, ao invés de utilizar o critério (2.21) proposto por La Cruz, Martínez e Raydan, vamos utilizar o critério (4.19), como também ocorre em [30]. No caso, foram mantidos $\gamma = 10^{-4}$ e $M = 7$. Para realizar a busca direta multidirecional, foram escolhidos os parâmetros $\xi = 0.95$, $a = 1$ e, assim como foi feito no algoritmo anterior, declaramos falha de execução quando o tamanho do passo atinge um valor inferior a 10^{-10}. Sendo assim, definimos $\mu = 10^{-10}$ e, determinamos que o Algoritmo 4.2 irá parar sempre que, no Passo 12 do Algoritmo 4.2, a busca multidirecional retornar o valor $\lambda = 0$. Os testes foram realizados utilizando o *software* Matlab 7.0, em uma máquina com processador Intel(R) Core I3-2100 3.10GHz e 4Gb de memória RAM.

O maior destaque neste conjunto de testes foi a resolução do problema 19 de [38] pelo Algoritmo `HBD1E` o que não ocorria com o algoritmo sem o uso da busca direta. Além disso, nos demais problemas, embora não tenha havido convergência, o valor da norma do resíduo, $\|F(x_k)\|$, no último ponto encontrado por esse algoritmo diminui com relação ao último ponto encontrado pelo Algoritmo `H1E`. A Tabela 4.3 compara os valores finais dos algoritmos sem o uso de busca direta com aqueles que a usam. Destacamos nesse conjunto de testes, as reduções obtidas no Problema 1 para ambos os algoritmos e as reduções nos Problemas 2 e 10 para o Algoritmo `HBD1E`. Podemos concluir ainda que há um impacto muito maior na adoção da estratégia pelo método `H1E` do que pelo método `H6E`.

	`H6E`			`HBD6E`		
	Final	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
13	EII	209	1218	EAVF	1121	10001
14	EII	150	824	EAVF	1069	10001
17	EAVF	1945	10001	EAVF	1945	10001
18	C	189	1166	C	189	1166
19	EST	183	1403	EAVF	1154	10001
	`H1E`			`HBD1E`		
	Final	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
13	EII	4	40	EAVF	248	10001
14	EII	5	40	EAVF	241	10001
17	EII	2	36	EST	25	2102
18	C	14	59	C	14	59
19	C	4	16	C	4	16

Tabela 4.1: Resultados para o teste com algoritmos acoplados com busca direta - problemas de [14].

Após esses testes e com a intenção de detectar uma possível causa para o fato de que sempre que o Algoritmo `DFSANE` fracassa, a falha é ocasionada por excesso de avaliações de função e nunca por estagnação no tamanho do passo, avaliamos detalhadamente o desempenho do algoritmo nos testes envolvendo os problemas de [38]. Foi possível perceber que a não monotonocidade do método criava uma grande facilidade para que novos pontos fossem aceitos, de maneira que poucas reduções

	H6E			HBD6E		
	Final	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
1	EII	737	5958	EAVF	983	10001
2	EII	786	5568	EAVF	1084	10001
5	C	285	1092	C	285	1092
6	EST	355	2799	EAVF	907	10001
10	EII	1019	7641	EAVF	1087	10001
19	C	1065	2706	C	1065	2706
20	EII	159	1324	EAVF	723	10001

	H1E			HBD1E		
	Final	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
1	EII	4	110	EAVF	225	10001
2	EII	1	48	EAVF	184	10001
5	EII	0	34	EAVF	87	10001
6	C	6	41	C	6	41
10	EII	1	35	EAVF	190	10001
19	EII	2	36	C	188	4084
20	EII	0	34	EAVF	89	10001

Tabela 4.2: Resultados para o teste com algoritmos acoplados com busca direta - problemas de [38].

P	H6E	HBD6E	H1E	HBD1E
1	5.6598	5.3229	2.7197	0.0506
2	0.7600	0.7676	1.5422	0.4792
5	-	-	27.8880	27.6237
6	32.0268	46.1424	-	-
10	1.0804	1.0814	14.9917	4.4618
20	452.0327	444.2314	459.6194	458.0485

Tabela 4.3: Norma de F no último ponto encontrado pelo método.

no tamanho do passo eram suficientes para a aceitação de um novo ponto pela direção do resíduo espectral. Esse fato pode ser observado na Tabela 4.4. Tal tabela indica, para cada problema, em quantas iterações o Algoritmo `DFSANE` necessita de nr reduções no tamanho do passo para aceitação de um novo ponto. Como podemos observar, em nenhum problema foram necessárias mais do que 8 reduções no tamanho do passo em qualquer iteração para que um novo ponto fosse aceito, o que acaba por justificar a constante falha por excesso de iterações e nunca por excesso de reduções no tamanho do passo.

	Problemas						
nr	**1**	**2**	**5**	**6**	**10**	**19**	**20**
1	36	40	90	25	35	356	0
2	102	75	119	18	52	153	34
3	174	497	25	125	632	50	55
4	675	570	0	565	495	2	50
5	153	57	0	295	40	0	333
6	7	1	0	40	1	0	331
7	0	1	0	3	0	0	46
8	0	0	0	0	0	0	5

Tabela 4.4: Número de iterações em que o Algoritmo `DFSANE` necessitou realizar nr iterações para aceitar um novo passo através da direção espectral.

Feita a detecção do problema e no sentido de possibilitar o teste do Algoritmo `DFSANE-BD`, modificamos o Algoritmo `DFSANE` de maneira que além de interrupção desta fase pela detecção da estagnação, a fase DFSANE também fosse interrompida quando ocorresse falta de progresso no valor da função de mérito. Agora o Algoritmo também irá declarar falha nesta fase sempre que

$$\max_{i=k,k-1,\ldots,k-10} \|F(x_i) - F(x_{i-1})\|_\infty < \varepsilon, \tag{4.44}$$

onde ε é um número real positivo, próximo de zero.

No entanto, fomos surpreendidos com o fato de que a flexibilidade do critério de aceitação fazia com que, em determinados momentos, houvesse um aumento consideravelmente grande no valor da função de mérito, provocando um efeito do tipo gangorra, o que impedia, ao mesmo tempo, o avanço para valores relativamente baixos da função de mérito e qualquer tipo de estagnação no tamanho do passo e/ou falta de progresso. Exemplificando, ao testar com $\varepsilon = 10^{-6}$ e $\varepsilon = 10^{-5}$ em momento algum foi detectada estagnação e/ou falta de progresso, com $\varepsilon = 10^{-4}$ e $\varepsilon = 10^{-3}$ foram detectadas faltas de progresso somente no problema 19.

No critério de aceitação utilizado na fase DFSANE há duas componentes que contribuem para que a busca não seja monótona : $\zeta_k > 0$ e o valor de M, que possibilita a comparação com o maior valor da função de mérito nas últimas iterações (e não propriamente com $f(x_k)$).

Se a dificuldade fosse causada por um valor muito grande para M, poderia ser observada simplesmente ao analisar o valor da função de mérito em cada iteração e detectar o efeito gangorra em blocos de, no máximo, M iterações, o que não ocorreu. Assim sendo, elegemos como principal causador desse efeito gangorra, o termo ζ_k da condição (2.21). Para remediar, implementamos novos testes, modificando o expoente do denominador na equação (3.55) de 1.1 para 2. Após essa modificação e utilizando $\varepsilon = 10^{-3}$, foram detectadas estagnações nos problemas $2, 6, 10, 19$.

Durante esses testes, foi possível observar que a busca direta realiza menos iterações do que esperávamos para encontrar um ponto satisfatório. No entanto, quando necessita de muitas iterações (e portanto, encontra um novo ponto através de um passo já pequeno) é comum a recorrência da necessidade de utilizar a busca direta com muitas reduções nas próximas iterações até que num momento essa necessidade desaparece. Uma possível causa para esse efeito é que os pontos encontrados estejam localizados em uma região em que o método DFSANE não tem bom desempenho. Para melhorar o desempenho do algoritmo frente a essa situação, adicionamos uma estratégia de extrapolação para a fase de busca direta, isto é, em caso de sucesso na busca direta, avalia-se o ponto na mesma direção, mas com o dobro do tamanho do passo e assim sucessivamente até encontrar um ponto que não seja satisfatório, adotando, neste caso, o penúltimo ponto testado. Pretende-se com isso dar passos maiores e fugir de regiões em que o método DFSANE não possui bom desempenho.

Os resultados podem ser encontrados na Tabela 4.5, onde os algoritmos que contam com a estratégia de extrapolação estão grafados com o termo "(EXTRAP)" em frente ao nome. Para esse teste, tanto os Algoritmos `DFSANE-BD`, `HBD1E` e `HBD6E`, quanto suas respectivas versões com extrapolação, contam com o critério (4.44) para detectar falta de progresso na fase DFSANE. Ademais, todos os algoritmos testados para obtenção dos dados dessa tabela utilizaram o valor 2 no expoente do denominador sequência (3.55).

Assim como no teste anterior, o algoritmo que mais se beneficiou das mudanças foi o algoritmo que não aceita redução no tamanho do passo resíduo espectral, `H1E`. Além do problema 19 que já havia sido resolvido pela versão sem extrapolação, o Algoritmo `HBD1E` com extrapolação também resolveu o problema 20. No que podemos concluir que a adoção dessas mudanças foram benéficas, no sentido de diminuir o número de problemas não resolvidos, o que era o objetivo deste capítulo.

Os demais algoritmos, embora tenham conquistado alguma redução no valor da função de mérito no critério de parada em alguns dos problemas resolvidos, não foram beneficiados de forma tão impactante. Como o algoritmo `H1E` é, dentre os algoritmos híbridos desenvolvidos no Capítulo 3, aquele que tende a utilizar mais vezes o método de Newton, nossos testes indicam que o uso de direções alternativas pode proporcionar um desempenho melhor em termos de robustez para o método de Newton inexato. Esse fato já era esperado, visto que as hipóteses necessárias para a demonstração de algoritmos tipo Newton e Newton inexato sem busca direta são mais fortes que as Hipóteses 2.1, 2.2

	DFSANE			DFSANE-BD			DFSANE-BD(EXTRAP)		
P	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
1	EAVF	1155	10002	EAVF	1155	10002	EAVF	977	10002
2	EAVF	1282	10001	EAVF	1282	10001	EAVF	1036	10001
5	C	285	1092	C	285	1092	C	285	1092
6	EAVF	1078	10001	EAVF	1078	10001	EAVF	890	10003
10	EAVF	1287	10001	EAVF	1287	10001	EAVF	1074	10001
19	C	1065	2706	C	1065	2706	C	1065	2706
20	EAVF	853	10001	EAVF	853	10001	EAVF	779	10001
	H6E			HBD6E			HBD6E(EXTRAP)		
P	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
1	EII	737	5958	EAVF	1192	10002	EAVF	978	10001
2	EII	786	5568	EAVF	1239	10001	EAVF	1083	10001
5	C	285	1092	C	285	1092	C	285	1092
6	EST	355	2799	EAVF	1028	10001	EAVF	906	10001
10	EII	1019	7641	EAVF	1250	10001	EAVF	1083	10001
19	C	1065	2706	C	1065	2706	C	1065	2706
20	EII	159	1324	EAVF	934	10001	EAVF	723	10001
	H1E			HBD1E			HBD1E(EXTRAP)		
P	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**	**Final**	**Iter**	**AvalF**
1	EII	4	110	EAVF	193	10001	EAVF	465	10001
2	EII	1	48	EAVF	176	10001	EST	188	7326
5	EII	0	34	EAVF	49	10001	PPF	2	213
6	C	6	41	C	6	41	C	6	41
10	EII	1	35	EAVF	190	10001	EAVF	583	10001
19	EII	2	36	C	188	4084	C	188	4115
20	EII	4	34	EAVF	89	10001	C	24	591

Tabela 4.5: Algoritmos de busca direta com extrapolação - problemas de [38]

e, 2.3, necessárias para a demonstração de algoritmos tipo resíduo espectral, conforme Teorema 2.2. E de acordo com o Teorema 3.4, mesmo o algoritmo híbrido desenvolvido no capítulo 3 possui hipóteses mais fortes que o algoritmo `DFSANE` e, ao adotar a busca direta, as hipóteses necessárias para obtenção dos resultados de convergência para os algoritmos híbridos com a adição desta fase são mais fracas do que aquelas necessárias para os algoritmos sem essa fase, conforme podemos observar ao compararmos os Teoremas 4.4 e 3.4.

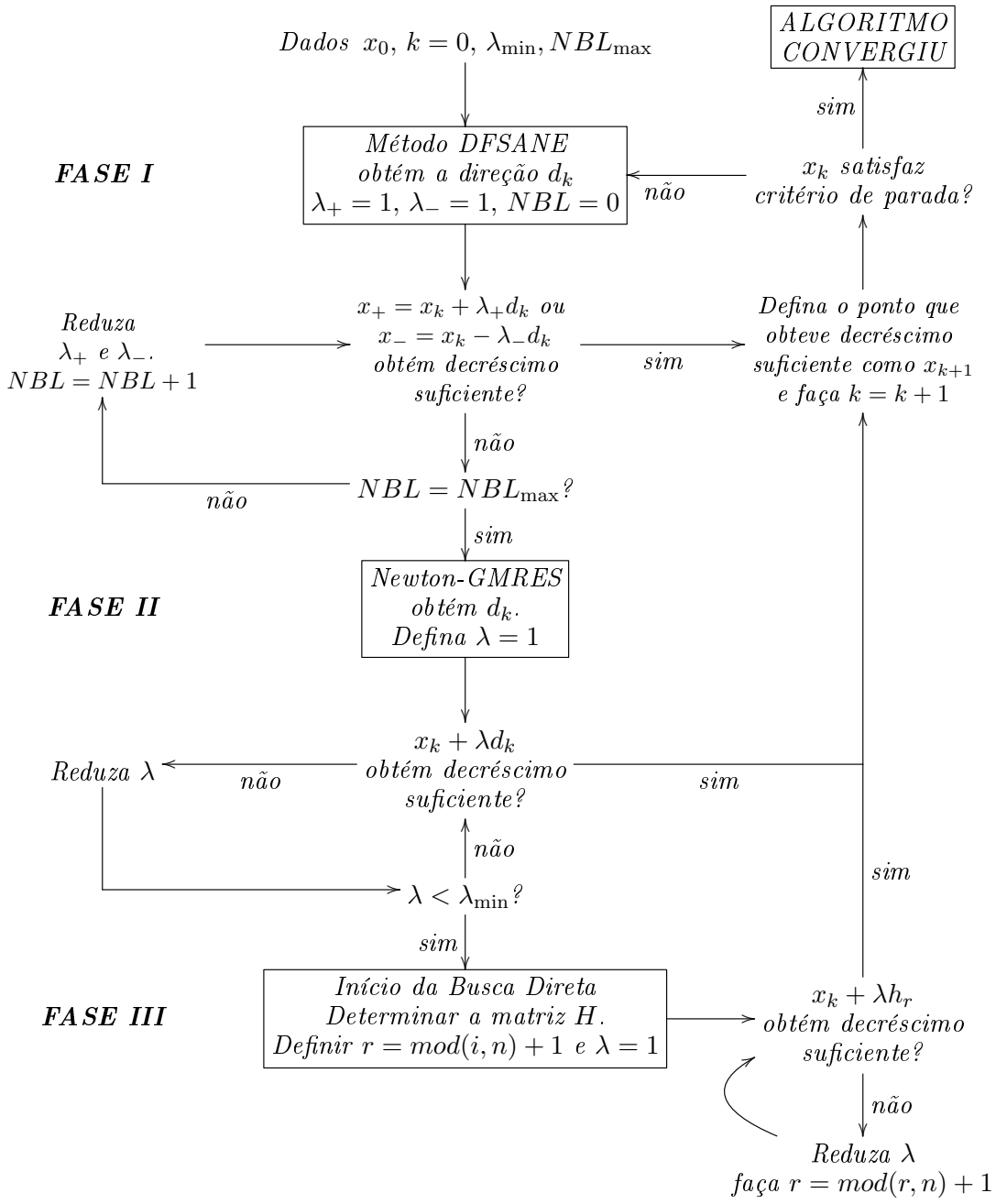

Figura 4.3: Diagrama representando o funcionamento do algoritmo HIB2

Capítulo 5

Considerações finais e indicações para trabalhos futuros

Neste trabalho foram propostos dois métodos híbridos para a resolução de sistemas não-lineares de grande porte: o Algoritmo 3.4 HIB1, que conta com duas fases, a primeira baseada em passos do tipo resíduo espectral ([14, 13]) seguida de uma fase correspondente ao método de Newton inexato e o Algoritmo 4.2 HIB2, que aciona uma terceira fase baseada em um método de busca direta, após falha nas duas fases do algoritmo anterior.

Do ponto de vista teórico, a primeira contribuição deste trabalho é o resultado do Teorema 3.4, que garante a existência de um ponto de acumulação da sequência gerada pelo método híbrido representado pelo Algoritmo 3.4 que é solução do sistema não linear. Além disso, o mesmo resultado pode ser estendido para o método de Newton inexato convencional, já que o algoritmo aceita dados de entrada de maneira que a fase correspondente ao método do resíduo espectral não é utilizada. Dessa forma, é possível configurar o método de Newton inexato com um critério de aceitação para novos pontos mais flexível do que aquele desenvolvido em [27].

Ainda com relação a este algoritmo, existe a possibilidade de realizar uma análise teórica e determinar as condições necessárias para estabelecer um resultado com respeito à velocidade de convergência do método. Este é um tema que ficou em aberto e no qual pretendemos trabalhar futuramente.

A segunda contribuição teórica deste trabalho é o resultado do Teorema 4.4, que assegura a convergência do Algoritmo 4.2. Embora esse resultado seja parecido com a Proposição 6.1 encontrada em [30], traz a novidade de permitir a substituição da estratégia de globalização através de buscas lineares como proposto em [30] por uma que utiliza a busca multidirecional proposta no presente trabalho, conforme Algoritmo 4.3. A busca multidirecional que propomos tem uma certa vantagem quando trabalhamos com problemas de grande porte pois, com essa abordagem, ao invés de reduzir o tamanho do passo sucessivas vezes em uma determinada direção antes de partir para a próxima, mudam-se as direções enquanto o tamanho do passo é reduzido evitando, dessa maneira, uma possível insistência em direções ruins, o que, na prática, demandaria em um número excessivo de avaliações de funções.

O Algoritmo 4.2 conta ainda com um conjunto inédito de direções de sondagem ortogonais, no qual estão contidas as direções utilizadas pelo método do resíduo espectral. No entanto, cabe destacar que embora o Algoritmo 4.2 esteja configurado com esse conjunto de direções de sondagem, o uso do mesmo não é necessário, de maneira que pode ser substituído por qualquer outra base positiva ortogonal que os resultados de convergência serão mantidos.

Para a etapa correspondente ao método tipo resíduo espectral DFSANE [13], foram elaboradas duas estratégias de ordenação das direções e, também, duas novas estratégias que utilizam, quando necessário, uma terceira direção de busca. As primeiras modificações não afetam os resultados de convergência, de maneira que não é necessária qualquer tipo de demonstração teórica. O mesmo não ocorre com as modificações do segundo tipo, já que, neste caso um novo resultado de convergência deve ser apresentado, a menos que o uso desta terceira direção seja limitado. Ressaltamos, entretanto, que essa limitação não acarreta em qualquer modificação no algoritmo prático.

Analisando agora pelo viés dos resultados computacionais, o Algoritmo 3.4 se mostrou competitivo perante os algoritmos DFSANE e Newton inexato utilizado de maneira pura, sendo que nos problemas mais difíceis teve um desempenho superior tanto em robustez quanto em eficiência. O Algoritmo 4.2 foi implementado com o objetivo de melhorar a robustez do Algoritmo 3.4 e cumpre esse papel.

Futuramente, temos a intenção de utilizar o Algoritmo 4.2 para resolução de sistemas não lineares mais difíceis e com menores dimensões. Para isso, a ideia é realizar os testes com o algoritmo híbrido e um novo algoritmo que utiliza somente a fase espectral combinada com a busca direta aqui proposta. Além disso, pretendemos testar esse último algoritmo para problemas não suaves. Neste caso, no entanto, é necessário o desenvolvimento de resultados teóricos que justifiquem a utilização da busca multidirecional neste tipo de problema.

Por fim, as alterações propostas para o método do resíduo espectral tiveram o desempenho prático analisado quando implementadas no algoritmo DFSANE e também quando do uso com o algoritmo híbrido. As mudanças que previam uma nova direção de descida se mostraram ligeiramente mais robustas do que se utilizadas as estratégias em sua forma pura.

Dentre os algoritmos com estratégia de ordenação, os algoritmos `DFSANE-O2`, `H1E-O2` e `H6E-O2` que testam primeiramente a direção cujo sentido foi utilizado na iteração anterior (ao invés de testar sempre pelo sentido negativo e depois o positivo) tiveram desempenhos similares ao algoritmo original. Já a proposta de avaliar as duas direções e, após isso, decidir em que sentido prosseguir não teve o impacto em melhorias de robustez que previámos quando estávamos dispostos a sacrificar a eficiência.

Pudemos notar, portanto, que a direção positiva tende a ter melhor desempenho que a direção negativa. Provavelmente este fato ocorre pois a direção no sentido positivo é uma aproximação, ainda que grosseira, de um método quase Newton para sistemas não lineares e, quando as condições geométricas dos problemas são favoráveis a aplicação deste tipo de método, é natural que seu desempenho seja superior ao da direção oposta.

Apêndice

A

Apêndice

Este apêndice contém as seguintes tabelas:

A.1 Problemas-testes propostos em La Cruz & Raydan [14] - Parte I

A.2 Problemas-testes propostos em La Cruz & Raydan [14] - Parte II

A.3 Problemas-testes propostos em Lukšan & Vlček [39] - Parte I

A.4 Problemas-testes propostos em Lukšan & Vlček [39] - Parte II

A.5 Problemas de [39], com pontos iniciais mais afastados da solução - Parte I

A.6 Problemas de [39], com pontos iniciais mais afastados da solução - Parte II

A.7 Problemas de [39], com pontos iniciais mais afastados da solução - Parte III

A.8 Problemas de [39], com pontos iniciais mais afastados da solução - Parte IV

	Problema	x_0 **padrão**	**dimensões**		
1	Exponential function 1	$x_0 = (n/(n-1), n/(n-1), \ldots, n/(n-1))^t$	1000	5000	10000
2	Exponential function 2	$x_0 = (1/n^2, 1/n^2 \ldots, 1/n^2))^t$	500	1000	2000
3	Exponential function 3	$x_0 = (1/4n^2, 2/4n^2, \ldots, n/4n^2)^t$	50	100	200
4	Diagonal function premultiplied by a quasi-orthogonal matrix	$x_0 = (-1, 1/2, \ldots, -1, 1/2)^t$	99	399	999
5	Extended Rosenbrock function	$x_0 = (5, 1, \ldots, 5, 1)^t$	1000	5000	10000
6	Chandrasekhar's H-equation	$x_0 = (1, 1, \ldots, 1)^t$	100	500	1000
7	Badly scaled augmented Powell's function	$x_0 = (10^{-3}, 18, 1, \ldots, 10^{-3}, 18, 1)^t$	9	99	399
8	Trigonometric function	$x_0 = (101/(100n), 101/(100n), \ldots, 101/(100n))^t$	1000	5000	10000
9	Singular function	$x_0 = (1, 1, \ldots, 1)^t$	2500	5000	10000
10	Logarithmic function	$x_0 = (1, 1, \ldots, 1)^t$	5000	10000	15000

Tabela A.1: Problemas-testes propostos em La Cruz & Raydan [14] - Parte I

	Problema	x_0 **padrão**	**dimensões**		
11	Broyden Tridiagonal function	$x_0 = (-1, -1, \ldots, -1)^t$	500	1000	2000
12	Trigexp function	$x_0 = (0, 0, \ldots, 0)^t$	100	500	1000
13	Variable band function 1	$x_0 = (0, 0, \ldots, 0)^t$	100	500	1000
14	Variable band function 2	$x_0 = (0, 0, \ldots, 0)^t$	100	500	1000
15	Function 15	$x_0 = (-1, -1, \ldots, -1)^t$	500	1000	5000
16	Strictly convex function 1	$x_0 = (1/n, 2/n, \ldots, 1)^t$	1000	10000	50000
17	Strictly convex function 2	$x_0 = (1, 1, \ldots, 1)^t$	100	500	1000
18	Function 18	$x_0 = (0, 0, \ldots, 0)^t$	399	999	9999
19	Zero Jacobian function	$x_0(1) = 100(n - 100)/n$ $x_0(j) = (n - 1000)(n - 500)/(60n)^2, \ \forall j \geq 2$	100	500	1000
20	Geometric programming function	$x_0 = (1, 1, \ldots, 1)^t$	50	100	500

Tabela A.2: Problemas-testes propostos em La Cruz & Raydan [14] - Parte II

	Problema	x_0 **padrão**
1	Countercurrent reactors problem 1 (modified)	$x_i = 0.1$, se $mod(i,8) = 1$; $x_i = 0.2$, se $mod(i,8) = 2$ ou $mod(i,8) = 0$; $x_i = 0.3$, se $mod(i,8) = 3$ ou $mod(i,8) = 7$; $x_i = 0.4$, se $mod(i,8) = 4$ ou $mod(i,8) = 6$; $x_i = 0.5$, se $mod(i,8) = 5$.
2	Countercurrent reactors problem 2 (modified)	
3	Trigonometric system	$x_0 = (1/n, 1/n, \ldots, 1/n)^t$
4	Trigonometric - exponential system (trigexp 1)	$x_0 = (0, 0, \ldots, 0)^t$
5	Trigonometric - exponential system (trigexp 2)	$x_0 = (1, 1, \ldots, 1)^t$
6	Singular Broyden problem	$x_0 = (-1, -1, \ldots, -1)^t$
7	Tridiagonal System	$x_0 = (12, 12, \ldots, 12)^t$
8	Five-diagonal system	$x_0 = (-2, -2, \ldots, -2)^t$
9	Seven-diagonal system	$x_0 = (-3, -3, \ldots, -3)^t$
10	Structured Jacobian problem	$x_0 = (-1, -1, \ldots, -1)^t$
11	Tridiagonal system Extended Freudenstein and Roth function	$x_0 = (90, 60, 90, 60, \ldots, 90, 60)^t$
12	Extended Powell singular problem	$x_0 = (3, -1, 0, 1, 3, -1, 0, 1, \ldots, 3, -1, 0, 1)^t$
13	Extended Cragg and Levy problem	$x_0 = (1, 2, 2, 2, 1, 2, 2, 2, \ldots, 1, 2, 2, 2)^t$
14	Broyden tridiagonal problem	$x_0 = (-1, -1, \ldots, -1)^t$
15	Generalized Broyden banded problem	$x_0 = (-1, -1, \ldots, -1)^t$

Tabela A.3: Problemas-testes propostos em Lukšan & Vlček [39] - Parte I

	Problema	x_0 **padrão**
16	Extended Powell badly scaled function	$x_0 = (0, 1, 0, 1, \ldots, 0, 1)^t$
17	Extended Wood problem	$x_0 = (-3, -1, -3, -1, \ldots, -3, -1)^t$
18	Tridiagonal exponential problem	$x_0 = (1.5, 1.5, \ldots, 1.5)^t$
19	Discrete boundary value problem	$x_0 = (h(h-1), 2h(2h-1), \ldots, nh(nh-1)^t$ com $h = 1/(n+1)$
20	Brent problem	$x_0 = (10, 10, \ldots, 10)^t$
21	Troesch problem	$x_0 = (1, 1, \ldots, 1)^t$
22	Flow in a channel	$u_0(x) = (x - 0.5)^2$
23	Swirling flow	$u_0(x) = (x - 0.5)^2$ e $v_0(x) = x - 0.5$
24	Bratu Problem	$u_0(x, y) = 0$
25	Poisson Problem 1	$u_0(x, y) = -1$
26	Poisson Problem 2	$u_0(x, y) = 0$
27	Porous medium problem	$u_0(x, y) = 1 - xy$
28	Convection-difussion problem	$u_0(x, y) = 0$
29	Nonlinear biharmonic problem	$u_0(x, y) = 0$
30	Driven cavity problem	$u_0(x, y) = 0$

Tabela A.4: Problemas-testes propostos em Lukšan & Vlček [39] - Parte II

PROBLEMA 3									
v	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	C	EST	EST	EST	EST	EST	EST	EST
DFSANE	C	C	C	C	EAVF	C	EST	EST	EST
H6E	C	C	EST	EST	EST	EST	EST	EST	EST
NIWD	C	C	EST	EST	EST	EST	EST	EST	EST
NI	C	C	EST	EST	EST	EST	EST	EST	EST

PROBLEMA 4									
v	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	EST	C	C	C	C	C	C	C
DFSANE	C	C	C	C	C	C	C	C	C
H6E	C	C	C	C	C	C	C	C	C
NIWD	C	C	C	C	C	C	C	C	C
NI	C	C	C	C	C	C	C	C	C

PROBLEMA 6									
v	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	C	C	EST	EST	C	EST	EST	EST
DFSANE	EAVF	C	C	EAVF	EAVF	C	EAVF	EAVF	EAVF
H6E	EAVF	C	C	EST	EST	C	EST	EST	EST
NIWD	C	C	C	EST	EST	C	EST	EST	EST
NI	C	C	C	EST	EST	C	EST	EST	EST

PROBLEMA 7									
v	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	C	C	C	C	C	EAVF	C	C
DFSANE	C	C	C	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF
H6E	C	C	C	C	C	C	EAVF	C	C
NIWD	C	C	C	C	C	EST	C	C	C
NI	C	C	C	C	C	C	EAVF	C	C

Tabela A.5: Problemas de [39], com pontos iniciais mais afastados da solução - Parte I

PROBLEMA 8									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	EST	C	C	EST	EST	C	EST	EST	EST
DFSANE	C	C	EAVF	EAVF	C	C	EAVF	EAVF	C
H6E	C	C	C	C	C	C	EST	EST	EST
NIWD	EST	C	C	EST	EST	C	EST	EST	EST
NI	EST	C	C	EST	EST	C	EST	EST	EST

PROBLEMA 9									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	EST	C	C	EST	EST	EST	EST	EST	EST
DFSANE	C	C	C	EAVF	C	C	EAVF	EAVF	C
H6E	C	C	C	EST	C	EST	EST	EST	EST
NIWD	EST	C	C	EST	EST	EST	EST	EST	EST
NI	EST	C	C	EST	EST	EST	EST	EST	EST

PROBLEMA 11									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	EST	C	C	C	C	C	C	C
DFSANE	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF
H6E	EST	C	C	C	C	C	C	C	C
NIWD	C	C	C	C	C	C	C	C	C
NI	C	C	C	C	C	C	C	C	C

PROBLEMA 12									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	EST	EST	EST	EST	EST	EST	EST	EST	EST
DFSANE	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF
H6E	EST	C	EST	EST	EST	EST	EST	EST	EST
NIWD	EST	EST	EST	EST	EST	EST	EST	EST	EST
NI	EST	EST	EST	EST	EST	EST	EST	EST	EST

Tabela A.6: Problemas de [39], com pontos iniciais mais afastados da solução - Parte II

PROBLEMA 13									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	EST	EST	PPF	PPF	EST	PPF	PPF	PPF	PPF
DFSANE	EAVF	EAVF	PPF	EAVF	EAVF	PPF	PPF	PPF	PPF
H6E	EST	EST	PPF	PPF	EST	PPF	PPF	PPF	PPF
NIWD	EST	EST	PPF	EST	EST	PPF	PPF	PPF	PPF
NI	EST	EST	PPF	EST	EST	PPF	PPF	PPF	PPF

PROBLEMA 14									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	EST	EST	EST	EST	EST	EST	EST	EST
DFSANE	C	C	EAVF	EAVF	EAVF	C	EAVF	EAVF	EAVF
H6E	C	C	EST	EST	EST	C	EST	EST	EST
NIWD	C	EST	EST	EST	EST	EST	EST	EST	EST
NI	C	EST	EST	EST	EST	EST	EST	EST	EST

PROBLEMA 15									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	C	C	C	C	C	C	C	C
DFSANE	C	C	C	C	C	C	C	C	C
H6E	C	C	C	C	C	C	C	C	C
NIWD	C	C	C	C	C	C	C	C	C
NI	C	C	C	C	C	C	C	C	C

PROBLEMA 16									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	EST	EST	EST	EST	EST	EST	EST	EST
DFSANE	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF	EAVF
H6E	C	C	EST	EST	EST	EST	EST	EST	EST
NIWD	C	EST	EST	EST	EST	EST	EST	EST	EST
NI	C	EST	EST	EST	EST	EST	EST	EST	EST

Tabela A.7: Problemas de [39], com pontos iniciais mais afastados da solução - Parte III

PROBLEMA 17									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	EST	EST	EST	EST	EST	EST	EST	EST	EST
DFSANE	EAVF	C	C	EAVF	EAVF	C	EAVF	EAVF	EAVF
H6E	EST	EST	EST	EST	EST	EST	EST	EST	EST
NIWD	EST	EST	EST	EST	EST	EST	EST	EST	EST
NI	EST	EST	EST	EST	EST	EST	EST	EST	EST

PROBLEMA 18									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	C	C	C	C	C	C	C	C
DFSANE	C	C	C	C	C	C	C	C	C
H6E	C	C	C	C	C	C	C	C	C
NIWD	C	C	C	C	C	C	C	C	C
NI	C	C	C	C	C	C	C	C	C

PROBLEMA 19									
υ	1			20			200		
ω	1	20	200	1	20	200	1	20	200
H1E	C	C	C	C	C	C	C	C	C
DFSANE	C	C	C	C	C	C	C	C	C
H6E	C	C	C	C	C	C	C	C	C
NIWD	C	C	C	C	C	C	C	C	C
NI	C	C	C	C	C	C	C	C	C

Tabela A.8: Problemas de [39], com pontos iniciais mais afastados da solução - Parte IV

Referências Bibliográficas

[1] M. A. Abramson, C. Audet, J.E. Dennis Jr. & S. Digabel, ORTHOMADS: A deterministic MADS instance with orthogonal directions, *SIAM Journal on Optimization*, 20, pp. 948-966, 2009.

[2] L. Armijo, Minimization of functions having Lipschitz-continuous first partial derivatives, *Pacific Journal of Mathematics*, 16, pp. 1-3, 1966.

[3] C. Audet & J. E. Dennis Jr., Analysis of generalized pattern searches, *SIAM Journal on Optimization*, 13, pp. 889-903, 2002.

[4] C. Audet & J. E. Dennis Jr., Mesh adaptive direct search algorithms for constrained optimization, *SIAM Journal on Optimization*, 17, pp. 188-217, 2006.

[5] J. Barzilai & J. M. Borwein, Two point step size gradient methods, *IMA Journal of Numerical Analysis*, 8, pp. 141-148, 1988.

[6] M. S. Bazaraa, H. D. Sherali & C. M. Shetty, "Nonlinear Programming.. Theory and Algorithms", 3ed., Wiley, New Jersey, 2006.

[7] G. E. P. Box, Evolutionary operation: A method for increasing industrial productivity, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 6, pp. 81-101, 1957.

[8] P. N. Brown & Y. Saad, Hybrid methods for nonlinear systems of equations, *SIAM Journal on Scientific and Statistical Computing*, 11, pp. 450-481, 1990.

[9] C. G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation*, 19, pp. 577-593, 1965.

[10] C. G. Broyden, J. E. Dennis Jr. & J. J. Moré, On the local and superlinear convergence of quasi-Newton methods, *IMA Journal of Applied Mathematics*, 12, pp. 223-245, 1973.

[11] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal & H. C. Pedersen, The watchdog technique for forcing convergence in algorithms for constrained optimization, *Mathematical Programming Studies*, 16, pp. 1-17, 1982.

[12] A. R. Conn, K. Scheinberg & L. N. Vicente, "Introduction to Derivative-Free Optimization", MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.

[13] W. La Cruz, J. M. Martínez & M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Mathematics of Computations*, 75, pp. 1429-1448, 2006.

[14] W. La Cruz & M. Raydan, Nonmonotone spectral methods for large-scale nonlinear systems, *Optimization Methods and Software*, 18, pp. 583-599, 2003.

[15] A. L. Custódio, Aplicações de derivadas simplécticas em métodos de procura directa, Tese de Doutorado, FCT-UNL, Lisboa, 2007.

[16] A. L. Custódio, comunicação privada, 2010.

[17] C. Davis, Theory of positive linear dependence, *American Journal of Mathematics*, 76, pp. 733-746, 1954.

[18] R. S. Dembo, S.C. Eisenstat & T. Steihaug, Inexact Newton methods, *SIAM Journal on Numerical Analysis*, 19, pp. 401-408, 1982.

[19] J. E. Dennis & R. B. Schnabel, "Numerical Methods for Unconstrained Optimization and Nonlinear Equations", SIAM, Philadelphia, 1996.

[20] E. D. Dolan & J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming*, 91, pp. 201-213, 2002.

[21] S. C. Eisenstat & H. F. Walker, Choosing the forcing terms in inexact-Newton methods, *SIAM Journal on Scientific Computing*, Vol.17, No.1, pp.16-32, 1996.

[22] E. Fermi & N. Metropolis, Los Alamos unclassified report LS-1492, relatório técnico, *Los Alamos National Laboratory*, EUA, 1952.

[23] R. Fletcher, Low storage methods for unconstrained optimization, *Lectures in Applied Mathematics*, 26, American Mathematical Society, pp. 165-179, 1990.

[24] A. Friedlander, J. M. Martínez & M. Raydan, A new method for large-scale box constrained convex quadratic minimization problems, *Optimization Methods and Software*, 5, pp. 57-74, 1995.

[25] M. A. Gomes-Ruggiero, V. L. R. Lopes & J.V. Toledo-Benavides, A globally convergent inexact Newton method with a new choice for the forcing term, *Annals of Operations Research*, 157, pp. 193-205, 2007.

[26] M. A. Gomes-Ruggiero, V. L. R. Lopes & J.V. Toledo-Benavides, A safeguard approach to detect stagnation of GMRES(m) with applications in Newton-Krylov methods, *Computational & Applied Mathematics*, pp. 175-199, 2008.

[27] L. Grippo, F. Lampariello & S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis*, 23, pp. 707-716, 1986.

[28] L. Grippo & M. Sciandrone. Nonmonotone globalization techniques for the Barzilai-Borwein gradient method. *Computational Optimization and Applications*, 23, pp. 143-169, 2002.

[29] L. Grippo & M. Sciandrone, Nonmonotone derivative-free methods for nonlinear equations, *Computational Optimization and Applications*, 37, pp. 297-328, 2007.

[30] L. Grippo & M. Sciandrone, Nonmonotone globalization of the finite-difference Newton-GMRES method for nonlinear equations, *Optimization Methods and Software*, 25, pp. 971-999, 2010.

[31] L. Grippo & M. Sciandrone, Nonmonotone globalization of inexact finite-difference Newton-iterative methods for nonlinear equations, Tech. Rep. Dis n.14, 2005.

[32] J. H. Halton, On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals, *Numerische Mathematik*, 2, pp. 84-90, 1960.

[33] R. Hooke & T. A. Jeeves, "Direct search" solution of numerical and statistical problems, *Journal of the ACM (JACM)*, 8, pp. 212-229, 1961.

[34] C. T. Kelley, "Iterative Methods for Linear and Nonlinear Equations", SIAM, Philadelphia, 1995.

[35] T. G. Kolda, R. M. Lewis & V. Torczon, Optimization by direct search: new perspectives on some classical and modern methods, *SIAM Review*, 45, pp. 385-482, 2003.

[36] R. M. Lewis & V. Torczon, Rank ordering and positive bases in pattern search algorithms, Relatório Técnico 96-71, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, EUA, 1996.

[37] D. H. Li & M. Fukushima, A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, *Optimization Methods and Software*, 13, pp. 181-201, 2000.

[38] L. Lukšan, Inexact trust region method for large sparse system of nonlinear equations, *Journal of Optimization Theory and Applications*, Vol. 81, pp. 569-590, 1994.

[39] L. Lukšan, J. Vlček, Sparse and partially separable test problems for unconstrained and equality constrained optimization, Report V-767, Prague, ICS AS CR, 1998.

[40] J. M. Martínez, & S. A. Santos, "Métodos Computacionais de Otimização", IMPA, SBM, 20^o. Colóquio Brasileiro de Matemática, Rio de Janeiro, 1995.

[41] J. A. Nelder & R. Mead, A simplex method for function minimization, *Computer Journal*, 7, pp. 308-313, 1965.

[42] J. Nocedal & S. J. Wright, "Numerical Optimization", Springer, New York, 1999.

[43] M. Pernice & H. Walker, NITSOL: a Newton iterative solver for nonlinear systems, *SIAM Journal on Scientific Computing*, 19, No. 1, pp. 302-318, 1998.

[44] M. Raydan, On the Barzilai Borwein choice of steplenght for the gradient method, *IMA Journal of Numerical Analysis*, 13, pp. 321-326, 1993.

[45] M. Raydan, The Barzilai Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization*, 7, pp. 26-33, 1997.

[46] Y. Saad, "Iterative Methods for Sparse Linear Systems", 2.ed with corrections, disponível em `http://www-users.cs.umn.edu/~saad/books.html`.

[47] Y. Saad & M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific Computing*, 7, pp. 856-869, 1986.

[48] R. B. Schnabel & P. D. Frank, Tensor methods for nonlinear equations, *SIAM Journal on Numerical Analysis*, 21, pp. 815-843, 1984.

[49] W. Spendley, G. R. Hext & F. R. Himsworth, Sequential application of simplex designs in optimisation and evolutionary operation, *Technometrics*, 4, pp. 441- 461, 1962.

[50] V. Torczon, Multi-directional search: a direct search algorithm for parallel machines, Tese de Doutorado, Department of Mathematical Sciences, Rice University, EUA, 1989.

[51] L. N. Vicente, Worst case complexity of direct search, *preprint* 10-17, Dept. de Matemática, Univ. Coimbra, 2010.

[52] H. F. Walker & L. Zhou, A simpler GMRES, *Numerical Linear Algebra with Applications*, 1, pp. 571-581, 1994.

[53] D. Watkins, "Fundamentals of Matrix Computations", John Wiley & Sons, New York, 2002.

[54] T. J. Ypma, Historical development of the Newton-Raphson method, *SIAM Review*, 37, pp. 531-551, 1995.