
UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA
E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

UM MÉTODO NEWTON-INEXATO COM ESTRATÉGIA HÍBRIDA PARA GLOBALIZAÇÃO

Dissertação de Mestrado

Rodolfo Gotardi Begiato

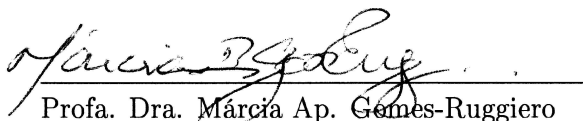
Orientadora: Márcia Aparecida Gomes-Ruggiero

CAMPINAS
MAIO DE 2007

**Um método Newton-inexato
com estratégia híbrida para globalização**

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Rodolfo Gotardi Begiato e aprovada pela comissão julgadora.

Campinas, 18 de maio de 2007


Prof. Dra. Márcia Ap. Gomes-Ruggiero
Orientadora

Banca Examinadora

1. Márcia Aparecida Gomes-Ruggiero (DMA-IMECC-UNICAMP)
2. Mário César Zambaldi (DM-CFM-IFSC)
3. Sandra Augusta Santos (DMA-IMECC-UNICAMP)

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito para a obtenção do título de Mestre em Matemática Aplicada.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Begiato, Rodolfo Gotardi

B394u Um método Newton-Inexato com estratégia híbrida para
globalização / Rodolfo Gotardi Begiato -- Campinas, [S.P. :s.n.], 2007.

Orientadora: Márcia Aparecida Gomes Ruggiero.

Dissertação (mestrado) - Universidade Estadual de Campinas,
Instituto de Matemática, Estatística e Computação Científica.

1. Sistemas não-lineares. 2. Métodos iterativos (Matemática). 3.
Otimização matemática. I. Ruggiero, Márcia Aparecida Gomes. II.
Universidade Estadual de Campinas. Instituto de Matemática,
Estatística e Computação Científica. III. Título.

Título em inglês: A Newton-Inexact algorithm with a hybrid method for globalization.

Palavras-chave em inglês (Keywords): 1. Nonlinear systems. 2. Iterative methods. 3.
Mathematical optimization.

Área de concentração: Otimização matemática

Titulação: Mestre em Matemática Aplicada

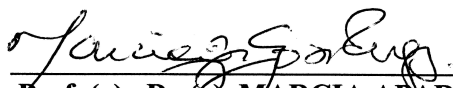
Banca examinadora: Profa. Dra. Márcia Aparecida Gomes Ruggiero (IMECC-Unicamp)
Prof. Dr. Mário César Zambaldi (CFM/DM-UFSC)
Profa. Dra. Sandra Augusta Santos (IMECC-Unicamp)

Data da defesa: 18/05/2007

Programa de Pós-Graduação: Mestrado em Matemática Aplicada

Dissertação de Mestrado defendida em 18 de maio de 2007 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.



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



Prof. (a). Dr (a). SANDRA AUGUSTA SANTOS



Prof. (a). Dr (a). MARIO CESAR ZAMBALDI



AGRADECIMENTOS

Agradeço:

- À Profa. Dra. Márcia Aparecida Gomes-Ruggiero, pela orientação e dedicação.
- Ao Prof. Dr. Márcio César Zambaldi e às Profas. Dras. Sandra Augusta Santos e Maria Aparecida Diniz Ehrhardt, pelos valiosos comentários a respeito deste trabalho.
- Ao grupo de otimização da Unicamp, pela aprendizagem proporcionada.
- Aos funcionários do IMECC, por me ajudar com prazos e datas, impedindo minha exclusão do programa por motivos burocráticos.
- 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, por tudo.



RESUMO

O principal objetivo deste trabalho é a proposta de uma estratégia híbrida de globalização para o método de Newton-inexato. Assim como o método de Newton, o método de Newton-inexato tem sua convergência garantida somente em vizinhanças adequadas da solução do sistema e uma estratégia de globalização deve, portanto, ser incorporada. Estratégias de globalização se baseiam na minimização de funções de mérito e duas abordagens podem ser consideradas: busca linear e regiões de confiança. Neste trabalho optamos pelo uso conjunto das duas abordagens, resultando numa estratégia híbrida, envolvendo inicialmente uma sequência de buscas lineares, e se necessário, prossegue-se com uma variação da estratégia Dogleg, proposta por Powell em 1970.

Para a resolução aproximada de sistemas lineares foi utilizado o método GMRES, que faz parte de métodos de projeções sobre subespaços de Krylov. Este método possibilita a implementação com a estratégia matrix-free. Para reduzir o uso de requerimentos de memória, optamos ainda pelo uso do método GMRES com recomeços. A eficiência dos algoritmos desenvolvidos foi avaliada através da resolução de um conjunto de sistemas não lineares acadêmicos e um conjunto de sistemas sistemas não-lineares resultantes da discretização de problemas de valor de contorno. Estes testes comprovaram a eficiência da estratégia híbrida empregada no processo de globalização.

ABSTRACT

The main objective of this work is to propose a hybrid globalization strategy for inexact-Newton method. Globalization strategies are based on line search or trust region procedures.

In this work, we choose a hybrid strategy which involves a cycle of line search and a variation of Powell dogleg trust region. For solving the linear systems we chose the GMRES method with restarts and to avoid the calculation of Jacobian matrices we used a matrix-free strategie.

The numerical performance of the algorithms was evaluated by means a set of academic problems and a set of nonlinear systems of boundary value problem discretization. These results showed the good performance of hybrid globalization strategy.

SUMÁRIO

1	Introdução	1
2	Noções Teóricas	5
2.1	Método de Newton Inexato	5
2.2	Estratégias de Globalização	7
2.2.1	O Passo <i>Dogleg</i>	12
2.2.2	Alterações no raio da região de confiança	17
2.3	Método GMRES	19
2.3.1	Rotações de Givens	24
2.3.2	GMRES com Recomeços	27
3	O Método Newton-GMRES(m) com uso de estratégia Matrix-Free	29
3.1	<i>Dogleg</i> e globalização	31
3.2	Modificações Necessárias para o Processo de Recomeços	35
4	Implementação e Testes Realizados	39
4.1	Algoritmos testados	39
4.1.1	Detalhamento da implementação	40
4.1.2	Análise de Desempenho	42
4.2	Testes iniciais	43
4.3	Problemas de Valor de Contorno	47
5	Conclusões	60
	Referências Bibliográficas	62

A proposta deste trabalho é resolver sistemas não-lineares do tipo:

$$\begin{aligned} F(x) &= 0 \\ F : \mathbb{R}^n &\rightarrow \mathbb{R}^n \\ \text{com } F &\in C^1(\mathbb{R}^n) \end{aligned} \tag{1.1}$$

Entre os métodos clássicos para a resolução de sistemas não-lineares encontra-se o método de Newton. Denotando a matriz Jacobiana de F por J , o método de Newton aplicado à resolução de (1.1) consiste em realizar iterações do tipo:

$$x_{k+1} = x_k + s_k \tag{1.2}$$

onde x_k é o ponto atual e s_k é direção obtida pela solução do seguinte sistema linear:

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

O método de Newton tem como principal vantagem a taxa de convergência quadrática. Resultados sobre a convergência local do método de Newton são facilmente encontrados na literatura, como por exemplo em [11] ou [13]. Apesar dessa atratividade, o método de Newton apresenta algumas dificuldades na resolução de problemas de grande porte, pois o cálculo da matriz Jacobiana, bem como a posterior resolução do sistema linear podem ser computacionalmente caros ou até impraticáveis.

Dembo, Eisenstat e Steihaug propuseram, em 1982, o método Newton-Inexato [4], que não preserva a taxa de convergência quadrática do método de Newton, mas requer um trabalho computacional consideravelmente menor por

iteração. A proposta do método é, a cada iteração, obter uma direção s_k que satisfaça a seguinte condição:

$$\|J(x_k)s_k + F(x_k)\| \leq \eta_k \|F(x_k)\| \quad (1.4)$$

onde $\eta_k \in (0, 1]$.

As variações entre as implementações do método Newton-inexato ocorrem basicamente por causa de diferentes escolhas que podem ser adotadas em três etapas do método:

1. método para resolução do sistema linear (1.3);
2. escolha para os valores do parâmetro η_k na equação (1.4), denominado termo forçante;
3. estratégia para garantir a convergência global do método Newton-inexato.

Usualmente o passo s_k que satisfaz a equação (1.4) é obtido através de um método iterativo para resolução de sistemas lineares. Neste trabalho para a resolução aproximada do sistema (1.3) e satisfação da condição (1.4) utilizaremos o método GMRES (*Generalized Minimal Residual*). O GMRES proposto em 1986 por Saad e Schultz [17] é um método iterativo e faz parte da família de métodos de projeção sobre subespaços de Krylov, da qual também faz parte, por exemplo, o método dos gradientes conjugados. Neste último, no entanto, só podemos trabalhar com matrizes simétricas definidas positivas, o que não ocorre no GMRES. Quando utilizamos esta estratégia para a resolução do sistema (1.3), o método de Newton-inexato recebe o nome de Newton-GMRES.

Devido à sua estrutura, o GMRES permite que trabalhem com um recurso *matriz-free* que elimina a necessidade de calcularmos a matriz Jacobiana. Como o GMRES utiliza a matriz Jacobiana somente para realizar o produto entre esta e um vetor v , iremos aproximar o resultado dessa operação por meio da seguinte expansão de séries de Taylor:

$$J(x_k)v \approx \frac{F(x_k + hv) - F(x_k)}{h}, \quad h \in \mathbb{R}$$

Embora o método apresente várias facilidades de implementação, seu custo computacional e os requerimentos de memória aumentam conforme o número de iterações realizadas. Uma saída para este problema é a estratégia $\text{GMRES}(m)$, que consiste em realizar no máximo m iterações do GMRES e caso não haja convergência recomeçamos o processo novamente, utilizando o valor encontrado na última iteração como chute inicial. Embora essa estratégia não tenha garantia teórica de convergência, tem sido aplicada com frequência e apresentado bons resultados na prática.

A escolha dos valores para o parâmetro η_k é de grande importância para o desempenho do método, uma vez que tem influência direta em sua taxa de convergência. Devemos ainda salientar que um valor muito pequeno certamente irá nos possibilitar a realização de poucas iterações do tipo (1.2) as quais chamaremos de iterações externas, porém valores muito próximos de zero provavelmente irão causar o fenômeno conhecido na literatura como *oversolving*, que consiste na realização de um grande número de iterações internas. Iterações internas são aquelas utilizadas para resolver aproximadamente o sistema (1.3) a fim de alcançar a desigualdade (1.4).

Portanto, a escolha dos valores para o termo forçante η_k deve ser cuidadosa. Uma opção é a escolha de uma sequência η_k constante, mas conforme será visto posteriormente, se nos cuidarmos para que a sequência $\{\eta_k\}$ convirja para zero conforme $k \rightarrow \infty$, iremos garantir uma taxa de convergência super-linear para o método Newton-inexato. Neste trabalho, a fim de garantir essa taxa, optamos pelo uso de valores não-constantes e foi utilizada a estratégia apresentada por Eisenstat e Walker em [8].

Assim como os métodos de Newton, o método Newton-Inexato tem sua convergência assegurada somente nas regiões próximas à solução. Para obter resultados de convergência global é necessário adotar uma estratégia para globalização, que por sua vez está associada à minimização de uma função de mérito pré-estabelecida. Duas propostas amplamente usadas são busca linear e regiões de confiança.

Quando usamos a estratégia de busca linear, devemos inicialmente encontrar uma direção s_k e buscar um tamanho de passo que cumpra alguma condição de decréscimo no valor da função de mérito para garantir a globalização.

No caso de utilizarmos regiões de confiança, devemos aproximar a função de mérito f em torno do ponto x_k por um modelo quadrático completo e fixar o raio da região (a partir de x_k) que supomos poder confiar que o modelo irá descrever bem a situação real. E então, a próxima direção, s_k , será dada pelo minimizador do modelo quadrático na região determinada por:

$$\|s\|_2 = \|x - x_k\|_2 \leq \delta_k$$

onde δ_k é o raio da região de confiança. Assim como na estratégia de busca linear, devemos exigir que os pontos encontrados satisfaçam algum critério de decréscimo suficiente no valor da função de mérito.

Durante a implementação optamos pelo uso de três estratégias distintas para globalização. A primeira estratégia consiste somente no uso de regiões de confiança, através da estratégia *Double Dogleg* proposta por Dennis e Mei em 1979 [6]. Essa estratégia é uma variação da proposta *Dogleg* de Powell [14].

Além disso, utilizamos também uma estratégia híbrida, com no máximo três buscas lineares prosseguidas, se necessário, da estratégia de regiões de confiança *Double Dogleg*. Com essa proposta criamos dois algoritmos com critérios de aceitação distintos.

No Capítulo 2 deste trabalho, descrevemos os fundamentos teóricos que formam a base da construção do algoritmo implementado para a resolução de sistemas não-lineares. Neste capítulo encontram-se resultados sobre o método de Newton-Inexato, sobre o GMRES e a estratégia de regiões de confiança *Dogleg*.

No Capítulo 3 é apresentada a construção de um algoritmo Newton-Krylov, com regiões de confiança e uso da técnica *matrix-free*, fundamentado com argumentos do Capítulo 2. Neste algoritmo usamos o GMRES com recomeços, que exige algumas alterações com relação ao caso convencional.

Os resultados computacionais obtidos com os testes realizados são apresentados no Capítulo 4. E no Capítulo 5 é feita uma avaliação geral dos resultados obtidos, além de alguns apontamentos para futuras pesquisas.

2.1. MÉTODO DE NEWTON INEXATO

Quando se trata da resolução de problemas não-lineares do tipo (1.1), uma opção atraente devido à sua taxa de convergência quadrática é o método de Newton.

Se na atual iteração estivermos no ponto x_k , o método de Newton irá encontrar a direção s_k através da resolução do seguinte sistema linear:

$$J(x_k)s = -F(x_k) \quad (2.1)$$

E daí, faz-se: $x_{k+1} = x_k + s_k$, onde x_{k+1} seria o próximo ponto da sequência gerada pelo método.

Definição 2.1 *Uma aplicação f é dita Lipschitz contínua com escalar L em uma vizinhança de x^* se existe um escalar $L > 0$ tal que:*

$$\|f(x) - f(x^*)\| \leq L\|x - x^*\|$$

para todo x pertencente a essa vizinhança de x^ .*

Teorema 2.1 *Considere x^* a solução exata do sistema (1.1), suponha que existe $L > 0$ de tal forma que a matriz Jacobiana existe, é Lipschitz contínua com escalar L numa vizinhança de x^* e $\|J(x^*)^{-1}\|_2 \leq \beta$ para algum $\beta > 0$. Então, existe $\varepsilon > 0$ tal que para todo x_0 satisfazendo $\|x_0 - x^*\| \leq \varepsilon$ a sequência gerada por:*

$$x_{k+1} = x_k - J^{-1}(x_k)F(x_k), \quad k = 0, 1, \dots$$

está bem definida, converge a x^* e satisfaz:

$$\|x_{k+1} - x^*\| \leq \beta L \|x_k - x^*\|^2, \quad k = 0, 1, \dots \quad .$$

PROVA: Ver [5] ■

Apesar da taxa quadrática de convergência, a resolução de problemas de grande porte através do método de Newton torna-se computacionalmente cara, visto que devemos, a cada iteração, determinar a matriz Jacobiana em x_k e encontrar a solução exata para o sistema (2.1), o que necessita de grandes requerimentos de memória e intenso esforço computacional.

Uma alternativa frequentemente usada para aliviar esse esforço computacional é o chamado método de Newton inexato. Por esse método, devemos encontrar uma direção s_k que resolva o sistema (2.1) apenas aproximadamente, isto é, devemos encontrar uma direção s_k satisfazendo:

$$\|J(x_k)s_k + F(x_k)\| \leq \eta_k \|F(x_k)\|. \quad (2.2)$$

O valor η_k deve pertencer ao intervalo $(0, 1]$ e é chamado termo forçante.

A contrapartida em relação ao método de Newton tradicional é a perda da taxa de convergência quadrática. No método de Newton inexato a convergência é no máximo superlinear e depende das escolhas do termo forçante conforme será visto nos Teoremas 2.2 e 2.3.

O Algoritmo 2.1 abaixo nos dá uma idéia básica do método de Newton Inexato:

Algoritmo 2.1 *Método de Newton Inexato*

Considere x_0 uma aproximação inicial, $k = 0$, $\varepsilon > 0$ e $\|\cdot\|$ uma norma em $\mathbb{R}^n$. Enquanto $\|F(x_k)\| > \varepsilon$, faça:

1. Escolha $\eta_k \in (0, 1]$.
2. Encontre s_k que satisfaça a condição:

$$\|J(x_k)s_k + F(x_k)\| \leq \eta_k \|F(x_k)\|.$$

$$3. \ x_{k+1} = x_k + s_k.$$

$$4. \ k = k + 1.$$

Iremos expor agora dois teoremas que tratam da convergência do método de Newton inexato:

Teorema 2.2 *Se $\eta_k \leq \eta_{max} < r < 1$ para todo k e, além disso, existe $\varepsilon > 0$ tal que $\|x_0 - x^*\| \leq \varepsilon$, então a sequência $\{x_k\}$ gerada pelo método de Newton-inexato converge a x^* . E mais, a taxa de convergência é linear com taxa r :*

$$\|J(x^*)(x_{k+1} - x_k)\| \leq r \|J(x^*)(x_k - x^*)\|. \quad (2.3)$$

PROVA: Ver [4] ■

Teorema 2.3 *Se a sequência gerada pelo método de Newton inexato converge a x^* e se $\lim_{k \rightarrow \infty} \eta_k = 0$, então a taxa de convergência é superlinear.*

PROVA: Ver [4] ■

2.2. ESTRATÉGIAS DE GLOBALIZAÇÃO

Conforme pode-se concluir pelo Teorema 2.2, o método Newton-Inexato tem sua convergência assegurada somente nas regiões próximas à solução, assim como o método de Newton. Os resultados de convergência global são obtidos incorporando estratégias de globalização ao Algoritmo 2.1. Tais estratégias levam em consideração a minimização de funções de mérito associadas ao sistema (1.1), como por exemplo $f(x) = 0.5\|F(x)\|_2$ ou $f(x) = 0.5\|F(x)\|_2^2$. Tais estratégias são baseadas em busca linear e/ou regiões de confiança.

Definimos a seguir uma direção de descida para f em x_k .

Definição 2.2 *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $x_k \in \mathbb{R}^n$, uma direção $s \in \mathbb{R}^n$ tal que $\nabla f(x_k)^t s < 0$ é chamada direção de descida para f a partir de x_k .*

Considerando uma direção de descida s_k , uma estratégia de busca linear consiste em encontrar um valor $\lambda_k \in (0, 1]$, de modo que o ponto $x_{k+1} = x_k + \lambda_k s_k$ satisfaça um decréscimo suficiente na norma de F . Um exemplo clássico desse decréscimo é a condição de Armijo [5]:

$$f(x_{k+1}) \leq f(x_k) + \alpha \nabla f(x_k)^t (x_{k+1} - x_k), \quad \text{com } \alpha \in (0, 1/2) \quad (2.4)$$

Embora o mais comum seja adotar uma estratégia que exige decréscimo monótono na norma de F , alguns autores ([20], por exemplo) têm trabalhado com a hipótese de busca linear com decréscimo não-monótono. Trabalharemos com essa última opção, utilizando a técnica proposta por Birgin, Krejić e Martínez em [2], que consiste em considerar o ponto x_k satisfatório se cumprir a condição:

$$\|F(x_k + \xi s_k)\| < (1 - \xi\sigma)\|F(x_k)\| + \mu_k \quad (2.5)$$

onde $\sigma \in (0, 1)$, $0 < \xi \leq 1$ e a sequência $\{\mu_k\}$ é tal que $\sum_{k=0}^{\infty} \mu_k < \infty$.

Algoritmo 2.2 *Busca Linear*

Suponha que estamos na k -ésima iteração:

1. Encontre a direção de descida s_k
2. Encontre θ_k que torne o ponto $x_{k+1} = x_k + \theta_k s_k$ aceitável.

A Proposição 2.1 nos garante que a direção Newton inexata é de descida e que, portanto, podemos associar a estratégia de busca linear para garantir convergência global do método Newton inexato.

Proposição 2.1 *A direção s_k que satisfaz a equação (2.2), é uma direção de descida para $f(x) = \frac{1}{2}\|F(x)\|_2^2$ a partir de x_k .*

PROVA: Considere $r_k = J(x_k)s_k + F(x_k)$. Temos então que:

$$F(x_k)^t r_k = F(x_k)^t J(x_k)s_k + F(x_k)^t F(x_k),$$

o que implica:

$$\nabla f(x_k)^t s_k = F(x_k)^t J(x_k)s_k = F(x_k)^t r_k - F(x_k)^t F(x_k).$$

E, portanto, s_k será uma direção de descida se:

$$F(x_k)^t r_k - F(x_k)^t F(x_k) < 0. \quad (2.6)$$

Fazendo o uso da desigualdade de Cauchy-Schwarz:

$$\|u^t v\|_2 \leq \|u\|_2 \|v\|_2 \quad (2.7)$$

teremos:

$$F(x_k)^t r_k \leq \|F(x_k)^t r_k\|_2 \leq \|F(x_k)\|_2 \|r_k\|_2.$$

Mas, se s_k satisfaz a inequação (2.2), usando a norma-2, temos que:

$$\|J(x_k)s_k + F(x_k)\|_2 \leq \eta_k \|F(x_k)\|_2.$$

Ou seja, $\|r_k\|_2 < \|F(x_k)\|_2$ o que nos permite concluir que:

$$F(x_k)^t r_k < \|F(x_k)\|_2^2 = F(x_k)^t F(x_k),$$

e, por (2.6), se verifica que s_k é uma direção de descida para $f(x) = \frac{1}{2}\|F(x)\|_2^2$ a partir de x_k . ■

A estratégia de adotar regiões de confiança tem o mesmo objetivo da busca linear, garantir a convergência global, sem no entanto sacrificar a convergência local. Neste caso, ao invés de fixarmos uma direção de descida e determinar o tamanho do passo (como é feito nos modelos de busca linear), criamos um modelo quadrático para determinar a direção de descida dentro da região pré-determinada pelo raio da região de confiança. Devemos, portanto, encontrar o vetor s que resolve o seguinte problema quadrático para f em torno de x_k :

$$\begin{aligned} \min m_k(x_k + s) &= f(x_k) + \nabla f(x_k)^t s + \frac{1}{2}s^t H_k s \\ \text{sujeito à } \|s\|_2 &\leq \delta_k \end{aligned} \quad (2.8)$$

onde δ_k é o raio estimado da região em que supomos poder confiar que o modelo irá se aproximar da situação original e H_k é a Hessiana de f no ponto x_k . A idéia básica da estratégia de regiões de confiança pode ser simplificada pelo seguinte algoritmo:

Algoritmo 2.3 *Regiões de Confiança*

Na k -ésima iteração faça:

1. defina δ_k ;
2. defina: $m_k(x_k + s) = f(x_k) + \nabla f(x_k)^t s + \frac{1}{2} s^t H_k s$;
3. encontre o minimizador aproximado para $m_k(x_k + s)$, para $\|s\|_2 \leq \delta_k$;
4. Considerando s_k o minimizador do item 3, faça:
 $x_{k+1} = x_k + s_k$;
5. Se x_{k+1} é um ponto aceitável, então faça $x_k = x_{k+1}$, $k = k+1$ e realize a próxima iteração, iniciando no passo 1.
Caso contrário diminua o tamanho da região de confiança e volte para o passo 3;

O resultado do Lema 2.1 a seguir é de extrema importância, pois será a base para a construção de nossa estratégia de convergência global. Uma das hipóteses para esse Lema é que H_k seja simétrica e definida positiva, embora pareça uma hipótese bastante forte, vale ressaltar que essa dificuldade será contornada quando definirmos o modelo quadrático, nesta ocasião iremos tomar os devidos cuidados para que a Hessiana do modelo satisfaça essa hipótese. Embora não seja a opção desse trabalho, pode-se trabalhar também com casos em que a matriz não cumpra esta hipótese. Sorensen em 1982 [18] propôs um Lema mais geral que contempla o caso em que a Hessiana do modelo quadrático para o problema restrito não é definida positiva. Por fim, ressaltamos que o resultado do Lema, bem como o posterior desenvolvimento desta seção está baseado no procedimento descrito em Dennis & Schnabel [5] no qual a hipótese de que H_k seja simétrica e definida positiva é essencial.

Lema 2.1 *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R} \in C^2$, $H_k \in \mathbb{R}^{n \times n}$ simétrica e definida positiva. Então: (i) o Problema (2.8) é resolvido por:*

$$s(\mu) \triangleq -(H_k + \mu I)^{-1} \nabla f(x_k)$$

para o único $\mu \geq 0$ tal que $\|s(\mu)\|_2 = \delta_k$, a menos que $\|s(0)\|_2 \leq \delta_k$, e neste caso $s(0) = s_k^N$ (o passo completo de Newton) é a solução.

(ii) Para qualquer $\mu \geq 0$, $s(\mu)$ define uma direção de descida de f em x_k .

PROVA: Ver [5], Lema 6.4.1. ■

Se obtivermos $\|s(0)\|_2 > \delta_k$ o lema acima nos leva, inicialmente, a procurar a solução do seguinte problema:

$$\Phi(\mu) = \|s(\mu)\|_2 - \delta_k \quad (2.9)$$

Isto é, a intersecção entre a curva $s(\mu)$ e a hiperesfera em $\mathbb{R}^n$ com o mesmo raio da região de confiança e centro em x_k , conforme a Figura 2.1.

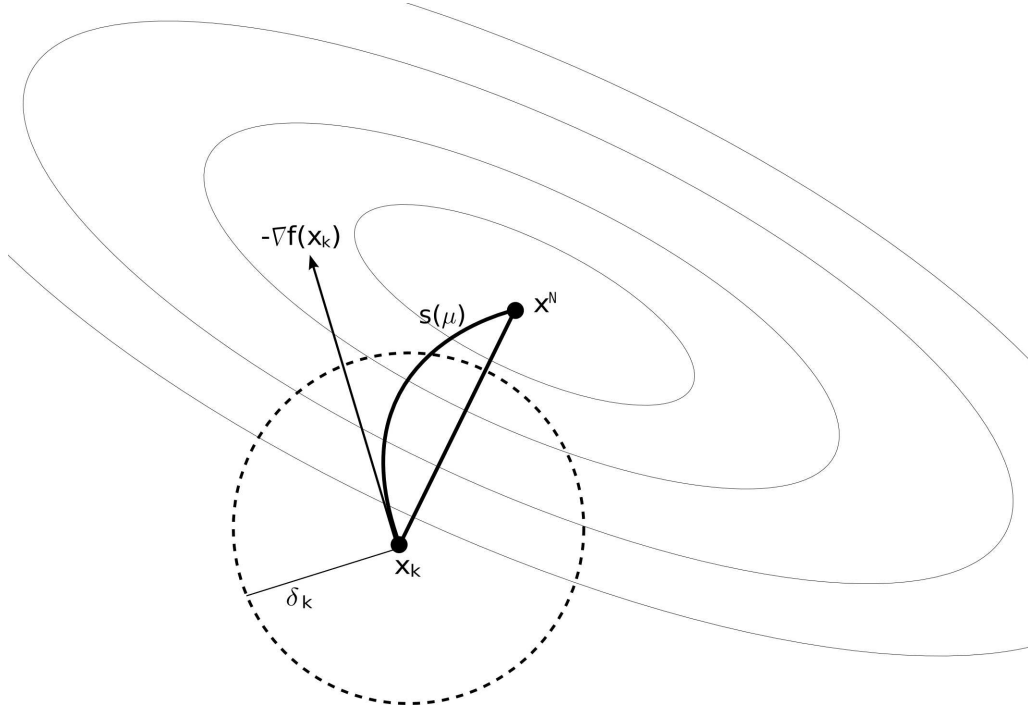


Figura 2.1: Obtendo a solução para $s(\mu)$

Uma das sugestões encontrada na literatura, chamada *Hook Step* [5] é aproximar o problema $\Phi(\mu)$ por um modelo local da forma:

$$\bar{m}_k(\mathbf{u}_k) = \frac{\mathbf{a}_k}{\mathbf{b}_k + \mathbf{u}_k} - \delta_k$$

Onde $\bar{m}$ é o novo modelo, $\mathbf{u}_k$ é a variável para iteração interna, e $\mathbf{a}_k$ e $\mathbf{b}_k$ são os parâmetros livres, que devem ser escolhidos a cada iteração interna. As variáveis denotadas em **negrito** representam as iterações internas, necessárias para encontrar a solução aproximada para o problema (2.9) enquanto as demais representam as iterações externas.

2.2.1 O Passo *Dogleg*

Assim como o *Hook Step*, esse método também encontra uma solução para o problema quadrático (2.8). Neste caso, porém, ao invés de encontrarmos um ponto $x_{k+1} = x_k + s(\mu_k)$ na curva $s(\mu)$ tal que $\|x_{k+1} - x_k\|_2 \cong \delta_k$, aproximamos a curva $s(\mu)$ por meio de uma curva linear por partes, ligando o minimizador do modelo quadrático na direção de máxima descida, aqui chamado de Ponto de Cauchy, a um ponto $x_{\hat{N}}$ na direção Newton para o problema acima (ver Figura 2.2).

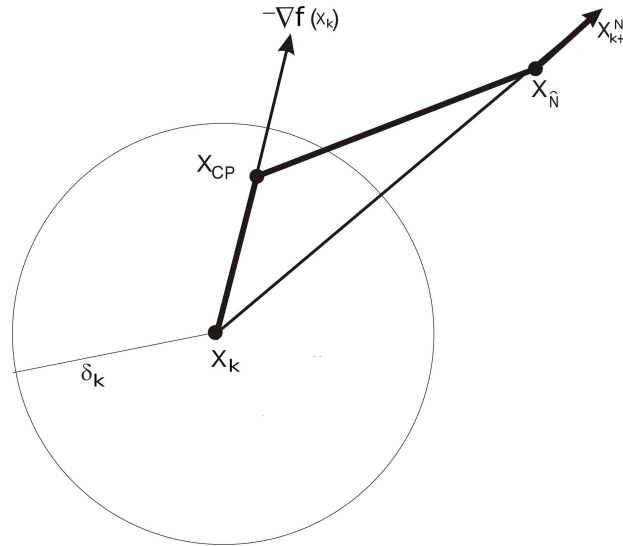


Figura 2.2: Caminho para encontrar o passo *Double Dogleg*

Conforme proposto por Powell em seu artigo original [14], o ponto $x_{\hat{N}}$ pode ser aquele gerado pelo método de Newton no modelo quadrático, a partir do ponto atual (ver Figura 2.3). Isso nos dá o chamado passo *Single Dogleg*.

Podemos também optar pelo uso do passo *Double Dogleg*, já representado na Figura 2.2, para isso consideramos um valor $\nu < 1$ adequado e fazemos $x_{\hat{N}} = \nu(x_{k+1}^N - x_k)$, onde x_{k+1}^N seria o ponto resultante da aplicação do passo completo na direção de Newton, para o modelo quadrático, a partir de x_k .

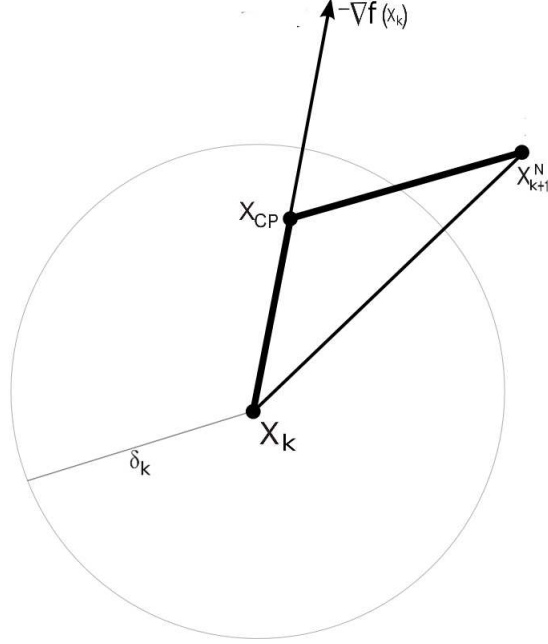


Figura 2.3: O passo *Single Dogleg* ($\nu = 1$)

A direção de descida será obtida pela intersecção entre o caminho $x_k \rightarrow x_{CP} \rightarrow x_{\hat{N}} \rightarrow x_{k+1}^N$ (ou $x_k \rightarrow x_{CP} \rightarrow x_{k+1}^N$ no caso do *Single Dogleg*) e a região dada por $\|x_{k+1} - x_k\|_2 = \delta_k$.

Como já visto no Lema 2.1, caso encontremos $\|H_k^{-1}\nabla f(x_k)\|_2 < \delta_k$, utilizaremos o passo completo do método de Newton para o modelo quadrático. Caso contrário devemos obter o Ponto de Cauchy (x_{CP}) resolvendo o problema irrestrito:

$$\min_{\lambda} m_k(x - \lambda \nabla f(x_k)),$$

que terá como solução única:

$$\lambda_* = \frac{\|\nabla f(x_k)\|_2^2}{\nabla f(x_k)^t H_k \nabla f(x_k)}. \quad (2.10)$$

Logo,

$$x_{CP} = x_k - \lambda_* \nabla f(x_k).$$

Agora, se tivermos $\delta_k \leq \|\lambda_* \nabla f(x_k)\|_2$, isto é, se o ponto de Cauchy estiver fora da região de confiança, devemos utilizar o passo na direção de máxima descida de tamanho δ_k e:

$$x_{k+1} = x_k - \frac{\delta_k}{\|\nabla f(x_k)\|_2} \nabla f(x_k).$$

Por fim, se tivermos o passo Newton fora da região de confiança e o passo de máxima descida dentro desta região, devemos encontrar o passo *Dogleg* no segmento de reta que liga o ponto x_{CP} ao ponto $x_{\hat{N}}$. Descreveremos a seguir os detalhes de como obter tal segmento.

Devemos forçar que a medida em que avançamos de x_k para x_{CP} , de x_{CP} para $x_{\hat{N}}$ e, se for o caso, de $x_{\hat{N}}$ para x_{k+1}^N (obtido com o passo completo Newton), a distância com relação a x_k aumente monotonamente. A única condição necessária para que isso ocorra, é que o ponto $x_{\hat{N}}$ seja mais distante de x_k do que x_{CP} (pois isso já ocorre de x_k para x_{CP} e de $x_{\hat{N}}$ para x_{k+1}^N). Vamos inicialmente mostrar que:

$$\|s_{CP}\|_2 = \|x_k - x_{CP}\|_2 \leq \|s_N\|_2 = \|x_k - x_{k+1}^N\|_2$$

De fato, fazendo o uso da desigualdade de Cauchy-Schwarz (2.7), teremos:

$$\begin{aligned} \|s_{CP}\|_2 &= \|\lambda_* \nabla f(x_k)\|_2 = \frac{\|\nabla f(x_k)\|_2^3}{\nabla f(x_k)^t H_k \nabla f(x_k)} \leq \\ &\leq \frac{\|\nabla f(x_k)\|_2^3}{\nabla f(x_k)^t H_k \nabla f(x_k)} \cdot \frac{\|H_k^{-1} \nabla f(x_k)\|_2 \|\nabla f(x_k)\|_2}{\nabla f(x_k)^t H_k^{-1} \nabla f(x_k)} = \\ &= \frac{\|\nabla f(x_k)\|_2^4}{(\nabla f(x_k)^t H_k \nabla f(x_k))(\nabla f(x_k)^t H_k^{-1} \nabla f(x_k))} \cdot \|s_N\|_2 \approx \gamma \|x_k - x_{k+1}^N\|_2 \end{aligned}$$

Como H_k é simétrica e definida positiva, se λ_m e λ_M são respectivamente o menor e o maior autovalor de H_k teremos:

$$\nabla f(x_k)^t H_k \nabla f(x_k) \leq \lambda_M \|\nabla f(x_k)\|_2^2 \text{ e } \nabla f(x_k)^t H_k^{-1} \nabla f(x_k) \leq \frac{1}{\lambda_m} \|\nabla f(x_k)\|_2^2.$$

Sendo assim, teremos que: $\gamma \leq \lambda_m/\lambda_M$, o que nos possibilita concluir que:

$$\|s_{CP}\|_2 \leq \gamma \|s_N\|_2 \leq \|s_N\|_2.$$

Se estivermos interessados em encontrar o passo *Single Dogleg*, basta agora que façamos o caminho x_{CP} para $x_{\hat{N}}$ pois este último ponto será equivalente ao ponto resultante do passo completo de Newton.

Se optarmos pelo *Double Dogleg* devemos encontrar o ponto $x_{\hat{N}}$ que pode ser escrito como:

$$x_{\hat{N}} = x_k - \nu H_k^{-1} \nabla f(x_k), \quad \text{com } \gamma \leq \nu \leq 1.$$

Conforme consta em [5], seria interessante, no entanto, que definíssemos o caminho $x_k \rightarrow x_{CP} \rightarrow x_{\hat{N}} \rightarrow x_{k+1}^N$ de modo que o valor do modelo m_k proposto, decresça monotonamente a medida que avançamos no caminho. Para que isso ocorra, o valor da derivada direcional em todo o segmento que liga x_{CP} a $x_{\hat{N}}$ deverá ser negativo. Parametrizamos então o segmento:

$$x_{k+1}(\lambda) = x_k + s_{CP} + \lambda(\nu s_N - s_{CP}), \quad \lambda \in [0, 1],$$

e tomamos a derivada direcional do modelo $g(\lambda) = m_k(x_{k+1}(\lambda))$:

$$\begin{aligned} g'(\lambda) &= \nabla m_k(x_{k+1}(\lambda))^t (\nu s_N - s_{CP}) = \\ &= (\nabla f(x_k) + H_k s_{CP})^t (\nu s_N - s_{CP}) + \lambda (\nu s_N - s_{CP})^t H_k (\nu s_N - s_{CP}). \end{aligned}$$

Dado que H_k é definida positiva e o lado direito da equação acima é uma função monotonamente crescente em λ , para que a derivada direcional seja negativa em $\lambda \in [0, 1]$, basta que seja negativa para $\lambda = 1$. Ou seja:

$$(\nabla f(x_k) + H_k s_{CP})^t (\nu s_N - s_{CP}) + (\nu s_N - s_{CP})^t H_k (\nu s_N - s_{CP}) < 0.$$

Obtemos que:

$$\begin{aligned} &(\nabla f(x_k) + H_k s_{CP})^t (\nu s_N - s_{CP}) + (\nu s_N - s_{CP})^t H_k (\nu s_N - s_{CP}) = \\ &= \nabla f(x_k)^t (\nu s_N - s_{CP}) + s_{CP}^t H_k (\nu s_N - s_{CP}) + \\ &+ \nu s_N^t H_k (\nu s_N - s_{CP}) - s_{CP}^t H_k (\nu s_N - s_{CP}) = \end{aligned}$$

$$\begin{aligned}\nabla f(x_k)^t(\nu s_N - s_{CP}) - \nu \nabla f(x_k)^t H_k^{-1} H_k(\nu s_N - s_{CP}) &= \\ &= (1 - \nu) \nabla f(x_k)^t(\nu s_N - s_{CP}).\end{aligned}\quad (2.11)$$

Além disso:

$$\begin{aligned}\nabla f(x_k)^t(\nu s_N - s_{CP}) &= \nabla f(x_k)^t(-\nu H_k^{-1} \nabla f(x_k) + \lambda_* \nabla f(x_k)) = \\ &= -\nu \nabla f(x_k)^t H_k^{-1} \nabla f(x_k) + \lambda_* \nabla f(x_k)^t \nabla f(x_k)\end{aligned}\quad (2.12)$$

Sabendo que:

$$\gamma = \frac{\|\nabla f(x_k)\|_2^4}{(\nabla f(x_k)^t H_k \nabla f(x_k))(\nabla f(x_k)^t H_k^{-1} \nabla f(x_k))},$$

de (2.10) teremos:

$$\gamma = \frac{\lambda_* \|\nabla f(x_k)\|_2^2}{\nabla f(x_k)^t H_k^{-1} \nabla f(x_k)}.\quad (2.13)$$

Consequentemente, podemos concluir por (2.12):

$$\nabla f(x_k)^t(\nu s_N - s_{CP}) = \gamma \frac{(\nabla f(x_k)^t H_k^{-1} \nabla f(x_k))}{\|\nabla f(x_k)\|_2^2} \|\nabla f(x_k)\|_2^2 - \nu \nabla f(x_k)^t H_k^{-1} \nabla f(x_k).$$

Daí, segue que:

$$\nabla f(x_k)^t(\nu s_N - s_{CP}) = (\gamma - \nu) \nabla f(x_k)^t H_k^{-1} \nabla f(x_k).$$

Queremos que o lado direito da equação (2.11) seja negativo, logo a seguinte desigualdade deve ser satisfeita:

$$0 > (1 - \nu)(\gamma - \nu) \nabla f(x_k)^t H_k^{-1} \nabla f(x_k),$$

para isso, já que H_k é definida positiva, basta escolher $\nu \in (\gamma, 1)$.

Seguindo as idéias aqui expostas podemos criar um algoritmo eficiente de modo que:

1. a distância com o ponto x_k aumente monotonamente a medida que avançamos no caminho criado, o que torna o algoritmo bem-definido;
2. o valor do modelo quadrático $m_k(x_k + s)$ decresça monotonamente a medida que avançamos neste caminho, o que é razoável.

Algoritmo 2.4 *Dogleg*

1. Defina s_n a direção de Newton e s_m a direção de máxima descida.
2. Se $\|s_n\|_2 \leq \delta_k$, defina $s_k = s_n$ e vá para o passo 3.
Caso contrário faça:
 - (a) Defina $s_{CP} = \lambda_* s_m$, com λ_* conforme (2.10).
 - (b) Se $\|s_{CP}\|_2 \geq \delta_k$, defina $s_k = \frac{\delta_k}{\|s_m\|_2} s_m$ e vá para o passo 3.
Caso contrário faça:
 - i. defina $s_{\hat{N}} = \nu s_n \nabla f(x_k)$, com $\gamma \leq \nu \leq 1$ e γ conforme a equação 2.13.
 - ii. encontre λ que satisfaça:
$$\|s_{CP} + \lambda(s_{\hat{N}} - s_{CP})\|_2 = \|\delta_k\|_2;$$
 - iii. defina $s_k = s_{CP} + \lambda(s_{\hat{N}} - s_{CP})$ e vá para o passo 3.
3. Faça $x_{k+1} = x_k + s_k$.

2.2.2 Alterações no raio da região de confiança

Quando utilizamos modelos, mesmo dentro de regiões de confiança, há sempre o risco de que nosso modelo não corresponda à situação real, isso exige um tanto de cautela. O que ocorre porém é que essa cautela pode ser excessiva ou então insuficiente.

Caso não sejamos suficientemente cautelosos, podemos entrar numa região em que o nosso modelo não reflete bem a situação real e o valor encontrado poderá não ser adequado para nossos objetivos.

Para evitar esse tipo de problema, devemos definir quando o valor encontrado x_{k+1} é aceitável para o problema. Dennis e Schnabel em [5], sugerem como aceitável um ponto x que satisfaça a condição de Armijo (2.4). Embora seja comum o uso de uma condição monótona, como a de Armijo, neste trabalho

optamos pelo uso da estratégia não-monótona proposta em [2] e já descrita pela condição (2.5).

Caso x_{k+1} não seja aceitável, em [5] adota-se uma estratégia de obter um novo valor entre $[0.1\delta_k, 0.9\delta_k]$. Para isso $f(x_k + \lambda(x_{k+1} - x_k))$ é aproximada por uma quadrática satisfazendo às seguintes condições: concordância com os valores de f nos pontos x_k e x_{k+1} e com a derivada direcional de f em x_k : $\nabla f(x_k)^t(x_{k+1} - x_k)$. O minimizador desse modelo ocorre em:

$$\hat{\lambda}_* = -\frac{\nabla f(x_k)^t(x_{k+1} - x_k)}{2[f(x_{k+1}) - f(x_k) - \nabla f(x_k)^t(x_{k+1} - x_k)]}$$

Escolhemos então:

- $\delta_{k+1} = \hat{\lambda}_* \|x_{k+1} - x_k\|_2$, se este valor estiver no intervalo $[0.1\delta_k, 0.9\delta_k]$;
 - $\delta_{k+1} = 0.9\delta_k$ caso contrário.
- (2.14)

Suponha que em uma dada situação encontremos o valor x_{k+1} aceitável, através de um passo diferente do passo de Newton completo. Cabe-nos pensar na hipótese de termos sido cautelosos demais e se não poderíamos ter dado um passo maior. Por exemplo, podemos supor que as iterações anteriores nos tenham feito diminuir o tamanho do passo e porventura tenhamos entrado numa região em que o modelo tenha se adequado bem à situação real. O custo para chegar à direção atual (construção do modelo, determinação do tamanho do passo e direção de descida) é um tanto alto e seria interessante avançar o quanto for possível no atual modelo.

Para decidir se devemos tentar um passo maior do que o atual, compara-se a redução real dada por:

$$\Delta f_{Ared} \triangleq f(x_{k+1}) - f(x_k)$$

com a redução prevista:

$$\Delta f_{Pred} \triangleq m(x_{k+1}) - f(x_k)$$

Caso nossa redução prevista satisfaça a seguinte condição:

$$|\Delta f_{Pred} - \Delta f_{Ared}| \leq 0.1|\Delta f_{Ared}|, \quad (2.15)$$

isso permite que tenhamos dois tipos de interpretação:

-
1. o valor do lado esquerdo da desigualdade é muito baixo, o que quer dizer que nesta região m_k representa muito adequadamente f e o valor para o raio da região de confiança, δ_k , pode ter sido subestimado;
 2. ou o valor do lado direito é alto, o que indica que a redução real em f é muito grande.

Em ambos os casos, convém armazenar os valores atuais para x_{k+1} e para $f(x_{k+1})$, e ao invés de avançarmos diretamente para a próxima iteração, dobramos o valor do raio δ_k e calculamos um novo valor para x_{k+1} usando o modelo atual. Se esse valor não for aceitável, voltamos para o último valor armazenado, caso contrário dobramos novamente o valor de δ_k e encontramos mais um novo valor para x_{k+1} . Procedemos assim sucessivamente até encontrar um valor não-satisfatório.

Para finalizar esta seção, suponha que já estamos satisfeitos com o valor para x_{k+1} e prontos para a próxima iteração. Devemos no entanto decidir o que fazer com o valor do raio de confiança para a próxima iteração. Dependendo de quão bem o modelo estiver se ajustando à função real podemos escolher como raio inicial para a próxima iteração (o valor de δ_{k+1}) o mesmo valor de δ_k , dobrar ou diminuir esse valor. Em [5], sugere-se o uso do seguinte critério:

$$\begin{aligned}
& \bullet \quad \delta_{k+1} = 2\delta_k \quad \text{se } \Delta f_{Ared} \leq 0.75\Delta f_{Pred} \\
& \bullet \quad \delta_{k+1} = 0.5\delta_k \quad \text{se } \Delta f_{Ared} \geq 0.1\Delta f_{Pred}; \\
& \bullet \quad \delta_{k+1} = \delta_k \quad \text{nos demais casos.}
\end{aligned} \tag{2.16}$$

2.3. MÉTODO GMRES

Nesta seção descreveremos o método GMRES (Generalized Minimal Residual), que é um método iterativo para a resolução de sistemas lineares

$$As = b$$

onde A é uma matriz $n \times n$, não-singular.

O método GMRES, proposto em 1986 por Saad e Schultz [17], faz parte da família de métodos de projeções sobre subespaços de Krylov. Começamos

então a seção definindo um subespaço de Krylov e apresentando algumas propriedades:

Definição 2.3 Dada uma matriz $A : n \times n$, não singular., e um vetor v compatível com esta, a sequência: $v, Av, A^2v, A^3v, \dots$ é chamada sequência de Krylov. O subespaço $\mathcal{K}_\ell(A, v) = \text{span}\{v, Av, A^2v, A^3v, \dots, A^{\ell-1}v\}$ é denominado subespaço de Krylov de ordem ℓ .

É interessante conhecer a dimensão do subespaço $\mathcal{K}_\ell(A, v)$. Com esse objetivo vamos, inicialmente, relembrar as seguintes definições:

Definição 2.4 Um polinômio do tipo $p(x) = \alpha x^n$, onde $\alpha \in \mathbb{R}$, é chamado polinômio mônico de grau n . O polinômio mônico não-nulo p de menor grau que satisfaz $p(A)v = 0$ é conhecido como polinômio mínimo do vetor v com relação à matriz A .

Proposição 2.2 Seja μ o grau do polinômio mínimo de v com respeito a A . Então $\mathcal{K}_\mu(A, v)$ é um espaço invariante em relação à matriz A e $\mathcal{K}_\ell(A, v) = \mathcal{K}_\mu(A, v)$, para todo $\ell \geq \mu$.

PROVA: Para provar que $\mathcal{K}_\mu(A, v)$ é um espaço invariante sobre A , consideremos $u \in \mathcal{K}_\mu(A, v)$, podemos escrever que $u = \alpha_1 v + \alpha_2 Av + \dots + \alpha_\mu A^{\mu-1}v$. Temos então que $Au = \alpha_1 Av + \alpha_2 A^2v + \dots + \alpha_\mu A^\mu v$. Mas $A^\mu v = 0$, logo $Au = \alpha_1 Av + \alpha_2 A^2v + \dots + \alpha_{\mu-1} A^{\mu-1}v \in \mathcal{K}_\mu(A, v)$, e portanto $\mathcal{K}_\mu(A, v)$ é invariante sobre A . Para provar a segunda parte da proposição, basta considerar que $\mathcal{K}_\ell(A, v) = \text{span}\{v, Av, \dots, A^{\ell-1}v\} = \text{span}\{v, Av, \dots, A^{\mu-1}v\}$, pois $A^k v = 0$ para todo k tal que $k \geq \mu$. ■

Corolário 2.1 Seja μ o grau do polinômio mínimo de v com relação à A . Então $\dim(\mathcal{K}_\mu(A, v)) = \mu$.

PROVA: Suponha por absurdo que $\dim(\mathcal{K}_\mu(A, v)) < \mu$. Se $v = 0$ a contradição é óbvia. Vamos então trabalhar com a hipótese de que $v \neq 0$, neste caso o conjunto $\{v, Av, A^2v, \dots, A^{\mu-1}v\}$ não seria linearmente independente e existiria $(\alpha_1, \alpha_2, \dots, \alpha_\mu) \neq 0$ tal que $\alpha_1 v + \alpha_2 Av + \dots + \alpha_\mu A^{\mu-1}v = 0$. Note que: $A^{\mu-1}(\alpha_1 v + \alpha_2 Av + \dots + \alpha_\mu A^{\mu-1}v) = 0$, logo $\alpha_1 A^{\mu-1}v = 0$ e, portanto, $\alpha_1 = 0$. Temos então que: $\alpha_2 Av + \dots + \alpha_\mu A^{\mu-1}v = 0$, multiplicando-se ambos os lados por $A^{\mu-2}$ tem-se, como anteriormente, que: $\alpha_2 A^{\mu-1}v = 0$ e $\alpha_2 = 0$. Repetindo-se sucessivas vezes esta estratégia, conclui-se que $(\alpha_1, \alpha_2, \dots, \alpha_\mu) = 0$, o que é uma contradição. ■

Proposição 2.3 *Se o grau do polinômio mínimo de v com respeito a A é μ , então $\dim(\mathcal{K}_\ell(A, v)) = \min\{\ell, \mu\}$.*

PROVA: Evidentemente se $\ell \geq \mu$ o resultado segue diretamente da Proposição 2.2 e do Corolário 2.1. Agora se $\ell < \mu$ a prova é semelhante à do Corolário 2.1. ■

Considerando s_0 a aproximação inicial, cada iteração do método GMRES consiste na minimização da norma-2 do resíduo sobre o subespaço de Krylov determinado pela matriz A e o resíduo inicial ($r_0 = b - As_0$), ou seja, se estivermos na ℓ -ésima iteração do método de GMRES, devemos encontrar o passo s_ℓ que minimiza a norma-2 do resíduo $r = As - b$ no subespaço:

$$\mathcal{K}_\ell(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{\ell-1}r_0\}.$$

O problema de minimização que devemos resolver a cada iteração pode ser escrito como:

$$\begin{aligned} \min \|b - As\|_2 \\ \text{sujeito a: } s \in s_0 + \mathcal{K}_\ell(A, r_0) \end{aligned} \quad (2.17)$$

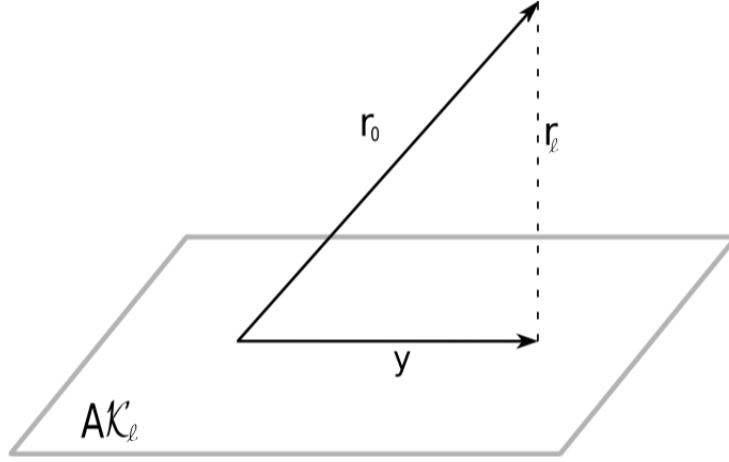


Figura 2.4: Projeção do resíduo inicial sobre o subespaço $\mathcal{AK}_\ell(A, r_0)$

O vetor s_ℓ que minimiza a norma-2 do resíduo é tal que $r_\ell = b - As_\ell$ é ortogonal ao subespaço $\mathcal{AK}_\ell(A, r_0)$. Sendo assim, teremos um problema de quadrados mínimos. Convém transformar a base atual do subespaço $\mathcal{K}_\ell(A, r_0)$

em uma base ortogonal e para isso usaremos o processo de Arnoldi, que é o processo de Gram-Schmidt para ortonormalização de uma base do subespaço de Krylov.

Algoritmo 2.5 *Processo de Arnoldi para Ortonormalização*

1. $v_1 = r_0 / (\|r_0\|_2)$;
2. Para $j = 1, 2, \dots, \ell$ faça:
 - (a) $\hat{v} = Av_j$;
 - (b) para $i = 1, 2, \dots, j$ faça:

$$\hookrightarrow h_{i,j} = \hat{v}^t v_i;$$
 - (c) $\hat{v} = \hat{v} - \sum_{i=1}^j h_{i,j} v_i$;
 - (d) $h_{j+1,j} = \|\hat{v}\|_2$;
 - (e) $v_{j+1} = \hat{v} / h_{j+1,j}$

Evidentemente pode se concluir pela Proposição 2.3 que, se o polinômio mínimo de r_0 com relação à matriz A for de grau $j < n$ não será possível construir uma base ortogonal com dimensão maior que j . Se no passo 2d do algoritmo encontrarmos $h_{j+1,j} = 0$, o algoritmo deverá ser interrompido pois o passo 2e não poderá ser realizado. A proposição abaixo garante que isso irá acontecer se, somente se, o polinômio mínimo tiver grau j .

Proposição 2.4 *O Algoritmo 2.5 falha na j -ésima iteração (isto é, $h_{j+1,j} = 0$ no passo 2d) se, e somente se, o polinômio mínimo de r_0 com respeito à matriz A tem grau j .*

PROVA: Ver [16]. ■

Supondo que o algoritmo não apresentou falha de execução, após a p -ésima iteração, teríamos obtido os vetores ortonormais $v_1, v_2, \dots, v_{\ell+1}$. Definimos então as matrizes $V_\ell : n \times \ell$ e $V_{\ell+1} : n \times (\ell+1)$, cujas colunas são dadas por esses vetores, diferenciando-se apenas pelo fato de que a matriz $V_{\ell+1} : n \times (\ell+1)$

terá uma coluna a mais (o vetor $v_{\ell+1}$). Além disso, também podemos definir a matriz $\bar{H}_\ell : (\ell + 1) \times \ell$ Hessenberg superior, onde:

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

o que nos permite construir a seguinte relação:

$$AV_\ell = V_{\ell+1}\bar{H}_\ell. \quad (2.18)$$

Pela equação (2.17) temos que $s \in s_0 + \mathcal{K}_\ell(A, r_0)$ e, considerando que as colunas de V_ℓ formam uma base para o subespaço $\mathcal{K}_\ell(A, r_0)$, existe $y \in \mathbb{R}^\ell$ tal que $s = s_0 + V_\ell y$ e podemos escrever:

$$\|b - As\|_2 = \|b - A(s_0 + V_\ell y)\|_2 = \|r_0 - AV_\ell y\|_2 = \|r_0 - V_{\ell+1}\bar{H}_\ell y\|_2.$$

Mas, pelo Algoritmo 2.5, $v_1 = r_0/\|r_0\|_2$ e, considerando-se $\beta = \|r_0\|_2$, teremos:

$$\|b - As\|_2 = \|\beta v_1 - V_{\ell+1}\bar{H}_\ell y\|_2 = \|V_{\ell+1}(\beta e_1 - \bar{H}_\ell y)\|_2 = \|\beta e_1 - \bar{H}_\ell y\|_2.$$

Podemos então reescrever o problema (2.17):

$$\begin{aligned} & \min \|\beta e_1 - \bar{H}_\ell y\|_2 \\ & \text{sujeito a } y \in \mathbb{R}^\ell, \end{aligned} \quad (2.19)$$

o que nos permite definir o seguinte algoritmo:

Algoritmo 2.6 *GMRES*

1. Defina $r_0 = b - Ax_0$, $\rho = \|r_0\|_2$, $\beta = \rho$, $v_1 = r_0/\beta$ e $k = 0$
2. Enquanto $k \leq kmax$ e $\rho > \varepsilon\|b\|_2$ faça:
 - (a) $k = k + 1$
 - (b) Executar o passo 2 do processo de Arnoldi (2.5) com $j = k$.
 - (c) Defina $e_1 = (1, 0, \dots, 0)^t \in \mathbb{R}^{k+1}$
 - (d) Encontre $y_k \in \mathbb{R}^k$ que minimiza $\|\beta e_1 - \bar{H}_k y\|_2$
 - (e) $\rho = \|\beta e_1 - \bar{H}_k y_k\|_2$
3. $x^* = x_0 + V_k y_k$

2.3.1 Rotações de Givens

Dado que a matriz $\bar{H}_\ell$ é Hessenberg superior, podemos encontrar facilmente a sua fatoração QR através de rotações de Givens, tornando a resolução do problema relativamente barata. Uma rotação de Givens de ordem 2, por exemplo, consiste na multiplicação de um vetor pela seguinte matriz:

$$G = \begin{pmatrix} c & -s \\ s & c \end{pmatrix},$$

onde $c = \cos \theta$ e $s = \sin \theta$. Essa multiplicação irá produzir uma rotação de ângulo $-\theta$ [16], com relação ao eixo x , no vetor ao qual for aplicada. O nosso interesse para aplicar as rotações de Givens é transformar a matriz de Hessenberg numa triangular superior, ou seja, anular os elementos $\bar{h}_{i+1,i}$, para $i = 1, \dots, \ell$.

Iremos fatorar a matriz de Hessenberg da seguinte maneira:

$$G_\ell \dots G_2 G_1 \bar{H}_\ell = R_\ell,$$

onde R_ℓ é a matriz triangular superior resultante e G_i é a matriz de rotação de Givens para eliminar o elemento $\bar{h}_{i+1,i}$, que deverá ter a seguinte forma:

$$G_i = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c_i & -s_i & & \\ & & & s_i & c_i & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix} \begin{array}{l} \leftarrow \text{linha } i \\ \leftarrow \text{linha } i + 1 \end{array} \quad (2.20)$$

com:

$$c_i = \frac{\bar{h}_{i+1,i}}{\sqrt{(\bar{h}_{i,i}^{(i-1)})^2 + (\bar{h}_{i+1,i})^2}} \quad \text{e} \quad s_i = \frac{\bar{h}_{i,i}^{(i-1)}}{\sqrt{(\bar{h}_{i,i}^{(i-1)})^2 + (\bar{h}_{i+1,i})^2}}. \quad (2.21)$$

Note que usamos a notação $\bar{h}_{i,i}^{(i-1)}$, isso porque esse elemento já foi alterado, $i - 1$ vezes, pela aplicação das matrizes $G_1, \dots, G_{i-1}$.

Para facilitar a resolução do problema, vamos definir $Q_\ell = G_1^t G_2^t \dots G_\ell^t$. Como G_i é ortogonal para todo i teremos Q_ℓ também ortogonal, logo:

$$\|b - As\|_2 = \|\beta e_1 - \bar{H}_\ell y\|_2 = \|\beta Q_\ell e_1 - R_\ell y\|_2. \quad (2.22)$$

Para compreender as vantagens proporcionadas pelas rotações de Givens, convém observar os resultados do seguinte teorema:

Teorema 2.4 *Considere $R_\ell = Q_\ell \bar{H}_\ell$ como definido 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 $\hat{g}_\ell$ um vetor de dimensão ℓ encontrado eliminando-se o elemento $g_{\ell+1,1}$ (a última linha) de g . Então podemos dizer que:*

1. *o posto de AV_ℓ é igual ao posto de $\hat{R}_\ell$. Em particular, se $r_{\ell,\ell} = 0$ então A deve ser singular;*
2. *o vetor y^* que é o minimizador da equação $\|\beta e_1 - \bar{H}_\ell y\|_2$ é dado por*

$$y^* = \hat{R}_\ell^{-1} \hat{g};$$

3. *a norma-2 do resíduo no passo ℓ será dada por $g_{\ell+1,1}$.*

PROVA: Para provar a primeira parte considere que:

$$AV_\ell = V_{\ell+1} Q_\ell^t Q_\ell \bar{H}_\ell \Rightarrow AV_\ell = V_{\ell+1} Q_\ell^t R_\ell.$$

Visto que tanto Q_ℓ quanto $V_{\ell+1}$ são unitárias, temos que posto de AV_ℓ é igual ao posto de R_ℓ , que é igual ao posto de $\hat{R}_\ell$, já que a última linha de R_ℓ é nula. Como R_ℓ é triangular superior, se $r_{\ell,\ell} = 0$ então R_ℓ é singular, e portanto A é singular. A segunda e a terceira parte saem da equação (2.22). Pois:

$$\|\beta e_1 - \bar{H}_\ell y\|_2 = \|\beta Q_\ell e_1 - R_\ell y\|_2 = |g_{\ell+1,1}| + \|\hat{g} - \hat{R}_\ell y\|_2.$$

O resíduo mínimo é encontrado zerando o segundo elemento da parte esquerda da equação acima. Logo $y^* = \hat{R}_\ell^{-1} \hat{g}$ e a norma-2 do resíduo será dada por $g_{\ell+1,1}$. ■

Esse resultado é um fator estimulante para a implementação pois, garantida a não-singularidade da matriz A , a aproximação para a solução do sistema linear não precisaria ser obtida a cada iteração do GMRES, mas somente quando tivermos um resíduo suficientemente pequeno, já que a norma-2 do resíduo será dada pelo módulo do último elemento do vetor $g = \beta Q_\ell e_1$. Ou seja, podemos reescrever o Algoritmo 2.6:

Algoritmo 2.7 *GMRES com Rotações de Givens*

1. Defina $r_0 = b - Ax_0$, $\rho = \|r_0\|_2$, $\beta = \rho$, $v_1 = r_0/\beta$, $Q = I : n \times n$ e $k = 0$.
2. Enquanto $k \leq kmax$ e $\rho > \varepsilon \|b\|_2$ faça:
 - (a) $k = k + 1$;
 - (b) executar o passo 2 do processo de Arnoldi (2.5) com $j = k$;
 - (c) se $k > 1$, aplique Q_k na k -ésima coluna de $\bar{H}_k$;
 - (d) faça: $c_k = \frac{\bar{h}_{k+1,k}}{\sqrt{(\bar{h}_{k,k})^2 + (\bar{h}_{k+1,k})^2}}$ e $s_k = \frac{\bar{h}_{k,k}}{\sqrt{(\bar{h}_{k,k})^2 + (\bar{h}_{k+1,k})^2}}$;
 - (e) $\bar{h}_{k,k} = c_k \bar{h}_{k,k} - s_k \bar{h}_{k+1,k}$ e $\bar{h}_{k+1,k} = 0$;
 - (f) defina G_k como em (2.20) e faça $g = G_k g$;
 - (g) faça $Q = G_k Q$ e defina Q_k a matriz quadrada correspondente às k primeiras linhas de Q ;
 - (h) $\rho = |g_{k+1}|$.
3. Considere R a matriz quadrada correspondente às k primeiras linhas de $\bar{H}$ e w o vetor correspondente às k primeiras linhas de g .
4. Encontre y_k que resolve o sistema: $Ry = w$.
5. $x^* = x_0 + V_k y_k$.

O Algoritmo 2.7 pode apresentar falha no passo 2b. Conforme a Proposição 2.4, isso acontecerá se, e somente se, o polinômio mínimo de r_0 em relação à matriz A tiver grau $c < n$. A proposição abaixo garante que, neste caso, teremos encontrado a solução exata do sistema, é o que chamam na literatura de *lucky breakdown*.

Proposição 2.5 *Seja A uma matriz não-singular, então o Algoritmo 2.7 irá falhar na ℓ -ésima iteração (isto é, $h_{\ell+1,\ell} = 0$) se, e somente se, s_k é a solução exata do sistema $As = b$.*

PROVA: Considere que $h_{\ell+1,\ell} = 0$. Da não-singularidade de A temos, pela primeira parte do Teorema 2.4 que $\hat{R}_{\ell,\ell} = h_{\ell,\ell}^{\ell-1} \neq 0$. Sendo assim por (2.21) temos que $s_\ell = 0$. O vetor g atual é obtido através do produto da matriz G_ℓ pelo vetor g anterior aumentado em uma dimensão. Chamando de $\bar{g} = (\bar{g}_1, \dots, \bar{g}_\ell, 0)^t$ este vetor g anterior, temos que $g = (\bar{g}_1, \dots, c_\ell \bar{g}_\ell, s_\ell \bar{g}_\ell)$. Pela última parte do Teorema 2.4, tem-se que $\|r_\ell\|_2 = |g_{\ell+1}| = |s_\ell \bar{g}_\ell| = 0$.

Para provar que a condição é suficiente, devemos notar que se a solução exata é encontrada no passo ℓ e não no passo $\ell - 1$ vamos ter $\|r_\ell\|_2 = 0$ e então $|g_{\ell+1}| = |s_\ell \bar{g}_\ell| = 0$ o que implica que $s_\ell = 0$ e, por (2.21), $h_{\ell+1,\ell} = 0$. ■

Esse resultado irá facilitar a demonstração do teorema abaixo que garante a convergência do Algoritmo 2.6:

Teorema 2.5 *Seja A uma matriz $n \times n$ não-singular. Então o Algoritmo 2.6 encontrará a solução para o problema linear $As = b$ em no máximo n iterações.*

PROVA: Ver [10] ■

2.3.2 GMRES com Recomeços

Para problemas de grande porte, torna-se complicado o uso do Método GMRES, pois à medida que o valor de ℓ aumenta, aumentam cada vez mais os requerimentos de memória e o custo computacional. Uma alternativa que vem sendo utilizada é o método GMRES com recomeços. Esta variação consiste em interromper o ciclo do GMRES a cada m iterações (daí o nome GMRES (m)). O Algoritmo 2.8 define basicamente como funciona esse processo:

Algoritmo 2.8 *GMRES*(m)

1. $k = 0$.
2. Aplique o método GMRES (Algoritmo 2.7) com $k_{\max}=m$.
3. $\ell = 1$.
4. Enquanto $\ell \leq \text{máximo de ciclos}$ e $\rho > \varepsilon \|b\|_2$ faça:
 - (a) $\ell = \ell + 1$;
 - (b) $k = 0$;
 - (c) aplique o método GMRES (Algoritmo 2.7) com $k_{\max}=m$.

Devemos observar que, apesar dessa estratégia funcionar bem na prática, não se pode garantir a convergência do método, pois apesar da norma-2 do resíduo, no *GMRES*(m), não ser crescente, conforme [16], este valor pode sofrer uma estagnação em casos em que a matriz não é positiva-definida. Neste caso, a taxa de convergência do método *GMRES*(m) pode ser ruim. Uma escolha adequada para o valor de m pode solucionar este tipo de problema e é tema de estudo de vários trabalhos.

O MÉTODO NEWTON-GMRES(M) COM USO DE ESTRATÉGIA MATRIX-FREE

No capítulo anterior descrevemos a estratégia de globalização *Dogleg* e o método GMRES que serão empregados na implementação do algoritmo Newton-inexato para resolução de sistemas não-lineares. Neste capítulo iremos apresentar os detalhes que justificam o uso conjunto do método GMRES (para resolução do sistema linear) com a estratégia *Double Dogleg* (para a globalização). Para facilitar a localização dentro da leitura, apresentamos na Figura 3.1 um escopo das principais etapas do algoritmo básico final.

Observação 3.1: Chamamos este algoritmo de básico final, apesar do escopo apresentado corresponder a um dos algoritmos testados, os outros seguem essa estrutura, com poucas modificações.

Nesse trabalho usaremos o GMRES para encontrar a direção de descida para o método Newton-inexato. Considerando que devemos resolver o sistema linear (denotemos por J a matriz Jacobiana de F no ponto x_k):

$$Js = -F(x_k) \quad (3.1)$$

e, supondo s_0 o chute inicial para a obtenção da direção Newton-inexata, temos $r_0 = Js_0 + F(x_k)$ o resíduo inicial e construiremos os subespaços de Krylov utilizando os vetores:

$$r_0, Jr_0, J^2r_0, J^3r_0 \dots$$

Como a matriz Jacobiana é não-singular, os métodos de subespaços de Krylov (e portanto o GMRES) teoricamente encontrariam a solução exata do sistema

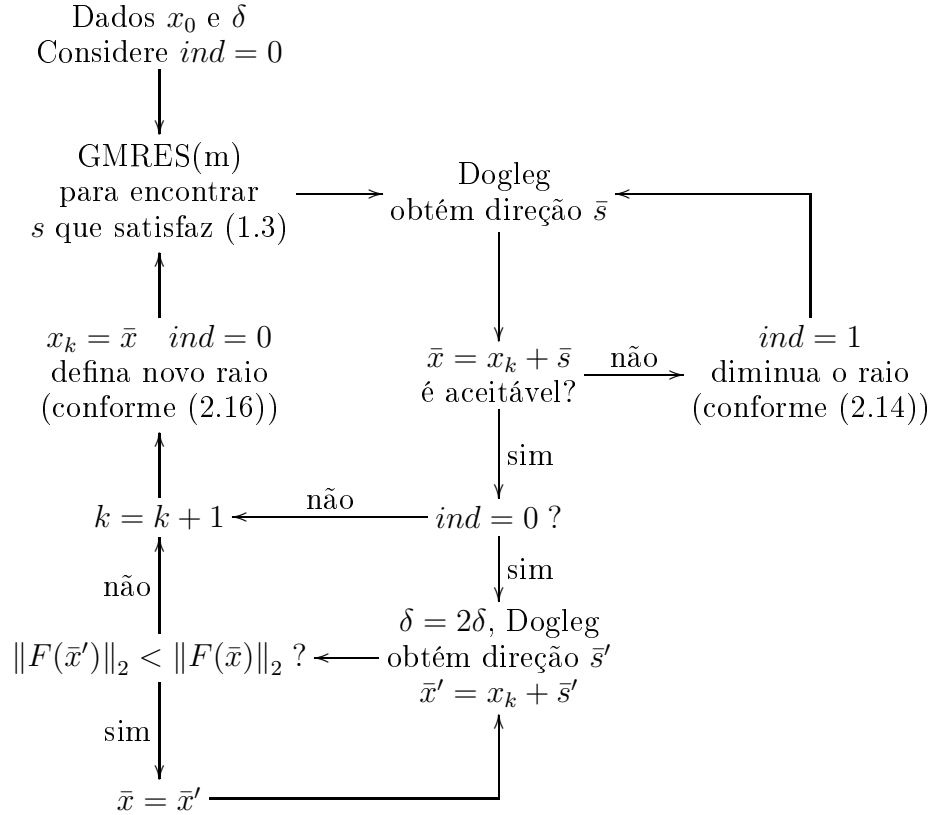


Figura 3.1: Escopo do algoritmo básico final

linear (3.1) em no máximo n iterações, e se assim fizéssemos, obteríamos o vetor s_k que seria a solução exata do sistema, ou seja o passo exato de Newton. No caso de problemas de grande porte, se exigirmos que todos os n passos do GMRES sejam realizados, necessitaremos de altos requerimentos de memória e o programa se tornará ineficiente. Portanto, iremos optar pelo método de Newton-inexato e realizar as iterações do GMRES somente até encontrar o valor de p para qual a direção s que minimiza a norma-2 do resíduo no espaço $\mathcal{K}_p$ satisfaça a condição (2.2).

O método GMRES possibilita o uso da técnica *matrix-free*, que permite que o produto entre a matriz Jacobiana e um vetor compatível seja realizado por meio de uma aproximação, o que evita o cálculo das derivadas parciais

de F e reduz o número de requerimentos de memória. Isso é possível pois, nos métodos Newton-Krylov, o uso do Jacobiano só é necessário no produto deste por algum vetor. Um valor aproximado para o resultado desse produto pode ser obtido, através de aproximação por expansão de série de Taylor:

$$J(x_k)v \approx \frac{F(x_k + hv) - F(x_k)}{h}, \quad h \in \mathbb{R}. \quad (3.2)$$

Um detalhe de extrema importância é a determinação de h para a aproximação acima. Dennis e Schnabel [5] sugerem o uso do seguinte valor para h :

$$h = \frac{1}{\|v\|_2} \sqrt{\varepsilon} \max\{|(x_k)^t v|, \text{tip} x_k^t |v|\} \text{sign}(x_k^t v), \quad (3.3)$$

onde ε é o erro da máquina, $\text{tip}(x_k)$ é um vetor composto por valores típicos para x_k e $\text{sign}(u)$ é uma função que retorna 1 caso u seja positivo e -1 caso u seja negativo.

Bellavia e Morini em [1], propõem uma alternativa mais simples:

$$h = \sqrt{\varepsilon} \frac{\|x_k\|_2}{\|v\|_2}, \quad (3.4)$$

onde ε é novamente a precisão da máquina.

Durante os testes numérico comparamos o desempenho entre esses dois valores para h , dos quais observamos um desempenho superior dessa última escolha.

3.1. *Dogleg* E GLOBALIZAÇÃO

Nesta seção será visto como relacionar a estratégia de obtenção do passo Newton-Inexato (através do GMRES) com a estratégia de globalização pelo método *Double Dogleg* minimizando-se o esforço computacional. O uso conjunto das duas estratégias é baseado na proposta de Brown e Saad [3], de 1990. Usaremos como notação s_N para o passo de Newton no modelo quadrático e s_M a direção de máxima descida também no modelo quadrático, além disso denotaremos $F(x_k)$, simplesmente por F e $J(x_k)$ por J .

Quando o GMRES termina obtemos uma direção Newton-Inexata do tipo $s = s_0 + V_p y$, onde p representa o número de iterações do GMRES, s_0 a aproximação inicial para o passo GMRES e V_p a matriz ortogonal cujas colunas geram o subespaço $\mathcal{K}_p$. Por motivos que ficarão mais claros, iremos considerar $s_0 = 0$ e teremos, portanto, $s = V_p y$.

A estratégia de globalização procurará encontrar um mínimo local para a função de mérito:

$$f(x) = \frac{1}{2} F(x)^t F(x)$$

Portanto, devemos resolver o seguinte problema não-linear:

$$\begin{aligned} \min g(y) &= f(x_k + V_p y) \\ \text{sujeito a } y &\in \mathbb{R}^p \end{aligned} \quad (3.5)$$

Se estamos usando o modelo linear $F + Js$ para aproximar $F(x_k + s)$, uma aproximação quadrática natural para $g(y)$ seria $\frac{1}{2} \|F + Js\|_2^2$, que podemos escrever:

$$\begin{aligned} \hat{g}(y) &= \frac{1}{2} \|F + JV_p y\|_2^2 = \\ &= \frac{1}{2} F^t F + F^t JV_p y + \frac{1}{2} y^t (JV_p)^t (JV_p) y \end{aligned} \quad (3.6)$$

Note que $f(x_k) = g(0) = \hat{g}(0)$.

Considerando B a matriz $p \times p$, simétrica e definida positiva dada por:

$$B = (JV_p)^t (JV_p) = (V_{p+1} \bar{H}_p)^t (V_{p+1} \bar{H}_p) = \bar{H}_p^t \bar{H}_p,$$

a estratégia de regiões de confiança deverá resolver o seguinte problema de minimização:

$$\begin{aligned} \min \hat{g}(y) &= \frac{1}{2} F^t F + F^t JV_p y + \frac{1}{2} y^t B y \\ \text{sujeito a } &\begin{cases} \|y\|_2 \leq \delta_k \\ y \in \mathbb{R}^p \end{cases} \end{aligned} \quad (3.7)$$

A direção de máxima descida a partir de x_k para o modelo (3.7) será dada por:

$$\begin{aligned} s_M &= -\nabla \hat{g}(0) = -(JV_p)^t F = -(V_{p+1} \bar{H}_p)^t F \\ &= -\bar{H}_p^t V_{p+1}^t F = \bar{H}_p^t (\beta e_1) \\ &= \beta \bar{H}_p^t e_1. \end{aligned} \quad (3.8)$$

Como a Hessiana do modelo quadrático é dada por B e $s_M = -\nabla \hat{g}(0)$, teremos a direção de Newton para o modelo quadrático dada por:

$$s_N = B^{-1}s_M = (\bar{H}_p^t \bar{H}_p)^{-1}(\beta \bar{H}_p^t e_1). \quad (3.9)$$

Portanto, s_N é a solução do sistema:

$$\bar{H}_p^t \bar{H}_p y = \beta \bar{H}_p^t e_1,$$

e, conseqüentemente, é a solução de quadrados mínimos para o sistema sobredeterminado:

$$\bar{H}_p y = \beta e_1.$$

Então, encontrar a direção de Newton para o modelo quadrático é equivalente resolver o seguinte problema:

$$\min \|\bar{H}_p y - \beta e_1\|_2, \quad \text{com } y \in \mathbb{R}^p.$$

Mas esse problema já foi resolvido pelo método GMRES (ver problema (2.19)), e portanto basta usar a direção de Newton-Inexato já encontrada como a direção de Newton para o modelo quadrático.

Devemos agora encontrar o ponto de Cauchy para o problema (3.6). Lembremos que o ponto de Cauchy será dado pelo minimizador do modelo quadrático na direção de máxima descida.

Desde que,

$$\hat{g}(\lambda s_M) = \frac{1}{2} F^t F + \lambda F^t J V_p s_M + \frac{\lambda^2}{2} s_M^t (J V_p)^t (J V_p) s_M, \quad (3.10)$$

teremos:

$$\nabla \hat{g}(\lambda s_M) = F^t J V_p s_M + \lambda (J V_p)^t (J V_p) s_M.$$

Pela equação (3.8), pode-se concluir que o valor de λ que minimiza a função (3.10) é dado por:

$$\lambda_* = \frac{-F^t J V_p s_M}{s_M^t (J V_p)^t (J V_p) s_M} = \frac{-F^t J V_p s_M}{s_M^t (V_{m+1} \bar{H}_p)^t (V_{m+1} \bar{H}_p) s_M} = \frac{\|s_M\|_2^2}{\|\bar{H}_p s_M\|_2^2}.$$

Teremos então a direção de Cauchy (CP):

$$s_{CP} = \lambda^* s_M = \frac{\|s_M\|_2^2}{\|\bar{H}_p s_M\|_2^2} s_M = \frac{\|s_M\|_2^2}{\|\bar{H}_p s_M\|_2^2} (\beta \bar{H}_p e_1).$$

Vamos definir agora, a direção $s_{\hat{N}}$. No caso do *Single Dogleg*, ele será dada pela direção Newton-Inexata retornada pelo GMRES. Caso optemos pelo passo *Double Dogleg* teremos: $s_{\hat{N}} = \nu s_N$, onde s_N é a direção Newton-Inexata vinda do GMRES. Em [5] sugere-se que o valor de ν seja dado por: $\nu = 0.8\gamma + 0.2$. Da equação (2.13), vamos ter que:

$$\gamma = \frac{\lambda_* \|\nabla \hat{g}(0)\|_2^2}{\nabla \hat{g}(0)^t (\bar{H}_p^t \bar{H}_p)^{-1} \nabla \hat{g}(0)} \equiv \frac{\lambda_* \|s_M\|_2^2}{-s_M^t s_N}.$$

Após encontrar o passo *Dogleg* (denotemos-o por y), devemos verificar se o ponto encontrado: $x_{k+1} = x_k + V_p y$ é satisfatório. Uma das opções é analisar se o ponto cumpre a condição de Armijo:

$$f(x_{k+1}) \leq f(x_k) + \alpha \nabla f(x_k)^t (x_{k+1} - x_k).$$

Neste caso temos que:

$$\nabla f(x_k)^t (x_{k+1} - x_k) = \nabla f(x_k)^t s_k = F^t J V_p y = F^t V_{m+1} \bar{H}_p y = \beta e_1^t \bar{H}_p y = s_M^t y.$$

E portanto, podemos escrever a condição de Armijo como:

$$\frac{1}{2} \|F(x_{k+1})\|_2^2 \leq \frac{1}{2} \|F(x_k)\|_2^2 + \alpha s_M^t y.$$

Neste trabalho também optamos pelo uso da condição não-monótona proposta em [2]. Conforme a condição (2.5), não teremos qualquer dificuldade de adaptação nesse caso.

Se o ponto encontrado não for satisfatório, devemos reduzir o raio da região de confiança. Como descrito no capítulo anterior (ver (2.16)), podemos encontrar o novo raio da região de confiança da seguinte maneira:

$$\begin{aligned}
\bar{\lambda}_* &= -\frac{\nabla f(x_k)^t(x_{k+1} - x_k)}{2[f(x_{k+1}) - f(x_k) - \nabla f(x_k)^t(x_{k+1} - x_k)]} = \\
&= \frac{s_M^t y}{\|F(x_{k+1})\|_2^2 - \|F(x_k)\|_2^2 + 2s_M^t y}.
\end{aligned}$$

Encontrado esse valor verificamos se $\bar{\lambda}_*\|x_{k+1} - x_k\|_2 \in [0.1\delta_k, 0.9\delta_k]$, caso positivo teremos $\delta_{k+1} = \bar{\lambda}_*\|x_{k+1} - x_k\|_2$, caso negativo $\delta_{k+1} = 0.9\delta_k$.

Para analisar a possibilidade de um passo maior com o mesmo modelo, bem como o valor do raio para a próxima iteração, precisamos comparar a redução real com a redução prevista, que serão dadas respectivamente por:

$$\Delta f_{Ared} = f(x_{k+1}) - f(x_k) = 1/2(\|F(x_{k+1})\|_2^2 - \|F(x_k)\|_2^2)$$

$$\begin{aligned}
\Delta f_{Pred} &= \hat{g}(y) - g(0) = \\
&= \frac{1}{2}F^t F + F^t J V_p y + \frac{1}{2}y^t B y - \frac{1}{2}F^t F = \\
&= \frac{1}{2}\|\bar{H}_p y\|_2^2 - s_M^t y.
\end{aligned}$$

3.2. MODIFICAÇÕES NECESSÁRIAS PARA O PROCESSO DE RECOMEÇOS

Com a adição do processo de recomenços, a aproximação inicial para o GMRES não seria o vetor nulo a partir do segundo ciclo. Com isso o problema de minimização em cada ciclo do GMRES seria um pouco diferente do problema (3.5) e teria a seguinte forma:

$$\begin{aligned}
&\min f(x_k + s_0 + V_p y) \\
&\text{sujeito a } y \in \mathbb{R}^p.
\end{aligned}$$

Entretanto, com esta escolha, a função $g(y)$ definida na seção anterior, na maioria dos casos, não estaria perto de $f(x_k)$ para y pequeno (ou seja, $g(0) \neq f(x_k)$). Para sanar essa deficiência, devemos reescrever o problema, aumentando uma dimensão. Suponhamos que cada ciclo do GMRES permita no máximo m iterações. Seja então $\bar{y} = [y, t]^t \in \mathbb{R}^{p+1}$ e $W = [V_p, s_0]$ e encontramos um passo s da forma $s = Wy$ e a função g modificada: $\bar{g}(\bar{y}) = f(x_k + W\bar{y})$ (verifique que $\bar{g}(0) = f(x_k)$). O modelo quadrático será dado por:

$$\hat{g}(\bar{y}) = \frac{1}{2} \|F + JW\bar{y}\|_2^2.$$

Dado que $r_0 = -F - Js_0$, teremos:

$$\begin{aligned} F + JW\bar{y} &= F + [V_p \ s_0] \begin{bmatrix} y \\ t \end{bmatrix} = F + [V_p y \ ts_0] = F + JV_p y + tJs_0 = \\ &= F + JV_p y + (-r_0 - F)t = (1-t)F - tr_0 + V_{p+1}\bar{H}_p y, \end{aligned}$$

o que nos permitirá que a função $\hat{g}$ seja avaliada sem o uso da matriz Jacobiana.

Encontraremos, então, a direção de máxima descida para o modelo quadrático modificado:

$$\begin{aligned} s_M &= -\nabla \hat{g}(0) = -(JW)^t F = -[JV_p \ Js_0]^t F = \\ &= -[V_{p+1}\bar{H}_p \ Js_0]^t F = \begin{pmatrix} -\bar{H}_p^t V_{p+1}^t F \\ (F + r_0)^t F \end{pmatrix} \in \mathbb{R}^{p+1}. \end{aligned}$$

O passo de Newton para o modelo quadrático, assim como no caso sem recomeços, será obtido relacionando os dados obtidos no processo GMRES e o passo de máxima descida já encontrado. Mas, diferentemente do caso anterior, a direção de Newton para o modelo quadrático não poderá ser aproximada pela direção Newton-inexata do GMRES pois $s_0 \neq 0$. Considere inicialmente que:

$$s_N = B^{-1} s_M.$$

Aqui, novamente, B é a Hessiana do modelo quadrático ($B = \nabla^2 \hat{g}(0)$). Considerando então $v = -\bar{H}_p^t V_{p+1}^t (F + r_0)$ e $a = \|F + r_0\|_2^2$ podemos reescrever a matriz B :

$$B = (JW)^t (JW) = \begin{pmatrix} \bar{H}_p^t \bar{H}_p & v \\ v^t & a \end{pmatrix}.$$

Para a implementação é conveniente lembrar que o método GMRES obteve, através das rotações de Givens, a fatoração QR da matriz $\bar{H}_p$. Logo $\bar{H}_p^t \bar{H}_p = (QR)^t(QR) = R^t R$. Como a dimensão de $\bar{H}_p$ é $(p+1) \times p$, de acordo com o processo de fatoração QR, a matriz $R : (p+1) \times p$ terá a seguinte forma:

$$R = \begin{pmatrix} \bar{R} \\ 0 \end{pmatrix},$$

onde $\bar{R} \in \mathbb{R}^{p \times p}$, e ainda: $\bar{R}^t \bar{R} = R^t R$. Consideremos então $w = \bar{R}^{-1}v$ e $b = \sqrt{a - w^t w}$, e obtemos facilmente a decomposição de Cholesky para B :

$$B = \begin{pmatrix} \bar{R}^t \bar{R} & v \\ v^t & a \end{pmatrix} = \begin{pmatrix} \bar{R}^t & 0 \\ w^t & b \end{pmatrix} \begin{pmatrix} \bar{R} & w \\ 0 & b \end{pmatrix} \equiv C^t C.$$

Para realizar o passo *Dogleg*, precisaremos encontrar o ponto de Cauchy que, como já vimos anteriormente, é o ponto minimizador para o modelo quadrático na direção de máxima descida (s_M):

$$\begin{aligned} \hat{g}(\bar{y}) &= \frac{1}{2} \|F + JW\bar{y}\|_2^2 = 1/2 F^t F + F^t JW\bar{y} + \bar{y}^t (JW)^t (JW)\bar{y} = \\ &= 1/2 F^t F + F^t JW\bar{y} + \bar{y}^t B\bar{y} = \frac{1}{2} F^t F + F^t JW\bar{y} + \bar{y}^t C^t C\bar{y}, \end{aligned}$$

e, para $\bar{y} = \lambda s_M$, teremos:

$$\hat{g}(\lambda s_M) = \frac{1}{2} F^t F + \lambda F^t JW s_M + \frac{\lambda^2}{2} s_M^t C^t C s_M$$

Portanto, o valor de λ minimizador será dado por:

$$\lambda^* = \frac{-F^t JW s_M}{s_M^t C^t C s_M} = \frac{s_M^t s_M}{s_M^t C^t C s_M} = \frac{\|s_M\|_2^2}{\|C s_M\|_2^2}.$$

Para encontrar o ponto $x_{\hat{N}}$ devemos encontrar o valor para γ segundo a equação (2.13):

$$\gamma = \frac{\lambda^* \|s_M\|_2^2}{-s_M^t s_N}.$$

Precisamos agora criar a nova condição de aceitação do ponto encontrado. Antes disso, verifique que:

$$\begin{aligned} \nabla f(x_k)^t s &= F^t J s = F^t J(s_0 + V_p y) = F^t J V_p y + F^t J s_0 = \\ &= F^t V_{p+1} \bar{H}_p y - (F + r_0)^t F \end{aligned},$$

e, daí:

$$\nabla f(x_k)^t s = -s_p^t \begin{pmatrix} y \\ 1 \end{pmatrix}.$$

Sendo assim, reescrevemos a condição de Armijo como:

$$\frac{1}{2} \|F(x_{k+1})\|_2^2 \leq \frac{1}{2} \|F(x_k)\|_2^2 - \alpha s_M^t \begin{pmatrix} y \\ 1 \end{pmatrix}.$$

Em caso de insatisfação, o novo raio da região de confiança será dado por:

$$\delta_{k+1}^* = \frac{s_M^t \begin{pmatrix} y \\ 1 \end{pmatrix}}{\|F(x_{k+1})\|_2^2 - \|F(x_k)\|_2^2 + 2s_M^t \begin{pmatrix} y \\ 1 \end{pmatrix}} \|x_{k+1} - x_k\|_2$$

se δ_{k+1}^* pertencer ao intervalo $[0.1\delta_k, 0.9\delta_k]$, caso contrário temos que $\delta_{k+1} = 0.9\delta_k$.

Observação 3.2: Se optarmos pelo uso da condição (2.5) ao invés da condição de Armijo, assim como no caso sem recomeços nenhuma adaptação se fará necessária.

O valor de Δf_{Ared} , evidentemente, pode ser calculado como no caso sem recomeços, mas Δf_{Pred} se altera, sendo obtido por:

$$\begin{aligned} \Delta f_{Pred} &= \hat{g}(\bar{y}) - g(0) = \frac{1}{2} F^t F + F^t J W \bar{y} + \frac{1}{2} \bar{y}^t (J W)^t J W \bar{y} - \frac{1}{2} F^t F = \\ &= \frac{1}{2} \|C \bar{y}\|_2^2 - s_M^t \bar{y}. \end{aligned}$$

4.1. ALGORITMOS TESTADOS

As idéias apresentadas nos capítulos anteriores nos possibilitou a construção de uma série de algoritmos para a resolução de sistemas não-lineares através do método Newton-GMRES. O primeiro algoritmo construído consiste num método Newton-GMRES com estratégia de globalização exclusivamente baseada em regiões de confiança com o uso do método *Double Dogleg*, conforme descrito na Figura 4.1.

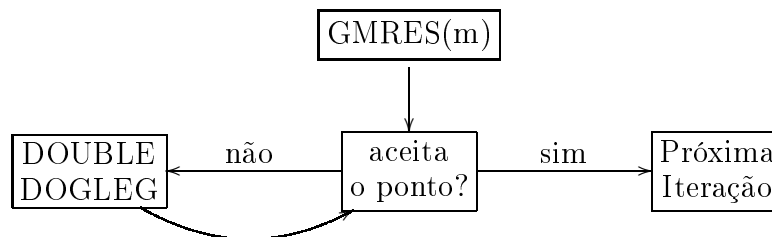


Figura 4.1: Escopo do programa com globalização baseada em *Double Dogleg*

O segundo algoritmo desenvolvido utiliza uma estratégia híbrida com uso inicial de busca linear (três tentativas) e caso não obtenha sucesso (isto é, caso não encontre um ponto aceitável), apela-se para a estratégia *Double Dogleg* até obter sucesso. Uma simplificação das etapas desse procedimento pode ser conferida na Figura 4.2.

Para ambos os algoritmos utilizamos como critério de aceitação, inicialmente, o critério de Armijo (2.4) e também o critério não-monótono (2.5) proposto em [2]. Já nos testes iniciais o desempenho dos algoritmos com o critério de Armijo foi inferior, sendo decidido adotar exclusivamente o critério (2.5) nos testes posteriores.

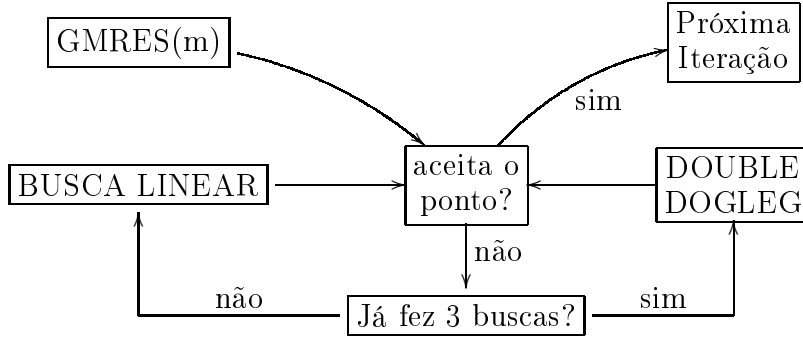


Figura 4.2: Escopo do programa com globalização híbrida

Implementamos ainda um último algoritmo, com uma pequena variação em relação a este segundo algoritmo, onde muda-se somente o critério de aceitação para os pontos advindos da estratégia *Double Dogleg*. Diferentemente dos anteriores, o critério para este algoritmo é baseado em dados de redução prevista e redução real. Note que, essa mudança só valerá para o teste após o *Double Dogleg*. No caso da busca linear o critério de aceitação continua como nos algoritmos anteriores. Sendo assim, em conformidade com a equação (2.15) propomos que fosse aceito o ponto x_k que proporcionasse a satisfação da seguinte desigualdade:

$$|\Delta f_{Pred} - \Delta f_{Ared}| \leq 0.1 |\Delta f_{Ared}|. \quad (4.1)$$

4.1.1 Detalhamento da implementação

Para a resolução de sistemas lineares através do GMRES, utilizamos uma modificação do algoritmo implementado por Kelley [10] e encontrado em:

<http://www.siam.org/books/kelley/fr16/matlabcode.php>.

Justifica-se o uso, pois ao final do algoritmo, além do valor de s_k encontrado e do número de iterações internas, precisamos também de outros dados que serão utilizados para o processo de globalização *Double Dogleg* (como por exemplo as matrizes V e H). Esse algoritmo preserva boa parte desses dados, e com pequenas alterações conseguimos o que precisamos, ao contrário do que ocorre com o algoritmo padrão disponível no Matlab®.

O termo forçante para o método de Newton-Inexato foi definido de acordo com a proposta de Eisenstat e Walker em [8]:

$$\eta_k = \gamma \left(\frac{\|F(x_k)\|_2}{\|F(x_{k-1})\|_2} \right)^\alpha.$$

Neste caso, utilizamos os parâmetros: $\alpha = 0.5(1 + \sqrt{5})$ e $\gamma = 1$. Além disso, foi introduzida uma salvaguarda de modo que $\eta_k \in [10^{-6}, 10^{-2}]$.

Como já foi dito no início deste capítulo, na maior parte dos testes e na totalidade daqueles apresentados neste trabalho, utilizamos como critério de aceitação a proposta encontrada em [2]:

$$\|F(x_k + \xi s_k)\| < (1 - \xi\sigma)\|F(x_k)\| + \mu_k. \quad (4.2)$$

Escolhemos o parâmetro $\sigma = 10^{-4}$ para o critério (4.2) acima. Além disso, fazemos ao início de cada iteração externa $\xi = 1$, e enquanto o critério de aceitação não se verifica, definimos $\xi = 0.5\xi$. A escolha de μ_k é feita da seguinte forma:

1. $ftip(0) = \|F(x_0)\|_2$
2. $\begin{cases} ftip(k) = \min\{\|F(x_k)\|_2, ftip(k-1)\}, & \text{se } k \text{ é múltiplo de } 3 \\ ftip(k) = ftip(k-1), & \text{caso contrário} \end{cases}$
3. $\mu_k = ftip(k)/(k+1)^{1.1}$

Além disso, estabelecemos como critério de parada que o programa termine se $\|F(x_k)\|_2 \leq \sqrt{n}10^{-6}$ ou $k > 100$. Utilizamos $m = 30$ como o número máximo de iterações para cada ciclo do GMRES e estabelecemos o limite máximo de 20 ciclos. Para o valor de h na implementação *matrix-free* (3.2), testamos

inicialmente a proposta de Dennis e Schnabel (3.3) e depois comparamos com a proposta de Bellavia e Morini (3.4).

Todos os testes computacionais foram realizados em Matlab® 7.0.

4.1.2 Análise de Desempenho

Para analisar o desempenho dos algoritmos criados, utilizamos a ferramenta perfil de desempenho, proposta em 2002 por Dolan e Moré [7]. Essa ferramenta é ideal para comparar a eficiência e robustez entre vários algoritmos que são aplicados aos mesmos problemas, sob as mesmas condições, como é o nosso caso.

Escolhida uma medida de desempenho (tempo, número de avaliação de funções, iterações, etc), devemos determinar a taxa de desempenho do algoritmo s na resolução do problema p . Considerando S e P o conjunto de todos os algoritmos e problemas respectivamente e ainda $m_{s,p}$ a medida de desempenho para esse problema, sua taxa de desempenho será dada por:

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

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

Após isso, e para cada problema, devemos construir uma função acumulativa $\rho_s : \mathbb{R} \rightarrow [0, 1]$. Considerando $\#(A)$ a cardinalidade de um conjunto A , essa função pode ser escrita como:

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

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 notar para qual valor de τ teremos $\rho_s(\tau) = 1$. Obviamente quanto menor for esse valor τ mais robusto é o algoritmo. Atentemo-nos para o fato de que se ao menos um problema não

obtiver convergência para o algoritmo s teremos $\rho_s(\tau) = 1$ somente para $\tau = r_M$.

Devemos salientar que o maior benefício adquirido com o uso dessa ferramenta é a visualização simultânea do comportamento de diferentes algoritmos. Isso nos permitirá compará-los facilmente, tanto na questão da eficiência quanto da robustez.

4.2. TESTES INICIAIS

Com o objetivo de testar as implementações e analisar a eficiência do algoritmo, realizamos uma bateria inicial de testes numéricos baseada na resolução dos sistemas não-lineares propostos por Luksān em [12].

Resolvemos 16 dos 17 problemas propostos, com 10 aproximações iniciais:

- | | |
|-----------------------------|-------------------------|
| 1. $x_0 = (0, 0, \dots, 0)$ | 6. $x_0 = 2x_{stand}$ |
| 2. $x_0 = (1, 1, \dots, 1)$ | 7. $x_0 = 5x_{stand}$ |
| 3. $x_0 = (2, 2, \dots, 2)$ | 8. $x_0 = -x_{stand}$ |
| 4. $x_0 = (5, 5, \dots, 5)$ | 9. $x_0 = -2x_{stand}$ |
| 5. $x_0 = x_{stand}$ | 10. $x_0 = -5x_{stand}$ |

Onde x_{stand} é o valor padrão proposto em [12] e pode ser conferido na Tabela 4.1.

Utilizando dimensão $n = 1000$ e os valores iniciais descritos acima, comparamos o desempenho de quatro algoritmos:

1. sem busca linear, estratégia de globalização baseada unicamente em regiões de confiança (**SBL**);
2. com busca 3 buscas lineares e teste de aceitação baseado no critério não-monótono proposto em [2], para ambas as estratégias de globalização (**3BL1**);
3. com três buscas lineares, com teste de aceitação não-monótono proposto em [2] após as buscas lineares e teste de aceitação (4.1) após a execução do processo *Double Dogleg* (**3BL2**);

4. algoritmo sem estratégia de globalização (**SGLOB**).

	PROBLEMA	x_{stand}^t
1	<i>Counter Current reactors problem1</i>	(0.1, 0.2, 0.3, 0.4, 0.5, 0.4, 0.3, 0.2, 0.1, 0.2...)
2	<i>Extended Powell badly scaled function</i>	(1, 0, 1, 0, 1, 0, ...)
3	<i>A trigonometric system</i>	(1/n, 1/n, ...)
4	<i>A trigonometric-exponential system 1</i>	(0, 0, 0, ...)
6	<i>Singular Broyden problem</i>	(-1, -1, -1, ...)
7	<i>Tridiagonal system</i>	(12, 12, 12, ...)
8	<i>Five-diagonal system</i>	(-2, -2, -2, ...)
9	<i>Seven-diagonal system</i>	(-3, -3, -3, ...)
10	<i>Structured Jacobian problem</i>	(-1, -1, -1, ...)
11	<i>Extended Rosenbrock function</i>	(-1.2, -1, -1.2, -1, ...)
12	<i>Extended Powell singular function</i>	(3, -1, 0, 1, 3, -1, ...)
13	<i>Extended Cragg and Levy function</i>	(1, 2, 1, 2, ...)
14	<i>Broyden tridiagonal function</i>	(-1, -1, -1, ...)
15	<i>Broyden banded problem</i>	(-1, -1, -1, ...)
16	<i>Discret boundary value problem</i>	($h_1, h_2, \dots$), onde $h_i = (i/(n+1))(i/(n+1) - 1)$
17	<i>Broyden tridiagonal problem</i>	(-1, -1, -1, ...)

Tabela 4.1: Problemas propostos em [12] e seus respectivos valores iniciais padrão

A Figura 4.3 representa um gráfico comparativo para este teste. O melhor desempenho do algoritmo sem globalização nas medidas de tempo e número de avaliação de funções já era esperado, visto que o acionamento de estratégias de globalização afeta diretamente estes itens. Este algoritmo só teria desempenho inferior no quesito eficiência, caso resolvesse um número muito menor de problemas do que os algoritmos com estratégia de globalização, o que não ocorreu.

Tanto em eficiência quanto em robustez o algoritmo **SBL** obteve o pior desempenho em todos os quesitos. As estratégias híbridas foram ligeiramente mais robustas que a estratégia sem globalização, já que resolveram alguns problemas a mais, e foram mais eficientes em números de iterações externas. O desempenho dos algoritmos **3BL1** e **3BL2** foi bem parecido, o que leva a crer, que na maioria dos casos, não tivemos a necessidade de acionar mais do que três buscas lineares. As estratégias de globalização não foram muito requisitadas, dado o baixo número de problemas resolvidos exclusivamente por algoritmos dotados dessas estratégias.

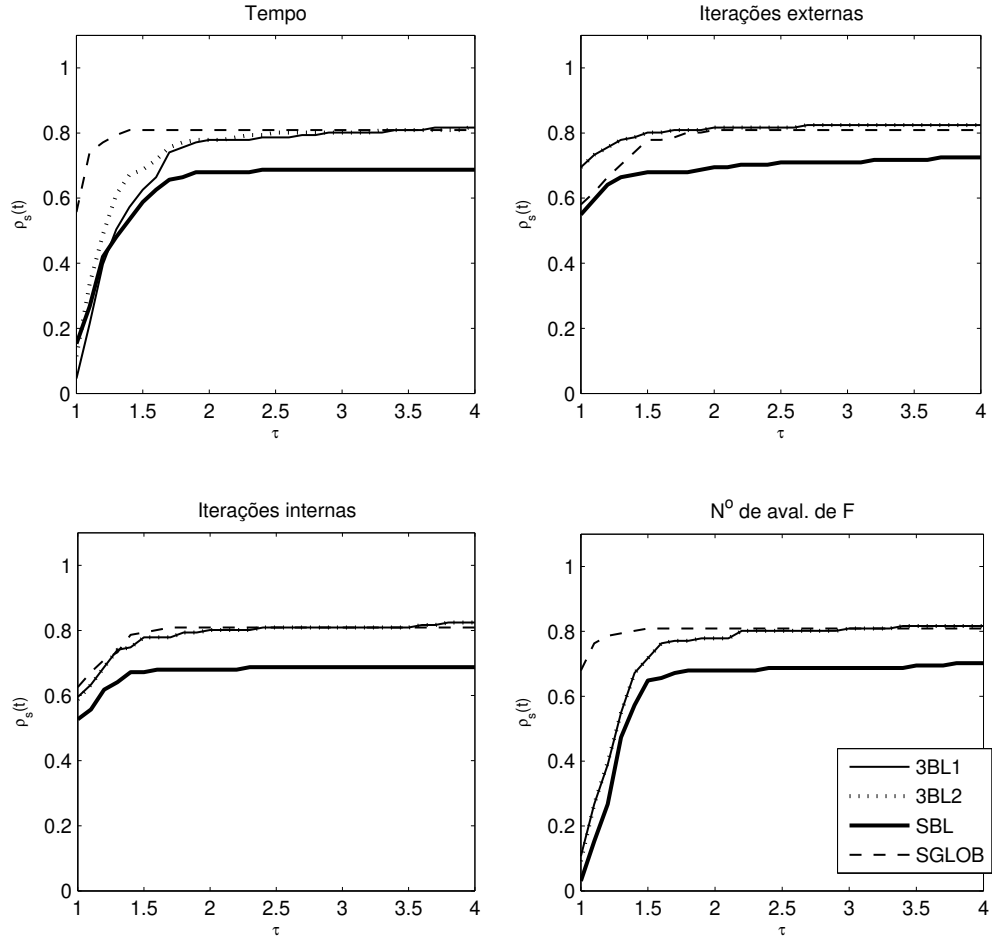


Figura 4.3: Gráfico de desempenho dos algoritmos - $\tau \in [1, 4]$

Após isso, comparamos o desempenho de diferentes valores para h na estratégia *matrix-free* utilizando os valores propostos por Dennis e Schnabel (3.3) e Bellavia e Morini (3.4). O gráfico de desempenho desse teste encontra-se na Figura 4.4, onde nota-se um desempenho melhor da segunda estratégia, denotada por $h : \mathbf{BM}$ sobre a primeira ($h : \mathbf{DS}$). Para ambos os testes considerou-se o Algoritmo **3BL1**.

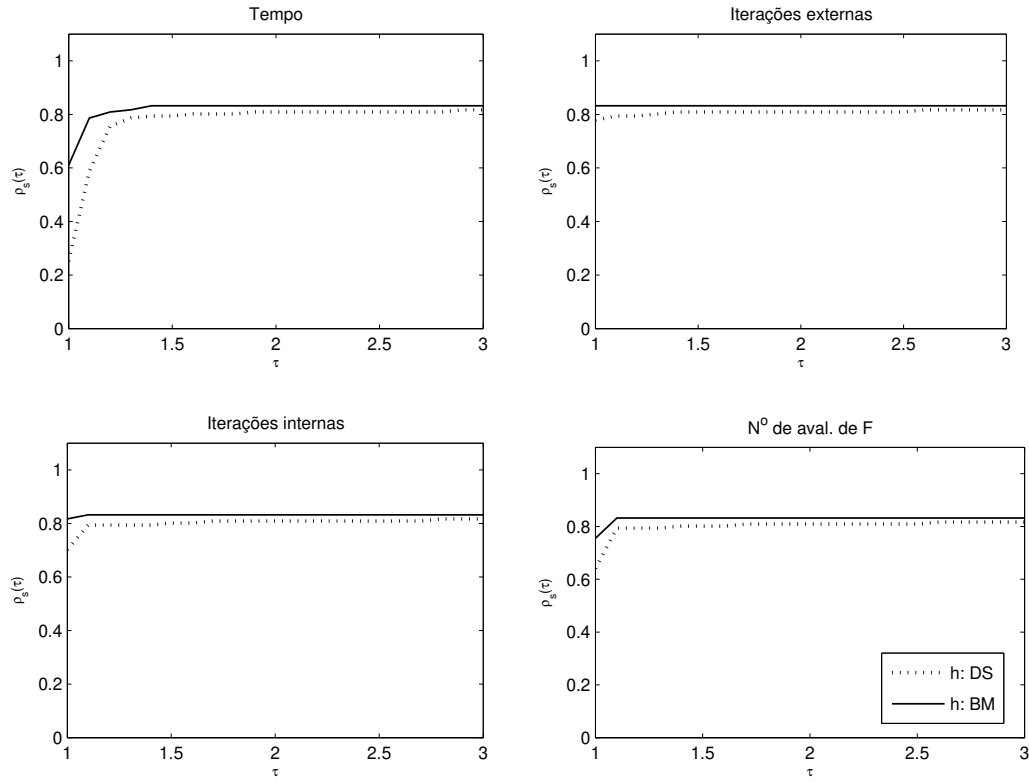


Figura 4.4: Gráfico de desempenho - parâmetro *matrix-free* - $\tau \in [1, 4]$

Com a intenção de comparar com os resultados encontrados por Bellavia e Morini e publicados em [1], resolvemos novamente o problema 2 (*Extended Powell Badly Scaled*), agora para $n = 4096$ e para os mesmo valores iniciais do caso anterior.

O algoritmo de Bellavia e Morini, consiste num método Newton-GMRES com estratégia híbrida de globalização, porém ao invés de utilizar a estratégia de regiões de confiança *Dogleg* é proposta uma avaliação das direções de descida encontradas durante o processo do GMRES comparando qual direção proporcionará a maior redução no valor da função de mérito. Chamando esta direção mais promissora de s_d , o método proposto por Bellavia e Morini propõe a criação de um caminho ligando a direção s_d à direção Newton-Inexata

encontrada pelo GMRES. Os dados comparativos entre esse Algoritmo, o qual chamamos de **ABM** e o nosso algoritmo **3BL1**, encontra-se na Tabela 4.2. Nesta tabela estão denotados por * os testes que não apresentaram convergência.

x_0	Iterações Externas			No. de Aval. de F		
	ABM	3BL1-BM	3BL1-DS	ABM	3BL1-BM	3BL1-DS
$(0, 0, \dots, 0)$	*	33	33	*	360	355
$(1, 1, \dots, 1)$	133	17	17	1040	167	168
$(2, 2, \dots, 2)$	134	16	16	1031	152	152
$(5, 5, \dots, 5)$	125	26	27	918	274	284
x_{stand}	135	17	17	1034	164	163
$2x_{stand}$	133	16	16	1029	154	154
$5x_{stand}$	125	26	27	1009	272	283
$-x_{stand}$	*	15	20	*	151	205
$-2x_{stand}$	*	29	29	*	301	301
$-5x_{stand}$	*	29	29	*	290	290

Tabela 4.2: Resultados para o problema *Extended Powell Badly Scaled*([12])

O algoritmo **3BL1** obteve desempenho consideravelmente melhor para ambas as escolhas para o valor de h na técnica *matrix-free*. Já esperávamos que um número de avaliações mais baixo fosse realizado por este algoritmo, devido ao processo de escolha da direção mais promissora do algoritmo de Bellavia e Morini que realiza um grande número de avaliações de função por iteração externa. Mas a eficiência de **3BL1** em termos de número de iterações externas, bem como a robustez, foi um fato que nos surpreendeu, já que este algoritmo obteve solução do problema para todos os valores iniciais.

4.3. PROBLEMAS DE VALOR DE CONTORNO

Para o segundo conjunto de testes, resolvemos uma série de problemas não-lineares oriundos da discretização de problemas de valor de contorno. De uma maneira genérica, os problemas de valor de contorno tratados neste conjunto consistem em encontrar uma função $u : \Omega = [0, 1] \times [0, 1] \rightarrow \mathbb{R}$ de forma que,

dados $\lambda \in \mathbb{R}$, teremos:

$$-\nabla u + h(\lambda, u) = w(s, t) \text{ em } \Omega \text{ e } u(s, t) = 0 \text{ em } \partial\Omega, \quad (4.3)$$

onde ∇u é o operador laplaciano.

Os diferentes problemas de valor de contorno são definidos através de variações na função $h(\lambda, \mu)$, do parâmetro λ e da função w .

Seguindo as idéias encontradas em [10], utilizamos o método de diferenças centrais para aproximar as derivadas. Para isso cria-se uma malha de pontos nos quais a equação (4.3) deve ser satisfeita. Assim, a equação (4.3) resulta em um sistema não linear. No caso, trabalhamos com uma malha de 63 pontos internos em cada eixo, do que obtemos um sistema não linear de dimensão 3969.

O primeiro problema que resolvemos é o problema de Bratu. Para este problema teremos $h(\lambda, u) = -\lambda \exp(u)$ na equação (4.3):

$$-\nabla u - \lambda \exp(u) = w(s, t) \text{ em } \Omega \text{ e } u(s, t) = 0 \text{ em } \partial\Omega. \quad (4.4)$$

A nossa idéia é comparar os resultados encontrado com a solução real e conforme sugerido em [10] geramos uma série de problemas onde a função w é calculada de tal forma que a solução exata do sistema resultante seja dada por:

$$u_* = 10st(1-s)(1-t)\exp(s^{4.5}). \quad (4.5)$$

Os diferentes valores para λ determinam o grau de dificuldade do problema. Segundo [20], problemas com valores negativos para λ são considerados fáceis, enquanto aqueles em que adotarmos valores positivos seriam mais difíceis.

Assim como em [20], geramos 22 testes com 11 problemas diferentes e duas aproximações iniciais. Os valores para λ escolhidos foram: -1000 , -500 , -250 , -100 , -50 , -10 , 1 , 3 , 5 , 7 , 10 .

Para a aproximação inicial escolhemos o vetor $x_0 = (0, 0, \dots, 0)^t$ e um vetor com componentes aleatórias pertencentes ao intervalo $[-5, 5]$.

	λ :	-1000	-500	-250	-100	-50	-10	1	3	5	7	10
3BL1	EX	5	5	5	5	5	4	3	4	4	5	6
BM	IN	60	83	113	172	244	419	633	696	875	1038	2336
3BL1	EX	5	5	5	5	5	4	3	4	4	5	6
DS	IN	60	83	113	172	244	418	631	696	875	1036	2347
SBL	EX	5	5	5	5	5	4	3	4	4	5	6
	IN	60	83	113	172	244	418	631	696	875	1036	2347
3BL2	EX	5	5	5	5	5	4	3	4	4	5	6
	IN	60	83	113	172	244	418	631	695	875	1036	2348
SGLOB	EX	5	5	5	5	5	4	3	4	4	5	6
	IN	60	83	113	172	244	418	631	695	1010	1036	2347

Tabela 4.3: Resultados para o problema de Bratu (4.4) para $x_0 = (0, 0, \dots, 0)^t$

Os resultados obtidos pelos algoritmos testados no problema de Bratu encontram-se nas Tabelas 4.3 e 4.4 e também no gráfico da Figura 4.5. Nas tabelas e no gráfico os algoritmos seguem com a mesma notação anterior, sendo que na tabela somente estão representadas as iterações externas (**EX**) e as iterações internas (**IN**).

Pode-se observar que o desempenho entre os algoritmos foi bem parecido, exceto, como já era esperado, no número de avaliações de funções. No caso do número de iterações externas, por exemplo, não houve nenhuma diferença no desempenho. Novamente a estratégia de globalização foi pouco acionada.

Com o intuito de acionar as estratégias de globalização, resolvemos um novo tipo de problema de valor de contorno: o problema de Convecção-Difusão. Neste caso definimos a função $h(\lambda, u) = \lambda u(\frac{\partial u}{\partial s} + \frac{\partial u}{\partial s})$, o que nos dá o seguinte problema:

$$-\nabla u + \lambda u(\frac{\partial u}{\partial s} + \frac{\partial u}{\partial s}) = w(s, t) \text{ em } \Omega \text{ e } u(s, t) = 0 \text{ em } \partial\Omega \quad (4.6)$$

Geramos os problemas da mesma forma que o caso anterior, isto é de maneira que a solução exata do sistema seja aquela apresentada em (4.5). Realizamos o teste para nove valores distintos para λ : 5, 10, 25, 50, 75, 100, 110, 125 e

	λ :	-1000	-500	-250	-100	-50	-10	1	3	5	7	10
3BL1	EX	8	8	8	8	8	4	4	5	5	6	7
BM	IN	86	99	138	200	263	468	846	937	1074	1335	3075
3BL1	EX	8	8	8	8	8	4	4	5	5	6	7
DS	IN	86	99	138	200	263	468	853	935	1085	1221	3075
SBL	EX	8	8	8	8	8	4	4	5	5	6	7
	IN	86	99	138	200	263	468	853	937	1085	1221	3075
3BL2	EX	8	8	8	8	8	4	4	5	5	6	7
	IN	86	99	138	200	263	468	853	937	1085	1221	3075
SGLOB	EX	8	8	8	8	8	4	4	5	5	6	7
	IN	86	99	138	200	263	468	853	937	1085	1332	3075

Tabela 4.4: Resultados para o problema de Bratu (4.4), x_0 com valores aleatórios entre 5 e -5

150. E para a aproximação inicial utilizamos o vetor $x_0 = (0, 0, \dots, 0)^t$ para todos os problemas gerados.

Os resultados para os testes com problemas de Convecção-Difusão podem ser conferidos na Tabela 4.5. Nela, novamente temos que **EX** é o número de iterações externas, **IN** é o número de iterações internas. Dessa vez, como alguns algoritmos não obtiveram convergência em determinados problemas, adicionamos nas tabelas uma variável que representa a diferença entre a solução exata e a solução encontrada pelos programas: $e = \|x^* - x\|_\infty$, onde x^* é a solução exata para o problema e x é a solução encontrada pelo algoritmo em questão. Os casos em que $e < 10^{-8}$ estão denotados simplesmente por —, já que convergiram para a solução exata do algoritmo.

O algoritmo sem globalização mostrou grande deficiência na resolução dos problemas mais difíceis e não resolveu 4 problemas propostos, enquanto o algoritmo **3BL1** com a escolha de Bellavia e Morini para o parâmetro h na técnica *matrix free* obteve melhor desempenho e falhou em somente um problema. Além disso, é interessante notar que entre os testes que divergiram, os resultados dos testes com estratégia de globalização ficaram relativamente muito mais próximos da solução exata do que os resultados obtidos pelo algoritmo sem globalização.

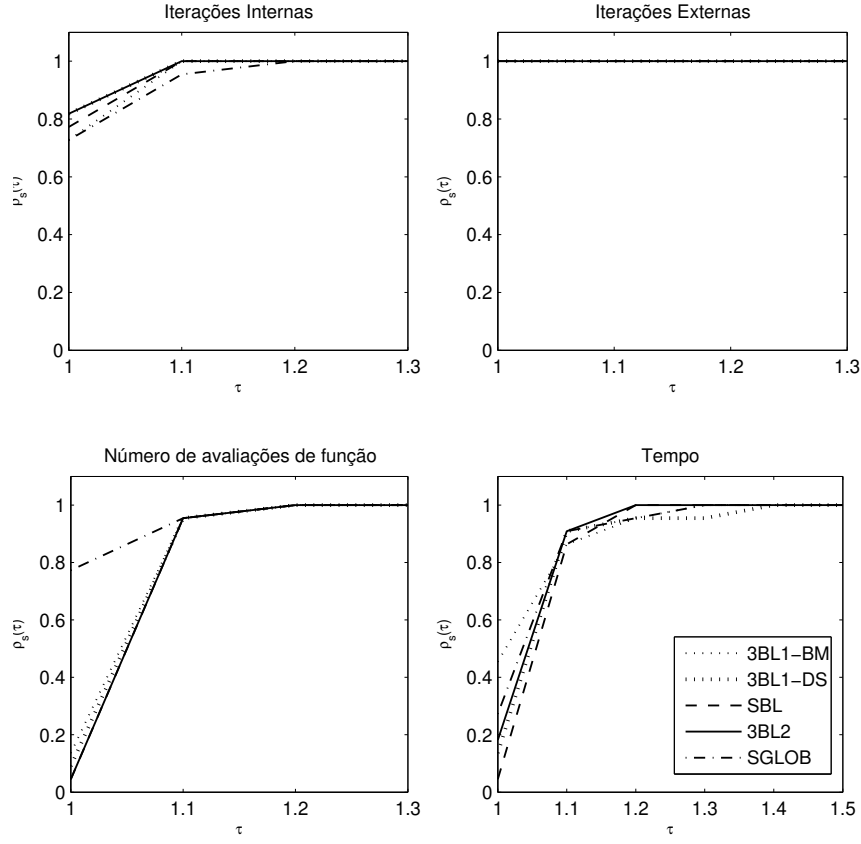


Figura 4.5: Gráfico de desempenho - Problema de Bratu

Outro dado que devemos ressaltar é o desempenho do algoritmo sem busca linear, que em 6 testes convergiu para uma solução diferente da que esperávamos. Isso talvez se justifique pelo fato de tratar-se de um algoritmo mais rígido que os demais, impedindo um avanço maior quando a estratégia de globalização é acionada. Além disso, houve divergência em um teste.

A primeira justificativa que consideramos como causa para a divergência de alguns testes foi a estagnação do método GMRES com recomeços. Resolvemos então refazer os testes, mas alterando o tamanho máximo do ciclo para $m = 50$.

	λ :	5	10	25	50	75	100	110	125	150
3BL1 BM	e	—	—	—	—	—	—	—	—	0.7350
	EX	5	5	7	9	10	18	21	26	100
	IN	596	510	531	834	1553	3679	4934	7924	28069
3BL1 DS	e	—	—	—	—	—	—	—	0.6064	0.8445
	EX	5	5	7	9	11	18	21	100	100
	IN	595	510	528	834	1583	3761	5040	27257	28022
SBL	e	—	—	0.4960	0.7495	1.5047	2.0062	2.2068	2.5077	3.0091
	EX	5	5	6	100	4	5	5	5	5
	IN	595	510	893	29619	500	699	702	686	657
3BL2	e	—	—	—	—	—	—	—	0.6212	0.7575
	EX	5	5	7	9	11	18	21	100	100
	IN	595	510	528	866	1605	3776	5003	27145	27986
SGLOB	e	—	—	—	—	—	536.0498	318.2929	90.7790	61.7805
	EX	5	5	7	10	14	100	100	100	100
	IN	595	510	533	591	681	57353	57900	58497	58499

Tabela 4.5: Resultados para o problema de Convecção-Difusão (4.6), com $m = 30$

Nesse novo teste, cujos resultados podem ser conferidos na Tabela 4.6, os Algoritmos **3BL1-BM** e **3BL2** tiveram desempenho satisfatório e resolveram todos os problemas propostos. O algoritmo **SGLOB** novamente se mostrou ineficiente e divergiu nos mesmos 4 testes do caso anterior. O algoritmo **SBL** novamente convergiu para outras soluções, só que desta vez mudou o único problema para o qual não houve convergência.

Por curiosidade e para ilustrar melhor os resultados encontrados pelo algoritmo, imprimimos inicialmente o gráfico da solução exata para o sistema linear (Figura 4.6) e depois os gráficos das soluções obtidas, para todos os valores de λ , pelos Algoritmos **3BL1**, **SBL** e **SGLOB**. Os gráficos estão representados nas Figuras 4.7, 4.8 e 4.9, respectivamente.

Para finalizar este capítulo realizamos a análise do perfil de desempenho dos algoritmos levando-se em consideração todos os problemas de valor de contorno resolvidos, com ciclo máximo de 30 iterações para o GMRES. Na Figura 4.10 encontra-se o gráfico de análise de desempenho no intervalo $\tau \in [1, 1.5]$ onde podemos analisar cada algoritmo com relação à eficiência. Já na

	$\lambda :$	5	10	25	50	75	100	110	125	150
3BL1 BM	e	—	—	—	—	—	—	—	—	—
	EX	5	5	7	9	11	18	21	26	34
	IN	475	485	484	862	1243	2860	4143	5830	8968
3BL1 BS	e	—	—	—	—	—	—	—	—	—
	EX	5	5	7	9	11	18	21	26	33
	IN	475	485	483	858	1264	2857	4499	5782	9072
SBL	e	—	—	0.4973	1.0029	1.5046	2.0061	2.2067	2.5076	3.0091
	EX	5	5	100	4	4	5	5	5	5
	IN	475	485	681	578	595	598	611	580	562
3BL2	e	—	—	—	—	—	—	—	—	—
	EX	5	5	7	9	11	18	22	26	34
	IN	475	485	483	860	1258	2877	4172	5775	9071
SGLOB	e	—	—	—	—	—	533.8325	442.7886	315.4074	75.6387
	EX	5	5	7	10	14	100	100	100	100
	IN	475	485	479	467	552	47202	48178	48214	48254

Tabela 4.6: Resultados para o problema de Convecção-Difusão (4.6), com $m = 50$

Figura 4.11, temos ilustrado o intervalo $\tau \in [1, 3]$, que facilita a análise da robustez. Os testes que convergiram para outras soluções foram considerados como falha do algoritmo.

Neste novo conjunto de testes, em que o acionamento de estratégias de globalização era uma necessidade, podemos ver claramente a eficiência da estratégia híbrida proposta. Os algoritmos providos de estratégia deste tipo, tiveram os melhores desempenhos tanto em números de iterações externas quanto de internas. Além disso, o algoritmo **3BL1-BM**, o único dotado da estratégia de obtenção para o valor de h no *matrix-free* proposta em [1], mostrou novamente desempenho superior aos demais. Inclusive obteve melhor desempenho quanto se trata do tempo de execução sendo melhor que o algoritmo sem globalização, o que é um fato surpreendente, dado que este último tem uma estrutura muito mais simples.

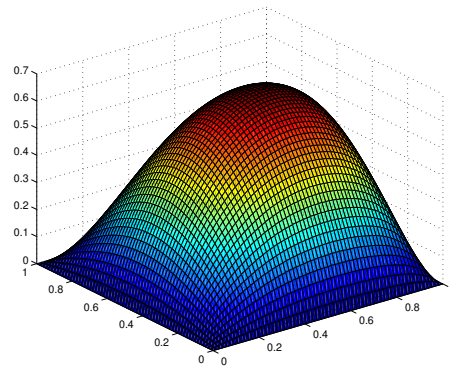


Figura 4.6: Solução exata para o problema de Convecção-Difusão 4.6

Quanto à robustez, o desempenho é melhor ainda, pois em um número maior de problemas obteve resultados corretos, além do que a não convergência de alguns problemas quando utilizada a estratégia híbrida se deve à estagnação do $\text{GMRES}(m)$, o que já estava previsto na teoria.

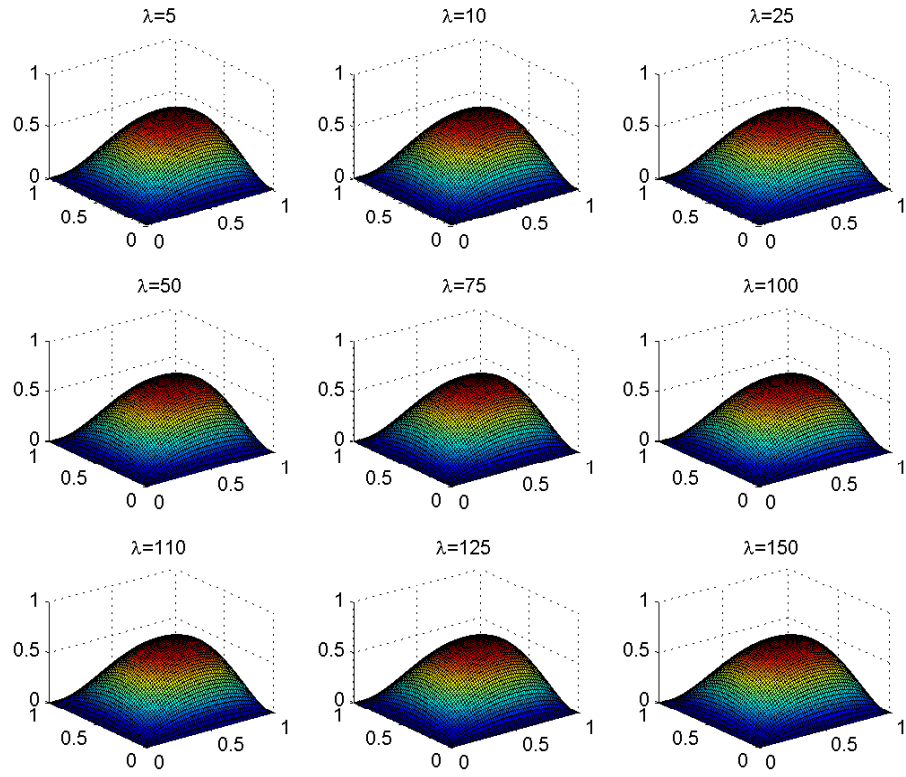


Figura 4.7: Soluções encontradas para o Problema de Convecção-Difusão (4.6) pelo algoritmo **3BL1-BM**

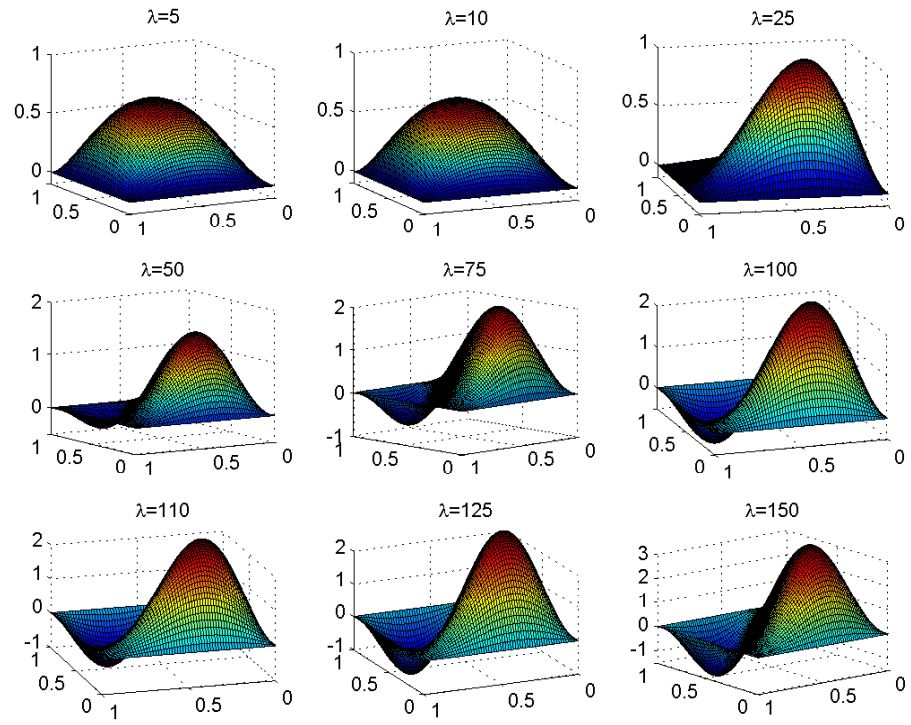


Figura 4.8: Soluções encontradas para o Problema de Convecção-Difusão (4.6) pelo algoritmo **SBL**

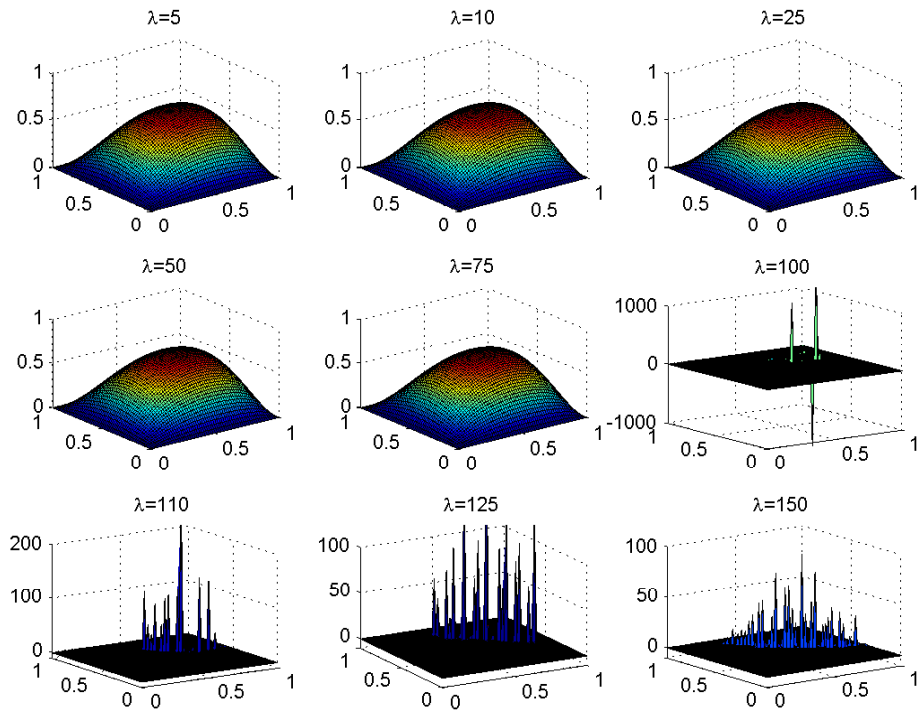


Figura 4.9: Soluções encontradas para o Problema de Convecção-Difusão (4.6) pelo algoritmo **SGLOB**

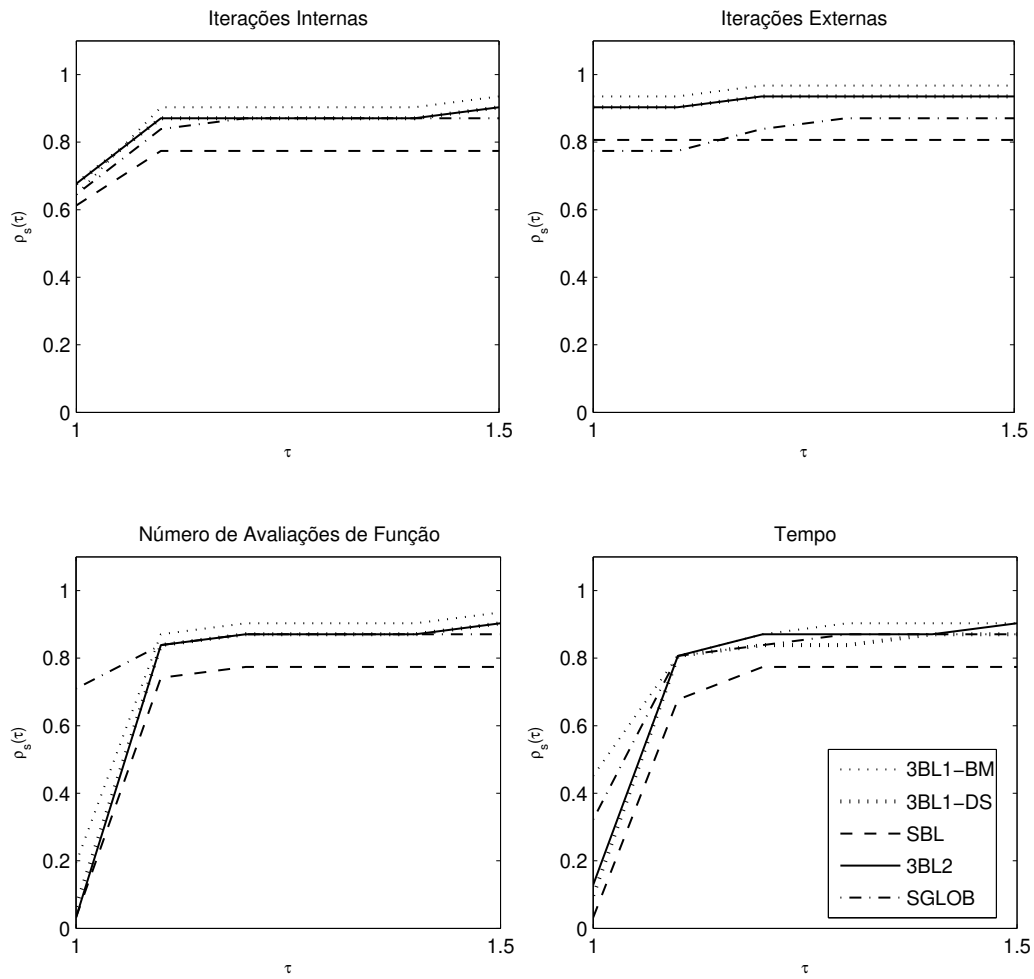


Figura 4.10: Gráfico de desempenho - Testes com Problema de Valor de Contorno - $\tau \in [1, 1.5]$

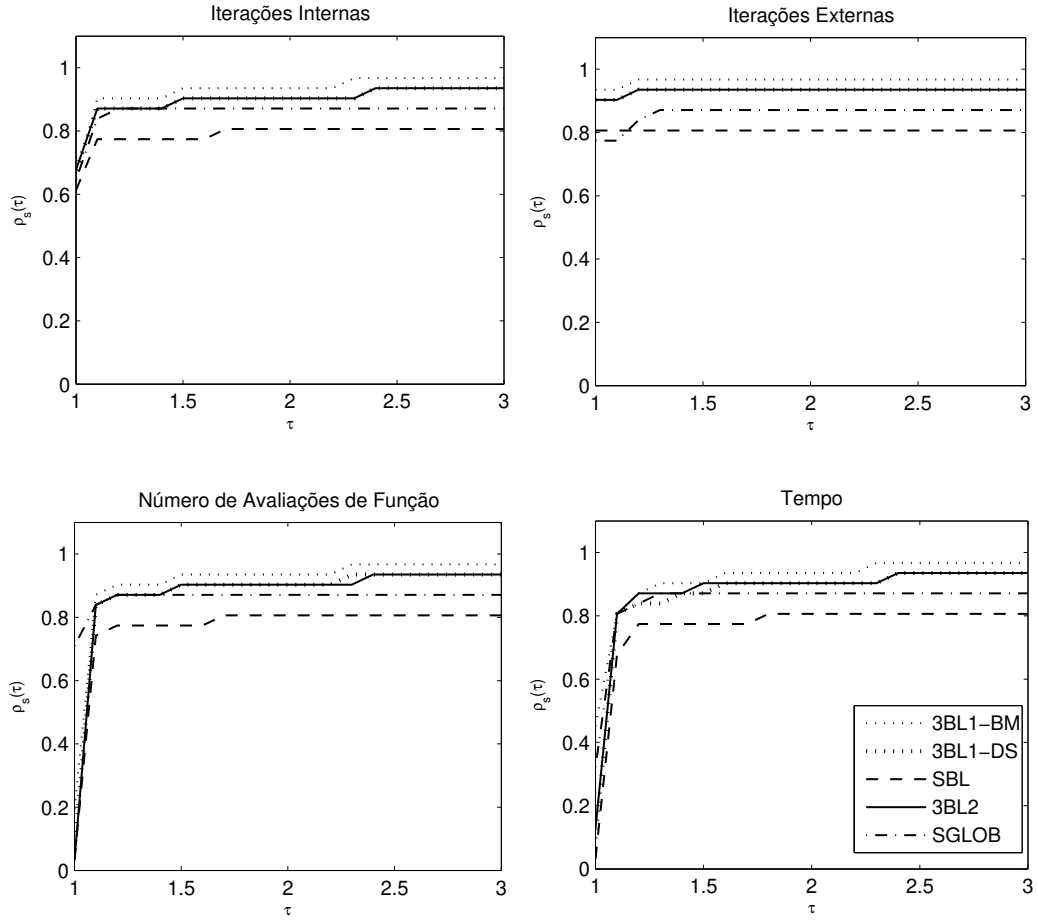


Figura 4.11: Gráfico de desempenho - Testes com problemas de Valor de Contorno - $\tau \in [1, 3]$

Nesse trabalho utilizamos algoritmos Newton-GMRES para a resolução de sistemas não-lineares de grande porte. Com o objetivo de diminuir o esforço computacional e os requerimentos de memória foram utilizadas três modificações clássicas: uso do método Newton-inexato, da estratégia com recomeços para o GMRES, que permite a diminuição da dimensão máxima do subespaço que trabalhamos e por fim, uso da estratégia *matrix-free* que evita o cálculo da matriz Jacobiana.

Apoiados nessa teoria, desenvolvemos três algoritmos com opções distintas na estratégia de globalização: o primeiro deles com uma estratégia exclusivamente baseada em regiões de confiança, e os outros dois com no máximo três buscas lineares prosseguidas, caso necessário, da estratégia de regiões de confiança. O que difere nesses dois últimos é o critério de aceitação para pontos advindos da estratégia de globalização.

O primeiro algoritmo não apresentou um desempenho satisfatório durante a resolução de testes numéricos, tendo desempenho inferior aos demais, inclusive em relação ao algoritmo sem globalização. Acreditamos que isso se deve principalmente à rigidez dessa estratégia com relação aos demais. Provavelmente melhores escolhas para o valor do raio inicial pode ser uma alternativa para contornar esse desempenho insatisfatório.

Os demais algoritmos mostraram eficiência e robustez, e nos problemas em que foi necessário o acionamento das estratégias de globalização, demonstraram bom desempenho resolvendo quase todos os testes propostos.

Alguns problemas não foram resolvidos por causa de estagnações causadas

pela estratégia de recomeços do GMRES, sendo que alguns problemas só foram resolvidos quando aumentamos o número máximo de iterações em cada ciclo do GMRES. Consideramos que uma saída ideal para este tipo de problema seria o uso de uma estratégia não-constante para o valor de m , esse é um tema para futuras pesquisas. Outra sugestão seria adicionar o uso de pré-condicionadores.

Por fim, gostaríamos de ressaltar o melhor desempenho de nosso algoritmo frente ao algoritmo proposto por Bellavia e Morini em [1]. Acreditamos que esse fato deve-se principalmente à estratégia não-monótona (2.5) adotada para aceitação de pontos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Bellavia, S. & Morini, B., *A Globally Convergent Newton-GMRES Subspace Method for System of Nonlinear Equations*, SIAM J. Sci. Comput., Vol. 23, No. 3, pp. 940-960, 2001.
- [2] Birgin, E.J., Krejić, N., & Martínez, J.M., *Globally Convergent Inexact Quasi Newton Methods for Solving Nonlinear Systems*, Numerical Algorithms, Vol. 32, pp. 249-260, 2003.
- [3] Brown, P.N. & Saad, Y., *Hybrid Methods for Nonlinear Systems of Equations*, SIAM Journal of Sci. Stat. Comput. Vol. 11, No. 3, pp. 450-481, 1990.
- [4] Dembo, R. S., Eisenstat, S. C. & Steihaug, T., *Inexact Newton Methods*, SIAM J. Numer. Anal., Vol.19, No. 2, pp. 401-408, 1982.
- [5] Dennis, J.E. & Schnabel R.B., **Numerical Methods for Unconstrained Optimization and Nonlinear Equations**, SIAM Classics in Applied Mathematics, 1996.
- [6] Dennis, J.E. & Mei, H.H.W., *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., No. 28, pp. 453-482, 1979.
- [7] Dolan, E.D. & Moré, J.J., *Benchmarking optimization software with performance profiles*, Math. Programming Ser. A91, pp. 201-213, 2002.
- [8] Eisenstat, S.C. & Walker, H. F., *Choosing the Forcing Terms in Inexact-Newton Methods*, SIAM J. Sci. Comput., Vol.17, No.1, pp.16-32, 1996.

- [9] Friedlander, A., *Elementos de Programação Não-Linear*, Editora da Unicamp, 1994.
- [10] Kelley, C.T., *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.
- [11] Luenberger, D.G., *Linear and Nonlinear Programming*, 2^a. ed., Addison - Wesley Publishing Company, 1986.
- [12] Luksán, L., *Inexact Trust Region Method for Large Sparse System of Nonlinear Equations*, Journal of Optimization Theory and Applications, Vol. 81, pp. 569-590, 1994.
- [13] Martínez, J.M. & Santos, S.A., *Métodos Computacionais de Otimização* - disponível em www.ime.unicamp.br/~sandra.
- [14] Powell, M. J. D., *A Hybrid Method for Nonlinear Equations*, in *Numerical Methods For Nonlinear Algebraic Equations*, Rabinowitz, P., Gordon and Breach, 1970.
- [15] Pernice, M. & Walker, H., *NITSOL: a Newton Iterative Solver for Nonlinear Systems*, SIAM, J. Sci. Comput. Vol.19, No.1, pp. 302-318, 1998.
- [16] Saad, Y., *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.
- [17] Saad, Y. & Schultz, M. H., *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM, J. Sci. Stat. Comput., Vol.7, No.3, 1986.
- [18] Sorensen, D. C., *Newton's Method with a model trust region modification*, SIAM J. Numer. Anal. Vol.19, pp. 409-426, 1982.
- [19] Trefethen, L. N. & Bau, D., *Numerical Linear Algebra*, SIAM, 1997.
- [20] Toledo-Benavides, J. V., *Um Método Newton-GMRES Globalmente Convergente com uma Nova Escolha para o Termo Forçante e Algumas Estratégias para Melhorar o Desempenho do GMRES(m)*, Tese de Doutorado, 2005. UNICAMP.