

Análise do desempenho do método GMRES

Marcos Eduardo Ribeiro do Valle Mesquita*

Maria Aparecida Diniz-Ehrhardt

DMA - IMECC - Unicamp

Resumo

Neste trabalho voltaremos nossa atenção para dois problemas encontrados em várias aplicações . O primeiro deles é a resolução de um sistema linear, de grande porte, no qual a matriz de coeficientes possui uma estrutura esparsa. Proporemos, para este tipo de sistemas, o método iterativo GMRES (Generalized Minimal RESidual), introduzido em 1986 por Saad e Schultz. Tal algoritmo tem como objetivo minimizar a cada iteração a norma do resíduo sobre o subespaço de Krylov, utilizando o método de Arnoldi (equivalente ao processo de Gram-Schmidt) para gerar uma base ortogonal para este subespaço.

O segundo problema considerado aqui é o de resolver um sistema não linear. Um método clássico para resolver este tipo de problema é o método de Newton, atraente devido à sua rápida convergência. Mas a necessidade da resolução de um sistema de equações lineares a cada iteração pode ser cara se o número de incógnitas for grande. Assim, proporemos o método de Newton Inexato, que utiliza um método iterativo para a resolução do sistema linear. Neste trabalho combinaremos o método de Newton com o método GMRES para resolver os sistemas lineares a cada iteração. No final apresentaremos alguns testes a fim de avaliar o desempenho do GMRES inserido no método de Newton Inexato. Para isto, utilizaremos programas em MATLAB implementados por C. T. Kelley, em 1995.

*Este autor foi financiado pela Fapesp

Abstract

In this work, we are interested in the resolution of two kind of problems that have many practical applications. The first one is the resolution of a large linear system of equations, in which the coefficient matrix is sparse. We propose the iterative method GMRES (Generalized Minimal RESidual), introduced by Saad and Schultz, in 1986. At each iteration of this algorithm, one minimizes the residual norm in the Krylov subspace, using the Arnoldi method (equivalent to the Gram-Schmidt process) to compute an orthonormal basis for this subspace.

The second problem is the resolution of nonlinear systems of equations. The classical Newton method is very attractive to solve this kind of problem, because it has fast convergence. But it requires the resolution of a linear system at each iteration, which can be too expensive. So, we propose the Inexact Newton method, which uses an iterative method to solve the linear equations. In this work, we combine the Newton method with GMRES applied to the linear system for the Newton step. Finally we present some numerical experiments in order to analyse the performance of Newton–GMRES. We use `MATLAB` codes written by C. T. Kelley, in 1995.

1 Introdução

A necessidade de resolver sistemas lineares e não lineares surge em vários problemas científicos como, por exemplo, quando discretizamos equações diferenciais. Neste trabalho vamos escrever um sistema de equações lineares como

$$Ax = b \tag{1}$$

onde A é uma matriz $n \times n$ não singular, $b \in \mathbb{R}^n$ é um vetor conhecido e

$$x_* = A^{-1}b \in \mathbb{R}^n \tag{2}$$

é a solução procurada. Trabalharemos também com a resolução de sistemas de equações não lineares, ou seja, com o problema:

$$\text{Dado } F : \mathbb{R}^n \longrightarrow \mathbb{R}^n, \text{ encontrar } x_* \in \mathbb{R}^n \text{ tal que } F(x_*) = 0. \tag{3}$$

Denotaremos f_i como a i -ésima componente de F . Se F é diferenciável em $x \in \mathbb{R}^n$, então definimos a matriz Jacobiana $J(x)$ ou $F'(x)$ por

$$J(x)_{ij} = F'(x)_{ij} = \frac{\partial f_i(x)}{\partial x_j}.$$

Para a resolução de um sistema linear podemos utilizar métodos diretos ou iterativos. Nos métodos diretos, operações elementares são aplicadas ao sistema de forma a transformá-lo em outro mais simples, cuja solução é a mesma do original. Ou, equivalentemente, a matriz de coeficientes do sistema é decomposta em um produto de dois ou mais fatores e, posteriormente, sistemas lineares fáceis são resolvidos, conduzindo à solução do sistema original. Entre estes métodos encontra-se o método da Eliminação de Gauss, Fatoração LU e Decomposição de Cholesky. Porém estes métodos não são recomendados se a ordem do sistema for grande, pois eles podem provocar um grande preenchimento da matriz (elementos nulos substituídos por não nulos) implicando na necessidade de maior alocação de memória. Pode ainda haver um grande acúmulo de erros de arredondamento, levando a soluções pouco precisas [4], [5] e [18].

Para evitarmos tais problemas podemos utilizar métodos iterativos, cuja idéia central é gerar uma seqüência $\{x_k\}$ que convirja para a solução do sistema. Os métodos iterativos não acumulam erros de arredondamento; além disso, aproveitam a estrutura da matriz, sendo preferíveis quando esta é esparsa (possui um número relativamente pequeno de elementos não nulos).

O melhor e mais eficiente método iterativo para a resolução de sistemas lineares, quando a matriz de coeficientes é simétrica e definida positiva, é o método dos Gradientes Conjugados [10]. Porém surge a necessidade de métodos iterativos para a

resolução de sistemas que não possuem tais características. Vários algoritmos foram propostos após o método dos Gradientes Conjugados e são baseados em projeções sobre o subespaço de Krylov. Como exemplo destes métodos temos SYMMLQ e MINRES para matrizes simétricas; GMRES, QMR, CGN e CGS para matrizes não simétricas. O mais popular dentre os métodos para sistemas não simétricos é o GMRES, que será discutido na próxima seção; os demais métodos não serão discutidos neste trabalho e apenas citaremos [1] como uma breve referência e [15] para um estudo da convergência dos métodos CGN, CGS e GMRES.

Para a resolução de um sistema não linear, utilizamos, em geral, o método de Newton, que trabalha com aproximações lineares para do problema não linear. Assim, reduzimos a resolução do sistema não linear á resolução de sistemas lineares da forma:

$$J(x_k)s_k = -F(x_k), \quad (4)$$

e posteriormente definimos $x_{k+1} = x_k + s_k$ como sendo a nova aproximação para a solução [7].

Como vimos anteriormente, para a resolução de alguns sistemas lineares, os métodos iterativos apresentam melhores resultados que os diretos, assim surge o interesse de implementarmos o método de Newton com algum método iterativo para a resolução dos sistemas lineares. Desta combinação temos o método de Newton Inexato. Seguindo esta linha, na terceira seção vamos descrever o método Newton-GMRES e posteriormente (seção 4) apresentaremos alguns testes numéricos a fim de avaliar o desempenho do algoritmo proposto.

2 GMRES (Generalized Minimum RESidual)

O método GMRES (Generalized Minimum RESidual) foi proposto em 1986 por Saad e Schultz [19] como um método iterativo para sistemas não simétricos, isto é, para sistemas da forma de (1). Sendo A uma matriz genérica, então não podemos pensar em minimizar a função $\phi(x) = (1/2)x^T Ax - x^T b$, como no método dos Gradientes Conjugados. Como alternativa consideraremos o problema de quadrados mínimos $\min_x \|b - Ax\|_2$ e, como em Gradientes Conjugados, minimizaremos a função sobre o subespaço de Krylov. Desta forma nosso objetivo será

$$\min_{x \in x_0 + \mathcal{K}_k} \|b - Ax\|_2 \quad (5)$$

onde \mathcal{K}_k é o k -ésimo subespaço de Krylov dado por

$$\mathcal{K}_k = \text{span}\{r_0, Ar_0, \dots, A^{k-1}r_0\},$$

com

$$r_0 = b - Ax_0.$$

2.1 Propriedades e Conseqüências da Minimização

Como $x \in x_0 + \mathcal{K}_k$, então teremos

$$x = x_0 + \sum_{j=1}^{k-1} \alpha_j A^j r_0$$

e, portanto,

$$r = b - Ax = b - Ax_0 - \sum_{j=0}^{k-1} \alpha_j A^{j+1} r_0 = r_0 - \sum_{j=1}^k \alpha_{j-1} A^j r_0.$$

Assim, se $x \in x_0 + \mathcal{K}_k$, então $r = \bar{p}(A)r_0$, onde $\bar{p} \in \mathcal{P}_k$ é conhecido como polinômio residual e $\mathcal{P}_k = \{p \mid p \text{ é um polinômio de grau } k \text{ e } p(0) = 1\}$. Esse resultado pode ser enunciado como o seguinte teorema.

Teorema 2.1 *Seja A uma matriz não singular e seja x_k a solução da k -ésima iteração do GMRES. Então para todo $\bar{p}_k \in \mathcal{P}_k$ temos*

$$\|r_k\|_2 = \min_{p \in \mathcal{P}_k} \|\bar{p}(A)r_0\|_2 \leq \|\bar{p}_k(A)r_0\|_2.$$

Deste teorema segue o seguinte corolário.

Corolário 2.2 *Seja A uma matriz não singular e seja x_k a solução da k -ésima iteração do GMRES. Então para todo $\bar{p}_k \in \mathcal{P}_k$ temos*

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \|\bar{p}_k(A)\|_2. \quad (6)$$

Com este corolário podemos mostrar que o GMRES termina com um número finito de passos.

Teorema 2.3 *Seja $A \in \mathbb{R}^{n \times n}$ uma matriz não singular. Então o método GMRES obtém a solução do sistema (1) com n iterações.*

Demonstração :

Sabemos que o polinômio característico de A é $p(z) = \det(A - zI)$. p tem grau n e $p(0) = \det(A) \neq 0$, desde que A é não singular. Então

$$\bar{p}_n(z) = \frac{p(z)}{p(0)} \in \mathcal{P}_n$$

é um polinômio residual. Mas sabemos que $p(A) = \bar{p}_n(A) = 0$. Pelo corolário acima temos que $r_n = b - Ax_n = 0$ e portanto x_n é a solução de $Ax = b$. \square

Com este teorema temos que o método GMRES converge para a solução x_* no máximo em n iterações (na ausência de erros de arredondamento). Podemos ainda obter outros resultados quanto à convergência do método; para isto vamos supor que A é uma matriz diagonalizável, isto é, existe uma matriz V (que pode ser complexa) tal que

$$A = V\Lambda V^{-1} \quad (7)$$

onde Λ é uma matriz diagonal, contendo os autovalores de A , ou seja, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ onde λ_i é autovalor de A , para $i = 1, \dots, n$. Temos então o teorema a seguir.

Teorema 2.4 *Seja A uma matriz não singular diagonalizável e considere a diagonalização (7). Seja x_k a k -ésima iteração do GMRES. Então para todo $\bar{p}_k \in \mathcal{P}_k$ temos que*

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \kappa_2(V) \max_{z \in \sigma(A)} |\bar{p}_k(z)| \quad (8)$$

onde $\kappa_2(V)$ é o número de condição da matriz V e $\sigma(A) = \{\lambda \mid \lambda \text{ é um autovalor de } A\}$ é o espectro da matriz A .

Demonstração :

Seja $\bar{p}_k \in \mathcal{P}_k$. Temos que $\bar{p}_k(A) = Vp_k(\Lambda)V^{-1}$. Então

$$\|\bar{p}_k(A)\|_2 \leq \|V\|_2 \|V^{-1}\|_2 \|\bar{p}_k(\Lambda)\|_2 \leq \kappa(V) \max_{z \in \sigma(A)} |\bar{p}_k(z)|.$$

□

No teorema acima supomos que A é uma matriz diagonalizável. Mas como o conjunto das matrizes não diagonalizáveis é um conjunto muito pequeno no espaço das matrizes $n \times n$, quase toda as matrizes são diagonalizáveis. Então, voltaremos nossa atenção apenas para estas matrizes.

Utilizando o resultado acima podemos demonstrar o seguinte teorema.

Teorema 2.5 *Seja A uma matriz não singular diagonalizável. Se A possui apenas k autovalores distintos, então, na ausência de erros de arredondamento, o GMRES encontrará a solução no máximo em k iterações.*

Demonstração :

De fato, sendo $\{\lambda_1, \dots, \lambda_k\}$ o conjunto com os k autovalores distintos, então podemos definir

$$\bar{p}_k(z) = \prod_{i=1}^k \frac{(\lambda_i - z)}{\lambda_i}.$$

Observe que $\bar{p}_k(z)$ tem grau k e $\bar{p}_k(0) = 1$; portanto este é um polinômio residual e $\bar{p}_k(z) = 0$ para todo $z \in \sigma(A)$. Assim, utilizando o teorema 2.4 temos que

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \kappa_2(V) \max_{z \in \sigma(A)} |\bar{p}_k(z)| = 0.$$

Logo, $\|r_k\|_2 = 0$, ou seja, $r_k = b - Ax_k = 0$. Portanto $x_k = x_*$ é a solução do sistema linear $Ax = b$. \square

Com este resultado temos que o GMRES converge com um número finito de passos, embora seu desempenho seja melhor quando visto como um método iterativo e não como um método direto.

2.2 Processo de Arnoldi

Consideremos V_k a matriz de projeção ortogonal sobre \mathcal{K}_k ; então $z \in \mathcal{K}_k$ pode ser escrito como

$$z = \sum_{l=1}^k y_l v_k^l$$

onde v_k^l é a l -ésima coluna de V_k . Logo, podemos converter (5) no problema de obter o vetor de coeficientes y , o qual é um problema de quadrados mínimos em \mathbb{R}^k . Desde que

$$x - x_0 = V_k y$$

para algum $y \in \mathbb{R}^k$, então temos $x_k = x_0 + V_k y$, onde y minimiza

$$\|b - A(x_0 + V_k y)\|_2 = \|r_0 - AV_k y\|_2.$$

Logo, nosso problema de quadrados mínimos em \mathbb{R}^k é

$$\min_{y \in \mathbb{R}^k} \|r_0 - AV_k y\|_2, \tag{9}$$

que é um problema padrão de quadrados mínimos e pode ser resolvido, por exemplo, por fatoração QR [16] e [21].

Antes de prosseguirmos nossos estudos vamos discutir um método para obter a matriz V_k . Como vimos anteriormente, as colunas de V_k formam uma base ortonormal para \mathcal{K}_k . Para obter esta base podemos utilizar o processo de Gram-Schmidt [21], que neste caso particular é conhecido como processo de Arnoldi. Neste trabalho utilizaremos somente o processo de Arnoldi usual para gerar uma base ortonormal para \mathcal{K}_k , embora existam trabalhos que usam outros métodos, como, por exemplo, o de H. F. Walker [20], que utiliza a transformação de Householder e o de A. Essai [9], que utiliza um produto interno com pesos ($\langle u, v \rangle = \sum_{i=1}^n d_i u_i v_i$ com $d_i > 0 \forall i \in \{1, \dots, n\}$) para gerar tal base.

No processo de Arnoldi fornecemos como entrada os vetores x_0 e b , a matriz A , e a dimensão k . O algoritmo calcula uma base ortonormal para \mathcal{K}_k , armazenando-a nas colunas da matriz V .

Algoritmo de Arnoldi - $\text{arnoldi}(x_0, b, A, k, V)$

1. Defina $r_0 = b - Ax_0$ e $v_1 = r_0 / \|r_0\|_2$.
2. Para $i = 1, \dots, k - 1$

$$v_{k+1} = \frac{Av_i - \sum_{j=1}^i ((Av_i)^t v_j) v_j}{\|Av_i - \sum_{j=1}^i ((Av_i)^t v_j) v_j\|_2}.$$

Se não ocorrer divisão por zero no passo 2 do algoritmo de Arnoldi, então as colunas de V_k formam uma base ortonormal para \mathcal{K}_k . A divisão por zero, referida como "breakdown", ocorre somente se a solução de $Ax = b$ está no subespaço \mathcal{K}_{k-1} .

Lema 2.6 *Seja A uma matriz não singular. Considere os vetores v_j gerados pelo processo de Arnoldi, e seja i o menor inteiro tal que*

$$Av_i - \sum_{j=1}^i ((Av_i)^t v_j) v_j = 0.$$

Então $x = A^{-1}b \in x_0 + \mathcal{K}_i$.

Demonstração :

Por hipótese $Av_i \in \mathcal{K}_i$ e portanto $A\mathcal{K}_i \subset \mathcal{K}_i$. Desde que as colunas de V_i formam uma base ortonormal para \mathcal{K}_i então

$$AV_i = V_i H,$$

onde H é uma matriz $i \times i$. Sabemos que H é não singular pois A é não singular. Tomando $\beta = \|r_0\|_2$ e $e_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^i$ então temos

$$\|r_i\|_2 = \|b - Ax_i\|_2 = \|r_0 - A(x_i - x_0)\|_2.$$

Mas $x_i - x_0 \in \mathcal{K}_i$ e portanto existe $y \in \mathbb{R}^i$ tal que $x_i - x_0 = V_i y$. Como $r_0 = \beta V_i e_1$ e V_i é uma matriz ortogonal, então

$$\|r_i\|_2 = \|V_i(\beta e_1 - Hy)\|_2 = \|\beta e_1 - Hy\|_{\mathbb{R}^{k+1}},$$

onde $\|\cdot\|_{\mathbb{R}^{k+1}}$ denota a norma Euclidiana no \mathbb{R}^{k+1} . Tomando $y = \beta H^{-1} e_1$ temos que $r_i = 0$ pela propriedade de minimização. \square

2.3 O Algoritmo GMRES

Se o processo de Arnoldi não falhar, então podemos usá-lo na implementação do método GMRES de forma muito eficiente. Tomando $h_{ij} = (Av_i)^T v_j$, pelo processo de Gram-Schmidt temos que a matriz H_k é uma matriz de Hessenberg superior, isto é, $h_{ij} = 0$ sempre que $i > j + 1$. Desta forma o processo de Arnoldi produz uma matriz ortogonal tal que

$$AV_k = V_{k+1}H_k. \quad (10)$$

Logo, para algum $y^k \in \mathbb{R}^k$,

$$r_k = b - Ax_k = r_0 - A(x_k - x_0) = V_{k+1}(\beta e_1 - H_k y^k).$$

Assim, $x_k = x_0 + V_k y^k$, onde y^k minimiza $\|\beta e_1 - H_k y\|_2$ sobre \mathbb{R}^k . Note que quando y^k é calculado, a norma de r_k pode ser encontrada sem explicitarmos x_k e $r_k = b - Ax_k$, ou seja, utilizando a ortogonalidade de V_k temos que

$$\|r_k\|_2 = \|V_{k+1}(\beta e_1 - H_k y^k)\|_2 = \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}. \quad (11)$$

Por esta razão, terminaremos as iterações quando encontrarmos um vetor x_k tal que

$$\|b - Ax_k\|_2 \leq \epsilon \|b\|_2.$$

para algum $\epsilon > 0$ pequeno.

Finalmente, dada a aproximação inicial x , o vetor do lado direito da equação b , a matriz A , a precisão $\epsilon > 0$ e o número máximo de iterações k_{max} , então podemos escrever o algoritmo

Algoritmo GMRES - gmresa($x, b, A, \epsilon, k_{max}$)

1. Defina $r = b - Ax$, $\rho = \|r\|_2$, $v_1 = r/\rho$, $\beta = \rho$ e $k = 0$.
2. Enquanto $\rho \geq \epsilon \|b\|_2$ e $k \leq k_{max}$ faça
 - (a) $k = k + 1$
 - (b) Para $j = 1, \dots, k$
 $h_{jk} = (Av_k)^T v_j$
 - (c) $v_{k+1} = Av_k - \sum_{j=1}^k h_{jk} v_j$
 - (d) $h_{k+1,k} = \|v_{k+1}\|_2$
 - (e) $v_{k+1} = v_{k+1}/\|v_{k+1}\|_2$
 - (f) $e_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^{k+1}$;
minimize $\|\beta e_1 - H_k y\|_{\mathbb{R}^{k+1}}$ sobre \mathbb{R}^k para obter y^k

$$(g) \quad \rho = \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}.$$

$$3. \quad x_k = x_0 + V_k y^k.$$

Note também que as bases para o subespaço de Krylov são armazenadas a cada iteração. Deste modo, na k -ésima iteração do método armazenamos k vetores de dimensão n . Esta é uma desvantagem, pois para sistemas grandes podemos ter problemas com alocação de memória e, além disso, na k -ésima iteração estaremos realizando da ordem de $(1/2)k^2n$ multiplicações. Para evitarmos este tipo de problema podemos reiniciar o método após um número m de iterações. O método GMRES com reinicialização é chamado de GMRES(m) e sua convergência é mais lenta que o GMRES usual. Neste trabalho não discutiremos o método GMRES(m), que está descrito com detalhes em [11] e [19].

Outro problema apresentado pelo algoritmo **gmresa** é quanto à ortogonalidade dos vetores v_j , ou seja, os vetores v_j podem ser não ortogonais devido a erros de arredondamento. Quando isto acontece, perderemos a precisão do método. Uma forma de evitarmos este problema é substituindo o processo de Gram-Schmidt clássico pelo modificado, pois embora sejam teoricamente equivalentes, na prática o método de Gram-Schmidt modificado apresenta uma precisão maior quanto à ortogonalidade. Desta forma, alteramos o passo 2c do algoritmo **gmresa** para

$$\begin{aligned} v_{k+1} &= Av_k \\ \text{Para } j &= 1, \dots, k \\ v_{k+1} &= v_{k+1} - (v_{k+1}^T v_j) v_j. \end{aligned}$$

obtendo assim um algoritmo mais eficiente.

Esta versão do GMRES é muito simples e ilustra idéias importantes. A primeira delas é que x_k é calculado somente após terminadas as iterações e ρ pode ser utilizado como uma aproximação para a norma do resíduo do sistema linear, pois eles são idênticos na ausência de erros de arredondamento. Além disso, podemos compensar a perda da ortogonalidade para os vetores da base do subespaço de Krylov, acrescentando um teste para verificar se isto ocorre e reortogonalizar quando necessário. Por exemplo, podemos acrescentar, após o algoritmo de Gram-Schmidt modificado, um processo para a reortogonalização. Esta reortogonalização pode ser executada em todas as iterações ou somente se um teste detecta a perda de ortogonalidade. Se reortogonalizarmos a cada passo, então temos que o custo computacional para o cálculo de V é o dobro. Deste modo iremos propor um critério para detectar a perda de ortogonalidade e o processo de Gram-Schmidt com reortogonalização pode ser visto como

$$\begin{aligned} 1. \quad v_{k+1} &= Av_k \\ \text{Para } j &= 1, \dots, k \\ h_{jk} &= v_{k+1}^T v_j \\ v_{k+1} &= v_{k+1} - h_{jk} v_j \end{aligned}$$

2. $h_{k+1,k} = \|v_{k+1}\|_2$
3. Se a perda da ortogonalidade for detectada
 Para $j = 1, \dots, k$
 $h_{tmp} = v_{k+1}^T v_j$
 $h_{jk} = h_{jk} - h_{tmp} v_j$
 $v_{k+1} = v_{k+1} - h_{tmp} v_j$
4. $h_{k+1,k} = \|v_{k+1}\|_2$
5. $v_{k+1} = v_{k+1} / \|v_{k+1}\|_2$

Neste trabalho utilizaremos a condição de Brown/Hindmarsh para detectar a perda da ortogonalidade, como proposto em [11]. Assim, após o processo de Gram-Schmidt, iremos verificar se

$$\|Av_k\|_2 + \delta \|v_{k+1}\|_2 = \|Av_k\|_2 \quad (12)$$

e quando esta igualdade for satisfeita, reortogonalizaremos. Isto significa que se o novo vetor é muito pequeno comparado com Av_k , podemos estar perdendo informações e, então, é necessário realizarmos o segundo passo do processo de Gram-Schmidt novamente. Neste trabalho utilizaremos $\delta = 10^{-3}$ no teste para reortogonalização.

2.4 GMRES com Rotações de Givens

Para terminarmos nosso estudo do GMRES, vamos discutir um método para a resolução do problema de quadrados mínimos encontrado no passo (e) do `gmres`. Para a resolução deste problema, podemos utilizar rotações de Givens para obter a decomposição QR da matriz H_k , ou seja, $H_k = Q_k R_k$ [21].

Assim podemos escrever o algoritmo `gmres` utilizando rotações de Givens para a resolução do problema de quadrados mínimos e (12) como um teste para detectar a perda da ortogonalidade.

Algoritmo GMRES - `gmres`($x, b, A, \epsilon, k_{max}$)

1. Defina $r = b - Ax$, $v_1 = r / \|r\|_2$, $\rho = \|r\|_2$, $\beta = \rho$, $k = 0$ e $g = \rho(1, 0, 0, \dots, 0)^t \in \mathbb{R}^{k_{max}+1}$.
2. Enquanto $\rho \geq \epsilon \|b\|_2$ e $k \leq k_{max}$ faça
 - (a) $k = k + 1$

- (b) $v_{k+1} = Av_k$
 Para $j = 1, \dots, k$
- i. $h_{jk} = v_{k+1}^T v_j$
 - ii. $v_{k+1} = v_{k+1} - h_{jk} v_j$
- (c) $h_{k+1,k} = \|v_{k+1}\|_2$
- (d) Realizar teste para detectar a perda da ortogonalidade e reortogonalizar se necessário.
- (e) $v_{k+1} = v_{k+1} / \|v_{k+1}\|_2$
- (f)
 - i. Se $k > 1$ aplicar Q_{k-1} sobre a k -ésima coluna de H .
 - ii. $\nu = \sqrt{h_{k,k}^2 + h_{k+1,k}^2}$.
 - iii. $c_k = h_{k,k} / \nu$, $s_k = -h_{k+1,k} / \nu$.
 - iv. $h_{k,k} = c_k h_{k,k} - s_k h_{k+1,k}$, $h_{k+1,k} = 0$.
 - v. $g = G_k(c_k, s_k)g$.
- (g) $\rho = |(g)_{k+1}|$.
3. Faça $r_{i,j} = r_{i,j}$ para $1 \leq i, j \leq k$.
 Faça $(w)_i = (g)_i$ para $1 \leq i \leq k$.
 Resolva o sistema triangular superior $Ry^k = w$.
4. $x_k = x_0 + V_k y^k$.

3 Método de Newton Inexato

O método de Newton Inexato consiste em resolver o sistema (4) utilizando um método iterativo para obter a solução aproximada dos sistemas lineares gerados a cada iteração. Um critério de parada usual para métodos iterativos é baseado no resíduo relativo $\|r_k\| / \|F(x_k)\|$, onde o resíduo r_k é dado por

$$r_k \equiv J(x_k)s_k + F(x_k).$$

Assim, calcularemos uma aproximação da solução satisfazendo

$$\frac{\|r_k\|}{\|F(x_k)\|} \leq \eta_k, \quad (13)$$

onde a seqüência não negativa $\{\eta_k\}$ é usada para controlar o nível de precisão e é conhecida como seqüência forçante ([6] e [8]). Note que η_k pode depender de x_k e se tomarmos $\eta_k \equiv 0$ temos o método de Newton.

Resumindo, um método de Newton Inexato é um método que, dada uma aproximação inicial x_0 , gera uma seqüência $\{x_k\}$ para aproximar x_* por:

$$\begin{aligned}
J(x_k)s_k &= -F(x_k) + r_k, \\
\text{onde } \|r_k\| &\leq \eta_k \|F(x_k)\|, \\
x_{k+1} &= x_k + s_k.
\end{aligned} \tag{14}$$

Assumindo que a seqüência $\{\eta_k\}$ é uniformemente menor que 1, então podemos mostrar que o método de Newton Inexato é localmente convergente; para isto, vamos enunciar os seguintes lemas¹:

Lema 3.1 *Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uma função não linear satisfazendo as seguintes propriedades:*

- (1) *Existe pelo menos um $x_* \in \mathbb{R}^n$, com $F(x_*) = 0$.*
- (2) *F é continuamente diferenciável em uma vizinhança Ω de x_* .*
- (3) *$J(x_*)$ é não singular.*

Então, dado $\epsilon > 0$, existe $\delta > 0$ tal que $J(y)$ é não singular e

$$\|J(y)^{-1} - J(x_*)^{-1}\| \leq \epsilon$$

sempre que $\|y - x_\| \leq \delta$.*

Lema 3.2 *Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uma função satisfazendo as propriedades do lema 3.1. Então, dado $\epsilon > 0$, existe $\delta > 0$ tal que*

$$\|F(y) - F(x_*) - J(x_*)(y - x_*)\| \leq \epsilon \|y - x_*\|$$

sempre que $\|y - x_\| \leq \delta$.*

Teorema 3.3 *Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uma função não linear satisfazendo as propriedades do lema 3.1:*

Assuma que $\eta_k \leq \eta_{max} < \sigma < 1$. Assim, existe $\epsilon > 0$ tal que, se $\|x_0 - x_\| \leq \epsilon$, então a seqüência $\{x_k\}$ do método de Newton Inexato converge para x_* satisfazendo*

$$\|x_{k+1} - x_*\|_* \leq \sigma \|x_k - x_*\|_*, \tag{15}$$

onde $\|y\|_ \equiv \|J(x_*)y\|$.*

Demonstração :

Sendo $J(x_*)$ não singular, então

$$\frac{1}{\mu} \|y\| \leq \|y\|_* \leq \mu \|y\| \quad \text{para } y \in \mathbb{R}^n. \tag{16}$$

onde

$$\mu = \max\{\|J(x_*)\|, \|J(x_*)^{-1}\|\}.$$

¹Os lemas apresentados a seguir, com suas respectivas demonstrações, se encontram em [17]

Desde que $\eta_{max} < \sigma$, então existe $\gamma > 0$ suficientemente pequeno de modo que

$$(1 + \gamma\mu)[\eta_{max}(1 + \mu\gamma) + 2\mu\gamma] \leq \sigma.$$

Agora, escolhemos $\epsilon > 0$ suficientemente pequeno tal que

$$\|J(y) - J(x_*)\| \leq \gamma, \quad (17)$$

$$\|J(y)^{-1} - J(x_*)^{-1}\| \leq \gamma, \quad (18)$$

$$\|F(y) - F(x_*) - J(x_*)(y - x_*)\| \leq \gamma\|y - x_*\| \quad (19)$$

sempre que $\|y - x_*\| \leq \mu^2\epsilon$. Tal ϵ existe em virtude da continuidade de $J(x)$ em x_* , pelo lema 3.1 e pelo lema 3.2.

Assuma que $\|x_0 - x_*\| \leq \epsilon$. Nós provaremos (15) por indução. Note que, por (16), a hipótese de indução e (16) novamente,

$$\|x_k - x_*\| \leq \mu\|x_k - x_*\|_* \leq \mu\sigma^n\|x_0 - x_*\|_* \leq \mu^2\|x_0 - x_*\| \leq \mu^2\epsilon,$$

e portanto (17) - (19) valem com $y = x_k$. Lembrando que a n -ésima iteração do método de Newton Inexato é definida de modo que s_k satisfaça (14) e que

$$J(x_*)(x_{k+1} - x_*) =$$

$$[I + J(x_*)[J(x_k)^{-1} - J(x_*)^{-1}] \cdot [r_k + [J(x_k) - J(x_*)](x_k - x_*) - [F(x_k) - F(x_*) - J(x_*)(x_k - x_*)]],$$

temos, tomando a norma $\|\cdot\|$,

$$\begin{aligned} \|(x_{k+1} - x_*)\| &= \\ [I + \|J(x_*)\| \|J(x_k)^{-1} - J(x_*)^{-1}\|] \cdot [\|r_k\| + \|J(x_k) - J(x_*)\| \|x_k - x_*\| - \|F(x_k) - F(x_*) - J(x_*)(x_k - x_*)\|] \\ &\leq (1 + \mu\gamma)[\eta_k \|F(x_k)\| + \gamma\|x_k - x_*\| + \gamma\|x_k - x_*\|], \end{aligned}$$

onde usamos a definição de μ e as equações (18), (14), (17) e (19). Mas

$$F(x_k) = [J(x_*)(x_k - x_*)] + [F(x_k) - F(x_*) - J(x_*)(x_k - x_*)],$$

e portanto

$$\|F(x_k)\| \leq \|x_k - x_*\|_* + \|F(x_k) - F(x_*) - J(x_*)(x_k - x_*)\| \leq \|x_k - x_*\|_* + \gamma\|x_k - x_*\|,$$

onde utilizamos (19). Finalmente, por (16), obtemos

$$\begin{aligned} \|x_{k+1} - x_*\|_* &\leq (1 + \mu\gamma)[\eta_k[\|x_k - x_*\|_* + \gamma\|x_k - x_*\|] + 2\gamma\|x_k - x_*\|] \\ &\leq (1 + \mu\gamma)[\eta_{max}(1 + \mu\gamma) + 2\mu\gamma]\|x_k - x_*\|_* \end{aligned}$$

O resultado agora segue pela escolha de γ . \square

Com este teorema mostramos que o método de Newton Inexato é localmente convergente. Agora vamos caracterizar sua ordem de convergência, indicando a influência da seqüência forçante $\{\eta_k\}$ na convergência do método.

Teorema 3.4 *Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uma função não linear satisfazendo as propriedades do lema 3.1: Assuma que as iterações $\{x_k\}$ do método de Newton Inexato convergem para x_* . Então:*

- (a) $x_k \rightarrow x_*$ superlinearmente se $\lim_{k \rightarrow \infty} \eta_k = 0$;
(b) $x_k \rightarrow x_*$ com ordem $1 + p$ se J tem continuidade tipo Hölder² com expoente p em x_* e

$$\eta_k = \mathcal{O}(\|J(x_k)\|^p) \quad \text{quando} \quad k \rightarrow \infty.$$

A demonstração do teorema acima encontra-se em [6] e [11].

3.1 Método Newton-GMRES

O nome Newton-GMRES é dado para o método de Newton Inexato que utiliza o método GMRES para a resolução do sistema linear dado em (14). Nesta seção forneceremos uma análise do método de Newton-GMRES, iniciando nossa discussão com os efeitos da aproximação do Jacobiano por diferenças finitas.

Sabemos que o método GMRES requer o cálculo do produto da matrix $J(x)$ por um vetor w . Para aproximarmos este produto por diferenças finitas, utilizaremos a derivada direcional. Assim, trabalhando com o vetor nulo como aproximação inicial, podemos escrever o método GMRES para a resolução do sistema dentro do método de Newton Inexato.

Algoritmo - fdgmres($s, x, F, \eta, kmax, \rho$)

1. $s = 0, r = -F(x), v_1 = r/\|r\|_2, \rho = \|r\|_2, \beta = \rho, k = 0$
2. Enquanto $\rho > \eta\|F(x)\|_2$ e $k < kmax$ faça
 - (a) $k = k + 1$
 - (b) $v_{k+1} = D_h F(x : v_k)$
Para $j = 1, \dots, k$
 - i. $h_{jk} = v_{k+1}^T v_j$
 - ii. $v_{k+1} = v_{k+1} - h_{jk} v_j$
 - (c) $h_{k+1,k} = \|v_{k+1}\|_2$
 - (d) $v_{k+1} = v_{k+1}/\|v_{k+1}\|_2$
 - (e) $e_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^{k+1}$
Minimize $\|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}$ para obter $y^k \in \mathbb{R}^k$.
 - (f) $\rho = \|\beta e_1 - H_k y^k\|_{\mathbb{R}^{k+1}}$.

² J tem continuidade tipo Hölder com expoente p ($0 < p \leq 1$) em x_* se existe $L \geq 0$ tal que $\|J(y) - J(x_*)\| \leq L\|y - x_*\|^p$ para $\|y - x_*\|$ suficientemente pequeno.

3. $s = V_k y^k$.

Supondo que a função $F(x)$ é calculada sem erros, então podemos mostrar que o desempenho do método Newton-GMRES não é alterado se utilizamos diferenças finitas com um passo h pequeno.

Proposição 3.5 *Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uma função não linear satisfazendo as seguintes propriedades:*

- (1) *Existe pelo menos um $x_* \in \mathbb{R}$ com $F(x_*) = 0$.*
- (2) *$J : \Omega \rightarrow \mathbb{R}^{n \times n}$ é Lipschitz contínua com constante γ^3 .*
- (3) *$J(x_*)$ é não singular.*

Seja $\eta \in (0, 1)$. Então existem \bar{h} e δ tais que, se $x \in B(x_, \delta)$, $h \leq \bar{h}$, e o algoritmo `fdgmres` termina com $k < kmax$, então*

$$\|J(x)s - F(x)\|_2 < (\eta + 4\gamma h)\|F(x)\|_2. \quad (20)$$

Desta proposição segue o teorema:

Teorema 3.6 *Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uma função não linear satisfazendo as propriedades da proposição 3.5. Então existe δ tal que, se $x_0 \in B(x_*, \delta)$ e as seqüências $\{\eta_k\}$ e $\{h_k\}$ satisfazem*

$$\sigma_k = \eta_k + 4\gamma h_k \in [0, \bar{\sigma}]$$

com $0 < \bar{\sigma} < 1$, então as iterações do Newton-GMRES

$$x_{k+1} = x_k + s_k$$

onde s_k é calculado pelo algoritmo `fdgmres` com entrada

$$(s_k, x_k, F, h_k, \eta_k, kmax, \rho)$$

converge q -linearmente com respeito à norma $\|\cdot\|_$ e s_k satisfaz*

$$\|J(x_k)s_k - F(x_k)\|_2 < \sigma_k \|F(x_k)\|_2. \quad (21)$$

. Além disso,

- *se $\sigma_k \rightarrow 0$, então a convergência é q -superlinear, e*
- *se $\sigma_k \leq K_k \|F(x_k)\|^p$ para algum $K_k > 0$, então a convergência é q -superlinear com q -ordem $1 + p$.*

A demonstração da proposição 3.5 e do teorema 3.6 encontram-se em [11].

³ J tem continuidade tipo Lipschitz em x_* se existe $\gamma \geq 0$ tal que $\|J(y) - J(x_*)\| \leq \gamma \|y - x_*\|$ para $\|y - x_*\|$ suficientemente pequeno.

3.2 Implementação do Método Newton-GMRES

Em nossa implementação para o método de Newton-GMRES, utilizamos o software MATLAB (MATrix LABORatory) e a rotina `nsolgm` implementada por C. T. Kelley, que está disponível na página www.siam.org/pub/books/kelley. Esta rotina exige como entrada uma aproximação inicial, a função, um vetor contendo as tolerâncias para o critério de parada e um termo para formarmos a seqüência forçante $\{\eta_k\}$.

Nós ajustaremos η a cada iteração tomando inicialmente

$$\eta_k^A = \gamma \|F(x_k)\|^2 / \|F(x_{k-1})\|^2,$$

onde $\gamma \in (0, 1]$ é um parâmetro. Se η_k^A é uniformemente limitada por 1, então tomando $\eta_k = \eta_k^A$ para $k > 0$ temos garantida a convergência dada pelo teorema 3.3. Deste modo, definiremos um valor $\eta_{max} < 1$ para $k = 0$ e tomaremos

$$\eta_k^B = \begin{cases} \eta_{max}, & \text{se } k = 0, \\ \min(\eta_{max}, \eta_k^A), & \text{se } k > 0. \end{cases} \quad (22)$$

Note que $\eta_{max} < 1$ é um limitante superior para a seqüência $\{\eta\}$ e portanto, temos a convergência garantida pelo teorema 3.3. Em nossas implementações, utilizaremos $\gamma = 0.9$ e $\eta_{max} = 0.9999$.

Mas, pode acontecer de que η_k^B seja pequeno para uma ou mais iterações quando x_k está próximo da solução. Um método sugerido em [8] para evitar isto consiste em avaliar se η_{k-1} é suficientemente grande; neste caso não tomamos η_k menor que um fator de η_{k-1} , ou seja,

$$\eta_k^C = \begin{cases} \eta_{max}, & \text{se } k = 0, \\ \min(\eta_{max}, \eta_k^A), & \text{se } k > 0, \gamma \eta_{k-1}^2 < 0.1, \\ \min(\eta_{max}, \max(\eta_k^A, \gamma \eta_{k-1}^2)), & \text{se } k > 0, \gamma \eta_{k-1}^2 > 0.1. \end{cases} \quad (23)$$

A constante 0.1 é um valor arbitrário.

Finalmente tomaremos

$$\eta_k = \min(\eta_{max}, \max(\eta_k^C, 0.5\tau_t/\|F(x_k)\|)), \quad (24)$$

onde

$$\tau_t = \tau_a + \tau_r \|F(x_0)\|,$$

é adicionado para evitar um trabalho desnecessário.

Utilizaremos a rotina `dirder` para aproximar a derivada direcional tomando $h = \sqrt{\textit{eps}}$, onde *eps* representa o zero da máquina. Em todos os testes computacionais utilizaremos $\gamma = 0.9$, como recomenda [8].

Algoritmo - nsolgm(x, F, τ, η)

1. $r_c = r_0 = \|F(x)\|_2$
2. Enquanto $\|F(x)\|_2 > \tau_r r_0 + \tau_a$ faça
 - (a) Escolha η .
 - (b) Execute `fdgmres`(s, x, F, η) obtendo s
 - (c) $x = x + s$
 - (d) Calcule $F(x)$
 - (e) $r_+ = \|F(x)\|_2, \sigma = r_+/r_c, r_c = r_+$
 - (f) Se $\|F(x)\|_2 \leq \tau_r r_0 + \tau_a$ então termine.

4 Testes Computacionais

Nesta seção apresentaremos os testes computacionais realizados. Os algoritmos utilizados foram `nsol` e `nsolgm` descritos anteriormente. Tanto na rotina `nsol` como na rotina `nsolgm`, utilizamos um número máximo de 40 iterações para o método de Newton. Para o GMRES, trabalhamos com $\eta_{max} = 0.9$ e um número máximo de 40 iterações. O Jacobiano foi aproximado por diferenças finitas e usamos $\tau_r = \tau_a = 10^{-6}$ no critério de parada.

Os resultados obtidos são apresentados por meio de tabelas e gráficos. Nas tabelas, a coluna F_{rel} registra os valores de $\|F(x_k)\|_2/\|F(x_0)\|_2$, em cada iteração k .

4.1 Equação de Chandrasekhar

No primeiro teste computacional, abordamos a equação de Chandrasekhar dada por

$$F(H)(\mu) = H(\mu) - \left(1 - \frac{c}{2} \int_0^1 \frac{\mu H(\nu) d\nu}{\mu + \nu}\right)^{-1} = 0, \quad (25)$$

que é utilizada para resolver problemas de distribuição de transferência radiativa [3].

Aproximamos a integral sobre o intervalo $[0, 1]$ pela regra do ponto médio, dada por:

$$\int_0^1 f(\mu) d\mu \sim \frac{1}{N} \sum_{j=1}^N f(\mu_j),$$

onde $\mu_j = (j - 1/2)/N$ para $1 \leq j \leq N$. O resultado do problema discretizado é

$$F(x)_i = x_i - \left(1 - \frac{c}{2N} \sum_{j=1}^N \frac{\mu_i x_j}{\mu_i + \mu_j}\right)^{-1}. \quad (26)$$

Sabemos que tanto (25) quanto (26) possuem duas soluções para $c \in (0, 1)$, embora apenas uma destas soluções possua uma interpretação física. Sabemos também que o método de Newton e o método de Newton Inexato, com as aproximações iniciais $x_0 = (0, 0, \dots, 0)^T$ ou $x_0 = (1, 1, \dots, 1)^T$, convergem para a solução desejada [12].

Nós utilizamos $N = 200$, $c = 0.9$ e o vetor unitário como aproximação inicial. Com estes valores comparamos o método de Newton com o método de Newton-GMRES obtendo os resultados apresentados na tabela 1 e nas figuras 1 e 2.

Iteração	F_{rel}	Iteração	F_{rel}	N. Iter. GMRES
0	4.5725e+000	0	3.2332e-001	0
1	5.0258e-001	1	4.6585e-002	1
2	8.5005e-003	2	5.4381e-003	1
3	2.4122e-006	3	3.8567e-005	2
		4	5.0628e-007	2

Tabela 1: Métodos de Newton e Newton-GMRES para a equação de Chandrasekhar

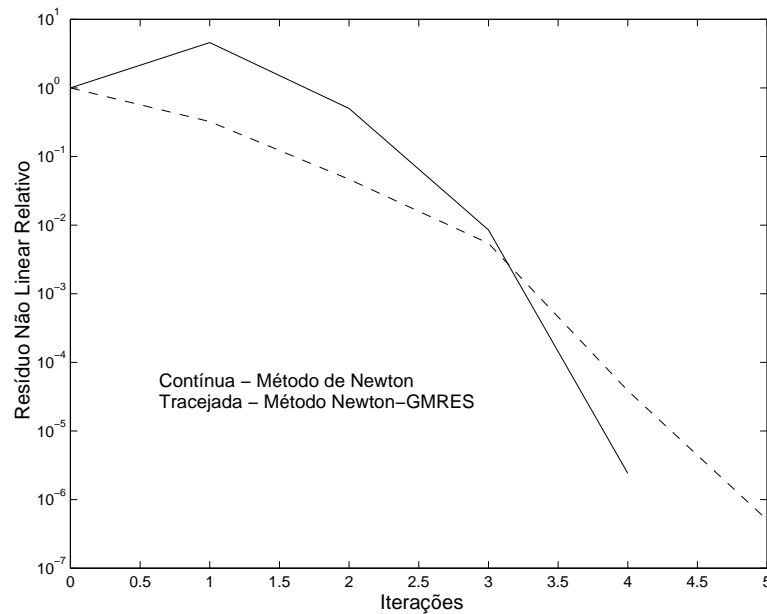


Figura 1: Métodos de Newton e Newton-GMRES para a equação de Chandrasekhar

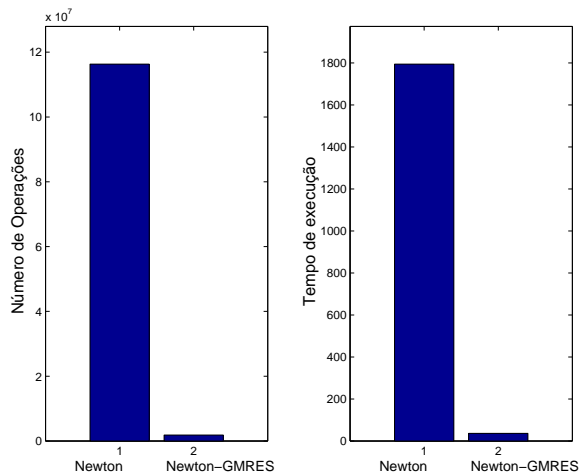


Figura 2: Métodos de Newton e Newton-GMRES para a equação de Chandrasekhar

4.2 Sistema Tridiagonal de Broyden

O sistema tridiagonal de Broyden, apresentado em [2] e [14], é definido por:

$$\begin{cases} f_1(x) = (3 - 2x_1)x_1 - 2x_2 + 1 \\ f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1 & i = 2, \dots, n - 1 \\ f_n(x) = (3 - 2x_n)x_n - x_{n-1} + 1 \end{cases} \quad (27)$$

Trabalhamos com $n = 500$, $x = (-1, -1, \dots, -1)^T$ como aproximação inicial e obtivemos os resultados apresentados na tabela 2 e nas figuras 3 e 4.

Iteração	F_{rel}	N. Iter. GMRES
0	2.2605e+001	0
1	2.8767e+000	1
2	8.5112e-002	1
3	1.3101e-004	1
4	1.2387e-009	3
5	6.7333e-005	6
6	4.6338e-007	7

Tabela 2: Métodos de Newton e Newton-GMRES para o sistema de Broyden

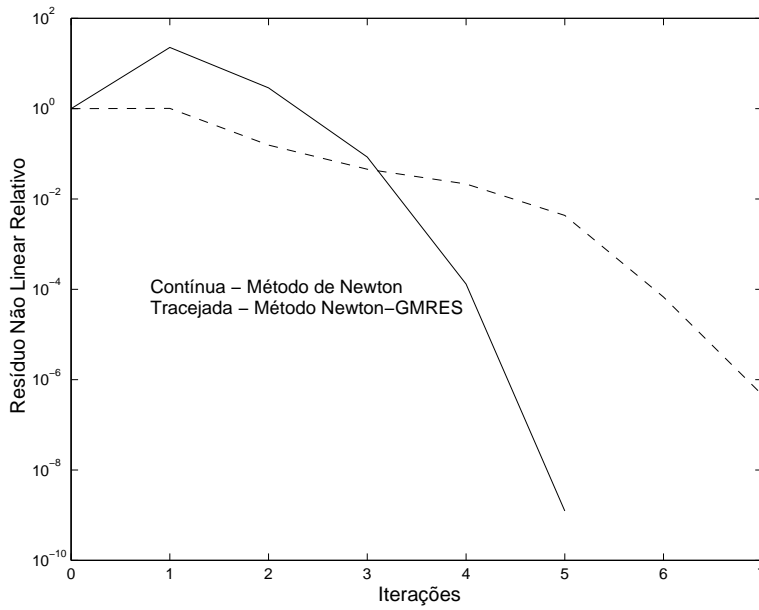


Figura 3: Métodos de Newton e Newton-GMRES para o sistema de Broyden

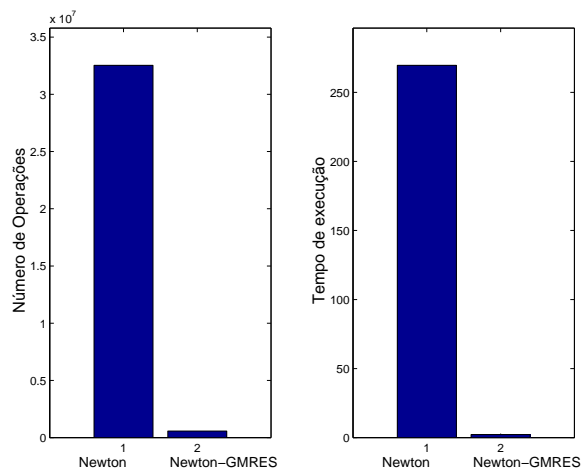


Figura 4: Métodos de Newton e Newton-GMRES para o sistema de Broyden

4.3 Problema de Contorno Não Linear

Consideramos o seguinte problema de contorno:

$$u''(t) = \frac{1}{2}(u(t) + t + 1)^3, \quad 0 < t < 1,$$

$$u(0) = u(1) = 0.$$

Se utilizarmos uma discretização com ordem $O(h^2)$, ou seja, tomando

$$u''(t) \sim \frac{u(t+h) - 2u(t) + u(t-h)}{h^2},$$

obtemos como resultado o sistema não linear definido por:

$$f_k(x) = 2x_k - x_{k+1} - x_{k-1} + \frac{h^2}{2}(x_k + t_k + 1)^3, \quad 1 \leq k \leq n, \quad (28)$$

onde $h = 1/(n+1)$, $t_k = kh$ e $x_k = u(t_k)$ com $x_0 = x_{n+1} = 0$.

Sabemos que o sistema (28) possui uma única solução x_* com $0 \leq x_i^* \leq 0.5$ para $1 \leq i \leq n$ [13].

Para resolver este problema, tomamos $n = 500$, e um número máximo de 100 iterações para o método GMRES. Fornecemos

$$x = \xi_i, \quad \text{com} \quad \xi_i = t_i(t_i - 1), \quad 1 \leq i \leq n,$$

como aproximação inicial e encontramos os resultados apresentados na tabela 3 e nas figuras 5 e 6.

Iteração	F_{rel}
0	2.9055e-002
1	3.6099e-006
2	2.1237e-009

Iteração	F_{rel}	N. Iter. GMRES
0	1.2994e-003	0
1	4.1331e-004	1
2	2.4233e-004	1
3	1.0926e-004	2
4	2.1129e-005	9
5	5.9899e-006	100
6	4.1987e-006	100
7	3.1258e-006	100
8	2.3876e-006	100
9	1.8345e-006	100
10	1.4170e-006	100
11	1.0957e-006	100
12	8.4903e-007	100

Tabela 3: Métodos de Newton e Newton-GMRES para o problema de contorno

Note que a convergência do método Newton-GMRES torna-se lenta a partir da quinta iteração do método de Newton Inexato. Isto provavelmente ocorre devido ao mau-condicionamento do Jacobiano perto da solução do sistema. Para contornar este problema poderíamos utilizar um pré-condicionador para o GMRES.

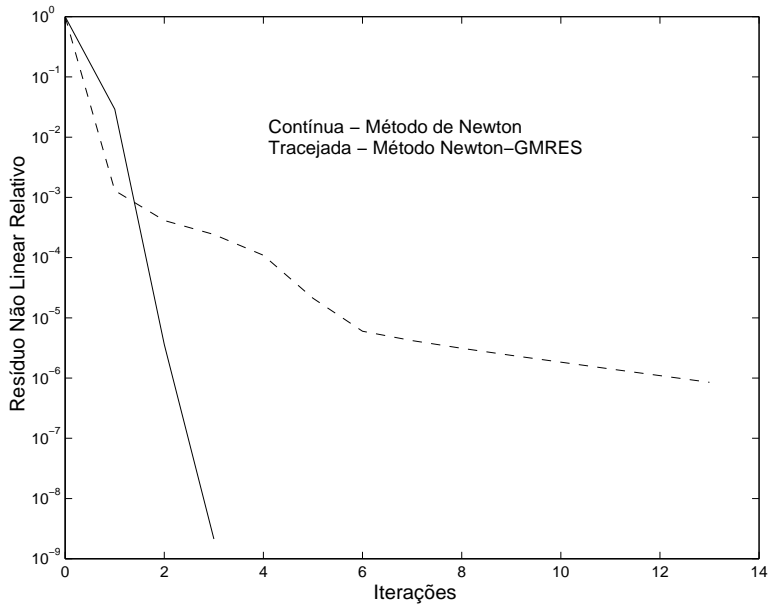


Figura 5: Métodos de Newton e Newton-GMRES para o problema de contorno

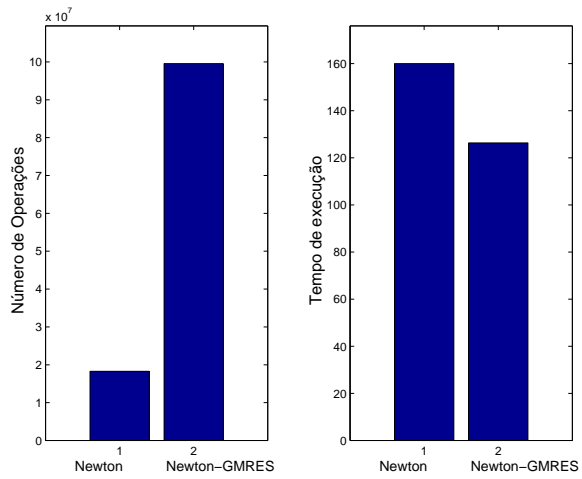


Figura 6: Métodos de Newton e Newton-GMRES para o problema de contorno

4.4 Função Trigexp de Toint

Neste teste computacional consideramos o seguinte sistema de equações :

$$f_1(x) = 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2) \sin(x_1 + x_2),$$

$$f_i(x) = -x_{i-1}e^{x_{i-1}-x_i} + x_i(4 + 3x_i^2) + 2x_{i+1} + \sin(x_i - x_{i+1}) \sin(x_i + x_{i+1}) - 8$$

$$\text{para } i = 2, 3, \dots, n-1,$$

$$f_n(x) = -x_{n-1}e^{x_{n-1}-x_n} + 4x_n - 3.$$

Sabemos que este sistema possui uma solução única [14]. Utilizando $n = 500$ e $x_0 = (0, 0, \dots, 0)^T$, obtivemos os resultados apresentados na tabela 4 e nas figuras 7 e 8.

Iteração	F_{rel}
0	1.7862e+002
1	5.9964e+002
2	1.6681e+002
3	4.7119e+001
4	1.3652e+001
5	3.2319e+000
6	4.5064e-001
7	1.5291e-002
8	1.9942e-005

Iteração	F_{rel}	N. Iter. GMRES
0	7.9882e+000	0
1	1.2285e+001	1
2	2.5148e+000	1
3	2.1791e-001	1
4	1.7556e-002	1
5	7.3809e-004	2
6	1.2914e-006	5

Tabela 4: Métodos de Newton e Newton-GMRES para a função Trigexp

4.5 Função de Rosenbrock Generalizada

Consideramos agora o seguinte sistema de equações :

$$f_{2i-1}(x) = 10(x_{2i} - x_{2i-1}^2),$$

$$f_{2i}(x) = 1 - x_{2i-1},$$

para $i = 1, 2, \dots, n$, com n par.

Sabemos que este sistema de equações possui uma solução para $x = (1, 1, \dots, 1)^T$ [14]. Utilizando $n = 500$, $x = (\xi_j)$, onde $\xi_{2j-1} = -1.2$ e $\xi_{2j} = 1$, como aproximação inicial, obtivemos os resultados apresentados na tabela 5 e nas figuras 9 e 10.

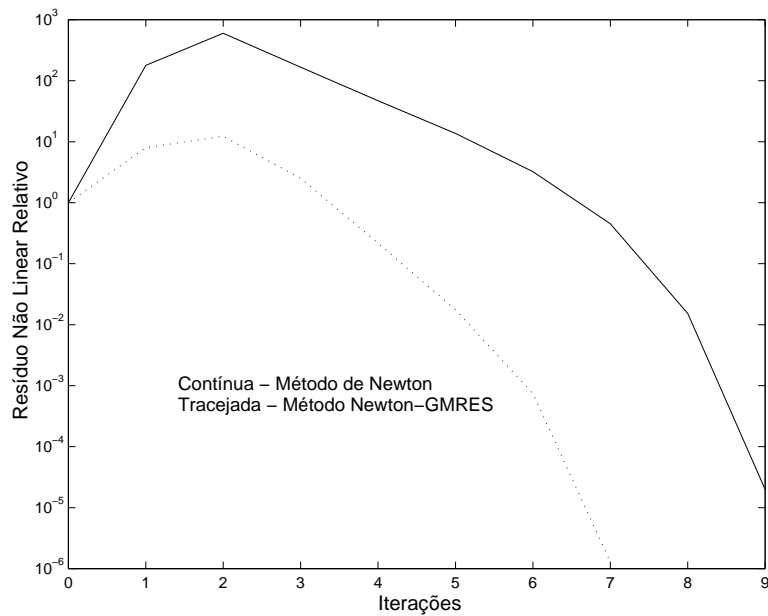


Figura 7: Métodos de Newton e Newton-GMRES para a função Trigexp

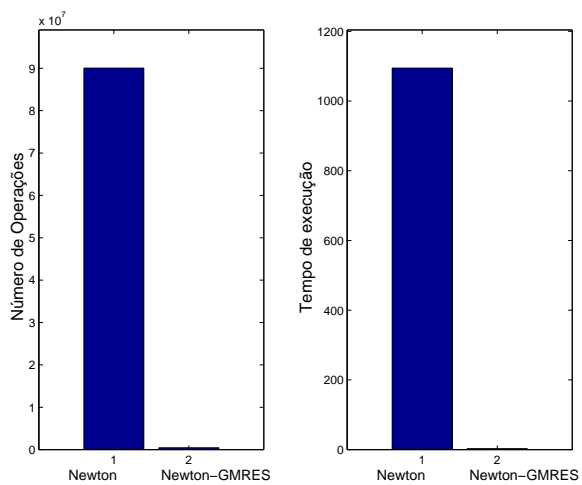


Figura 8: Métodos de Newton e Newton-GMRES para a função Trigexp

Iteração	F_{rel}
0	7.7782e+001
1	7.6527e+002
2	2.3833e-007

Iteração	F_{rel}	N. Iter. GMRES
0	3.4785e+000	0
1	1.4254e+000	1
2	2.7247e+001	2
3	2.6220e+001	1
4	3.0195e+001	1
5	2.4682e+001	1
6	4.4247e+001	1
7	2.2020e+001	1
8	6.8455e+000	2
9	1.7002e+000	1
10	1.6485e+000	2
11	1.2924e-001	1
12	9.5605e-002	2
13	4.8011e-003	1
14	3.2155e-004	2
15	3.4598e-012	2

Tabela 5: Métodos de Newton e Newton-GMRES para a função de Rosenbrock

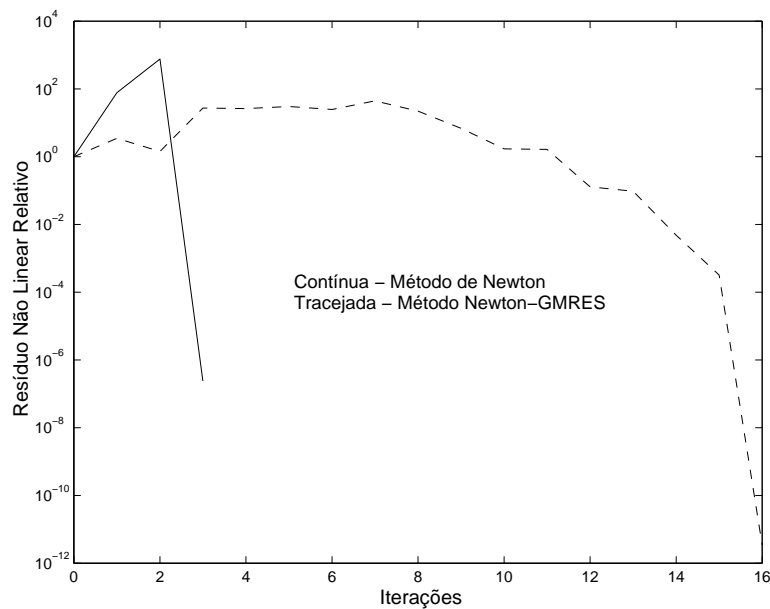


Figura 9: Métodos de Newton e Newton-GMRES para a função de Rosenbrock

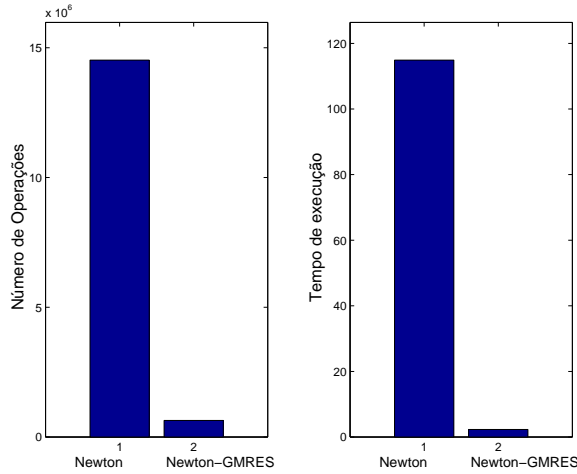


Figura 10: Métodos de Newton e Newton-GMRES para a função de Rosenbrock

4.6 Função Singular de Powell

Trabalhamos aqui o sistema de equações :

$$\begin{aligned}
 f_{4i-3}(x) &= x_{4i-3} + 10x_{4i-2}, \\
 f_{4i-2}(x) &= \sqrt{5}(x_{4i-1} - x_{4i}), \\
 f_{4i-1}(x) &= (x_{4i-2} - 2x_{4i-1})^2, \\
 f_{4i}(x) &= \sqrt{10}(x_{4i-3} - x_{4i})^2,
 \end{aligned}$$

para $i = 1, 2, \dots, n$, com n múltiplo de 4. Neste sistema temos que o Jacobiano é singular na solução [14].

Sabemos que este sistema de equações possui uma solução na origem e utilizando $n = 500$, $x = (\xi_j)$, onde $\xi_{4j-3} = 3$, $\xi_{4j-2} = -1$, $\xi_{4j-1} = 0$, $\xi_{4j} = 1$, como aproximação inicial, obtivemos os resultados apresentados na tabela 6 e nas figuras 11 e 12.

Devemos notar que, neste teste, o método de Newton terminou as iterações com $F_{rel} = 1.35 \times 10^{-4}$ e o método de Newton-GMRES com $F_{rel} = 5.46 \times 10^{-6}$. Isto acontece pois utilizamos como critério de parada a seguinte condição:

$$\|F(x)\|_2 \leq \tau_r \|F(x_0)\|_2 + \tau_a,$$

e como $\tau_r = \tau_a = 10^{-6}$, $\|F(x_0)\|_2 = 163.94$ então terminamos as iterações quando

$$\|F(x)\|_2 \leq 1.65 \times 10^{-4},$$

como ocorreu em ambos os métodos.

Iteração	F_{rel}
0	1.6394e+002
1	3.5466e+001
2	8.8664e+000
3	2.2166e+000
4	5.5415e-001
5	1.3854e-001
6	3.4634e-002
7	8.6586e-003
8	2.1647e-003
9	5.4116e-004
10	1.3529e-004

Iteração	F_{rel}	N. Iter. GMRES
0	7.3314e+000	0
1	4.9561e+000	1
2	1.2921e+000	3
3	6.5481e-001	3
4	8.9458e-002	4
5	2.2364e-002	4
6	5.5911e-003	4
7	1.3978e-003	4
8	3.4944e-004	4
9	8.7361e-005	4
10	2.1840e-005	4
11	5.4601e-006	4

Tabela 6: Métodos de Newton e Newton-GMRES para a função de Powell

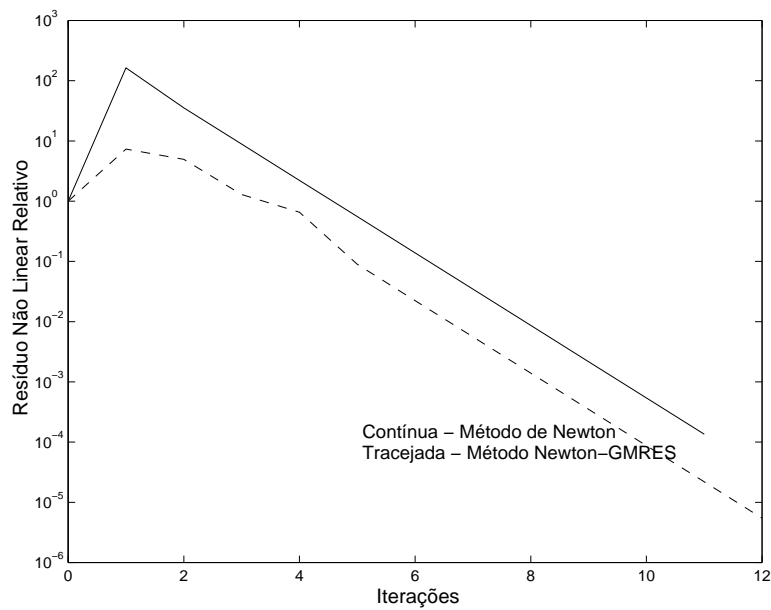


Figura 11: Métodos de Newton e Newton-GMRES para a função de Powell

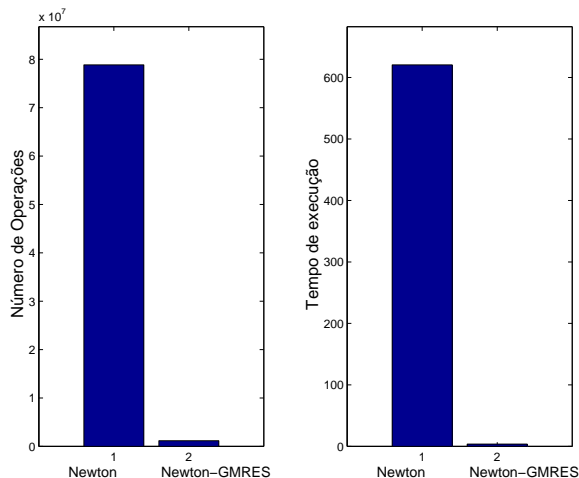


Figura 12: Métodos de Newton e Newton-GMRES para a função de Powell

4.7 Função Quase-Linear de Brown

O sistema quase-linear de Brown é descrito por:

$$f_i(x) = x_i + \sum_{j=1}^n x_j - (n + 1), \quad 1 \leq i < n,$$

$$f_n(x) = \left(\prod_{j=1}^n x_j \right) - 1.$$

Sabemos que este sistema possui uma solução $x_* = (\alpha, \alpha, \dots, \alpha, \alpha^{1-n})^T$ onde α satisfaz $n\alpha^n - (n + 1)\alpha^{n-1} + 1 = 0$. Utilizando $n = 500$ e $x_0 = (1/2, 1/2, \dots, 1/2)^T$ como aproximação inicial, obtivemos os resultados apresentados na tabela 7 e nas figuras 13 e 14.

Devemos observar que o método de Newton não convergiu e o resíduo relativo não linear ficou indeterminado a partir da primeira iteração. Isto acontece porque o Jacobiano é singular (a última linha é um vetor de zeros) em $x_* = (1, 1, \dots, 1)^T$, que é a solução para a qual o método deveria convergir. No entanto, o método Newton-GMRES não apresentou dificuldades.

5 Conclusão

Com base nos testes realizados percebemos que o método de Newton-GMRES obteve um desempenho melhor que o método de Newton; realizou menor número de operações, gastou muito menos tempo de execução e apresentou os menores resíduos.

Iteração	F_{rel}
0	5.5957e+003
1	Inf
2	Inf
⋮	⋮
40	Inf

Iteração	F_{rel}	N. Iter. GMRES
0	2.5025e+002	0
1	7.8250e-003	1
2	2.2255e-003	2
3	6.0372e-004	2
4	1.5828e-004	2

Tabela 7: Métodos de Newton e Newton-GMRES para a função de Brown

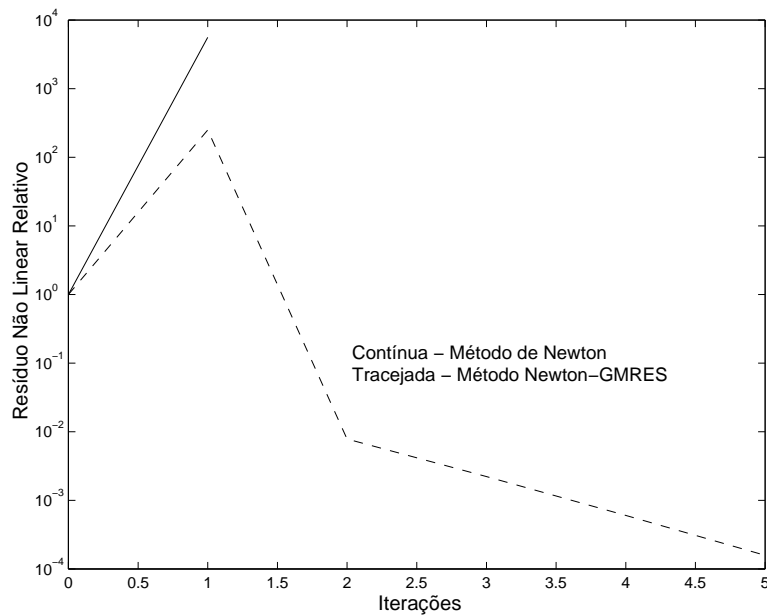


Figura 13: Métodos de Newton e Newton-GMRES para a função de Brown

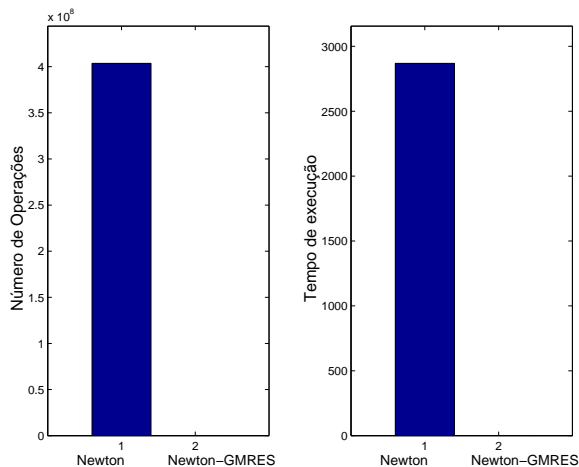


Figura 14: Métodos de Newton e Newton-GMRES para a função de Brown

Os bons resultados apresentados mostram a importância da alternativa de utilização de métodos iterativos na resolução do sistema linear necessária no método de Newton. E, entre estes métodos, o GMRES constitui-se numa boa escolha, apresentando, na grande maioria dos testes, convergência rápida e resultados bastante precisos.

Referências

- [1] R. BARRET, M. BERRY, T. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. EIJKHOUT, R. POZO, C. ROMINE, H. VAN DER VORST, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.
- [2] C. BROYDEN, A class of methods for solving nonlinear simultaneous equations, *Math. Comput.*, 19 (1965), pag. 577-593.
- [3] S. CHANDRASEKHAR, *Radiative Transfer*, Dover, New York, 1960.
- [4] C. CUNHA, *Métodos Numéricos para as Engenharias e Ciências Aplicadas*. Editora da Unicamp, Campinas, 1993.
- [5] B. N. DATTA, *Numerical Linear Algebra and Applications*. Brooks/Cole Publishing Company, California, 1995.
- [6] R. DEMBO, S. EISENSTAT & T. STEIHAUG, Inexact Newton methods, *SIAM J. Numer. Anal.*, 19 (1982), pag. 400-408.

- [7] J. DENNIS & R. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.
- [8] S. EISENSTAT & H. WALKER, Choosing the forcing terms in an inexact Newton method, *SIAM J. Sci. Comput.*, 17 (1996), pag. 16-32.
- [9] A. ESSAI, Weighted FOM and GMRES for solving nonsymmetric linear systems, *Numerical Algorithms*, 18 (1998), pag. 277-292.
- [10] HESTENES & STIEFEL, Methodos of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Stand.*, 49 (1952), pag. 409-436.
- [11] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1986.
- [12] C. T. KELLEY, Solution of the Chandrasekhar H-equation by Newton's methods, *J. Math. Phys.*, 21 (1980), pp. 1625-1628.
- [13] J. MORÉ & M. COSNORD, Numerical Solution of Nonlinear Equations, *ACM Transactions on Mathematical Software*, 5,1 (1975), pag. 64-85.
- [14] J. MORÉ, B. GARBOW & K. HILLSTROM, Testing Unconstrained Optimization Software, *ACM Transactions on Mathematical Software*, 5,1 (1981), pag. 17-41.
- [15] N. NACHTIGAL, S. REDDY, L. TREFETHEN, How fast are nonsymmetric matrix iterations?, *SIAM J. Matrix Anal. Appl.*, 13 (1992), pag. 778-795.
- [16] B. NOBLE, *Applied Linear Algebra*. Prentice Hall, Englewood Cliffs, N.J., 1969.
- [17] J. ORTEGA & W. RHEINOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.
- [18] M. RUGGIERO & V. LOPES, *Cálculo Numéricos - Aspéctos Teóricos e Computacionais*. MAKRON books do Brasil Editora Ltda, São Paulo, 1996.
- [19] SAAD & SCHULTZ, GMRES: a Generalized Minimum RESidual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp.*, 7 (1986), pag. 856-869.
- [20] H. F. WALKER, Implementation of the GMRES method using Householder transformations, *SIAM J. Sci. Stat. Comp.*, 9 (1988), pag. 152-163.
- [21] D. S. WATKINS, *Fundamentals of Matrix Computations*. John Wiley & Sons. Inc., 1991.