Universidade Estadual de Campinas

Instituto de Matemática, Estatística e Computação Científica

Método Newton-Inexato e Problemas de Valor de Contorno

João Luiz Santos Gomes

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação da Prof. Marcia A. Gomes Ruggiero.

27 de Novembro de 2019.

Resumo

O objetivo primordial deste projeto é estudar e entender a teoria do método Newton-Inexato e realizar alguns testes frente à alguns problemas valor de contorno. Assim como o método Newton-Exato, esse também possui uma zona de aproximação local tal que a taxa de convergência depende dos parâmetros do método os quais foram estudados. Utilizamos algumas técnicas de otimização para substituir os sistemas não-lineares como um problema de minimização. Para encontrar as direções de descida, estudamos o método GMRES que faz parte da família de métodos baseados em subespaços de Krylov, e assim determinamos esta através de um sistema linear. O método GMRES será fundamental como método iterativo quando a matriz do sistema apresentar estrutura esparsa.

Abstract

The primary objective of this project is to study and understand the theory of the Newton-Inexact method and perform some tests on boundary value problems. Like the Newton-Exact method, it also has a local approximation zone such that the convergence rate depends on the method's parameters that were studied. We use some optimization techniques to replace nonlinear systems as an minimization problem. To find the directions of descent, we study the GMRES method which is part of the Krylov subspace family of methods, and thus we determine it by solving a linear system. The GMRES method will be fundamental as an iterative method when the system matrix has sparse structure.

Conteúdo

1 Método GMRES

Ao deparar-se com um sistema linear onde a matriz do sistema é esparsa, é necessário estudar métodos iterativos que resolvam com eficiência estes problemas. Assim, neste projeto, iremos estudar o método GMRES proposto por **Youcef Saad** e **Martin Schultz** [7] o qual faz parte da família de métodos baseados em projeções em subespaços de Krylov. Feita toda a introdução teórica, alguns testes serão realizados para avaliar o desempenho dos algoritmos utilizando um conjunto de sistemas aleatórios e a matriz de Hilbert.

1.1 Iteração de Arnoldi

Em primeiro lugar, em todo o texto iremos considerar a matriz quadrada não-singular $A \in \mathbb{R}^{n\times n}$ e o vetor $b \in \mathbb{R}^n$. A partir disso, definimos o subespaço de **Krylov** associado à A e b como sendo:

$$\mathcal{K}_s(A, b) = span\{\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \ldots, A^{s-1}\boldsymbol{b}\}, \tag{1}$$

onde s é a dimensão do subespaço gerado.

A **Iteração de Arnoldi** é um método que busca, de maneira semelhante ao método Gram-Schmidt, computar uma base ortonormal para o subespaço de Krylov. Para deduzi-lo, consideramos a transformação de similaridade $A = QHQ^T \Leftrightarrow AQ = QH$. Dessa maneira, em alguma iteração $s < n$ do método teremos:

$$AQ_s = Q_{s+1}\bar{H}_s. \tag{2}$$

Note que $Q_s \in \mathbb{R}^{n\times s}$, $Q_{s+1} \in \mathbb{R}^{n\times s+1}$ e $\bar{H}_s \in \mathbb{R}^{s+1\times s}$ é matriz de Hessemberg superior. A partir da relação acima, note que

$$A\boldsymbol{q}_s = \sum_{i=1}^{s+1} h_{is}\boldsymbol{q}_i \quad \Leftrightarrow \quad \boldsymbol{q}_{s+1} = \frac{A\boldsymbol{q}_s - \sum_{i=1}^{s} h_{is}\boldsymbol{q}_i}{h_{n+1,n}}.$$

Aqui, o vetor $\boldsymbol{q}_j$ é a j-ésima coluna de Q e h_{ij} é um elemento de $\bar{H}$. A partir da relação de recorrência encontrada acima, descrevemos o algoritmo no pseudo-código a seguir:

Algoritmo 1: Iteração de Arnoldi

Entrada: Vetor b arbitrário

início
  $q_1 = b/||b||_2$
  para *s = 1,2,...* **faça**
    $\boldsymbol{v} = A\boldsymbol{q}_s$
    para *i = 1,...,s* **faça**
      $h_{is} = \boldsymbol{q}_i^T\boldsymbol{v}$
      $\boldsymbol{v} = \boldsymbol{v} - h_{is}\boldsymbol{q}_i$
    fim
    $h_{s+1,\,s} = ||\boldsymbol{v}||_2$
    $\boldsymbol{q}_{s+1} = \boldsymbol{v}/h_{s+1,\,s}$
  fim
fim

O método de Arnoldi será de extrema importância para o desenvolvimento do método seguinte.

1.2 O Método GMRES

Como dito anteriormente, o algoritmo GMRES: ***Generalized Minimal RESidual*** é um método iterativo que pode ser utilizado para determinar a solução de sistemas lineares onde a matriz do sistema deve ser apenas quadrada e não singular.

Dado o sistema linear $A\boldsymbol{x} = \boldsymbol{b}$, a ideia central do método é a seguinte: na iteração s aproximamos a solução exata $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ por um vetor $\boldsymbol{x}_s \in \mathcal{K}_s$ minimizando a norma-2 do resíduo dado por

$$||\boldsymbol{r}_s||_2 = ||A\boldsymbol{x}_s - \boldsymbol{b}||_2.$$

Vamos denotar $K_s \in \mathbb{R}^{n\times s}$ a matriz cujas colunas são dadas por $\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{s-1}\boldsymbol{b}\}$. A partir disso, seja $\boldsymbol{\gamma} \in \mathbb{R}^s$, então

$$\boldsymbol{x}_s \in \mathcal{K}_s \quad \Leftrightarrow \quad \boldsymbol{x}_s = K_s\boldsymbol{\gamma}.$$

Portanto, conseguimos obter o seguinte problema de quadrados mínimos

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}^s} ||(AK_s)\boldsymbol{\gamma} - \boldsymbol{b}||_2,$$

cuja solução pode ser obtida através da fatoração QR da matriz AK_s. Entretanto, pela dimensão da matriz A, este processo pode ser muito custoso e instável. Com isso, podemos utilizar de uma base ortonormal para o subespaço de Krylov, ou seja, a cada iteração utilizamos o método de Arnoldi para obtermos

$$\boldsymbol{q}_{s+1} \perp \boldsymbol{q}_i \qquad \text{para } i = 1, \ldots, s$$

tal que $span\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{s-1}\boldsymbol{b}, A(A^{s-1}\boldsymbol{b})\} = span\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_s, A\boldsymbol{q}_s\}$. Consequentemente, para $\boldsymbol{\beta} \in \mathbb{R}^s$ apropriado, a solução aproximada pode ser reescrita por:

$$\boldsymbol{x}_s = Q_s\boldsymbol{\beta}.$$

Utilizando a transformação de similaridade (2), o problema é reformulado como:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^s} ||(AQ_s)\boldsymbol{\beta} - \boldsymbol{b}||_2 \quad \Leftrightarrow \quad \min_{\boldsymbol{\beta}\in\mathbb{R}^s} ||(Q_{s+1}\bar{H}_s)\boldsymbol{\beta} - \boldsymbol{b}||_2.$$

Em seguida, tomamos vantagem do fato de que a norma-2 é invariante mediante aplicação de uma matriz unitária e obtemos

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^s} ||(Q_{s+1}^T Q_{s+1}\bar{H}_s)\boldsymbol{\beta} - Q_{s+1}^T\boldsymbol{b}||_2 \quad \Leftrightarrow \quad \min_{\boldsymbol{\beta}\in\mathbb{R}^s} ||\bar{H}_s\boldsymbol{\beta} - Q_{s+1}^T\boldsymbol{b}||_2.$$

Note que $Q_{s+1}^T\boldsymbol{b} = \boldsymbol{q}_i^T\boldsymbol{b} = 0$, para $i > 1$. Como $\boldsymbol{q}_1 = b/||b||_2$, então $Q_{s+1}^T\boldsymbol{b} = ||b||_2\boldsymbol{e}_1 = \boldsymbol{\rho}$ e, assim, chegamos à simplificação final do problema dada por:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^s} ||\bar{H}_s\boldsymbol{\beta} - \boldsymbol{\rho}||_2 \ , \tag{3}$$

Algoritmo 2: GMRES

Entrada: Vetor b arbitrário, *tol*

início
  $q_1 = b/||b||_2, \quad \boldsymbol{\rho} = ||b||_2 \boldsymbol{e}_1$
  para *s= 1,2,3...* **faça**
    Executar a s-ésima iteração do método de Arnoldi.
    Encontrar $\boldsymbol{\beta}$ que minimiza $||\bar{H}_s\boldsymbol{\beta} - \boldsymbol{\rho}||_2$. *(Utilizando fatoração QR)*
    se $||\bar{H}_s\boldsymbol{\beta} - \boldsymbol{\rho}||_2 < tol$ **então**
      $k = s$
      Break!
    fim
  fim
  Faça $\boldsymbol{x}_k = Q_k\boldsymbol{\beta}$.
fim

o qual pode ser resolvido utilizando a fatoração QR da matriz $\bar{H}_s$ por **Rotações de Givens** (veja [7]). Finalmente, podemos escrever o pseudo-código para o GMRES:

Note que a cada passo do algoritmo a base para o espaço de Krylov é armazenada. Ou seja, ao realizar k iterações do GMRES, é necessário guardar k vetores de tamanho n. Para problemas de grande porte isso pode se tornar uma dificuldade na medida que a máquina não irá dispor de memória suficiente para prosseguir com o algoritmo. A fim de contornar este contratempo, fazemos com que GMRES seja executado m vezes ($m < n$) e assim minimizamos a norma-2 do resíduo dado por $||\boldsymbol{r}_m|| = ||\boldsymbol{b} - A\boldsymbol{x}_m||$. Caso $||\boldsymbol{r}_m|| > tol$, então executamos novamente o GMRES fazendo o vetor inicial igual a $\boldsymbol{x}_m$. O algoritmo com esta modificação será denominado **GMRES(m)**.

Em seguida demonstramos uma propriedade que diz respeito à convergência dos métodos GMRES e GMRES(m)[3].

Proposição 1.2.1. *Sejam A uma matriz não-singular e o vetor $\boldsymbol{q}_{s+1}$ gerado pela iteração de Arnoldi. Se $h_{s+1,\,s} = 0$, então $\boldsymbol{x} = A^{-1}\boldsymbol{b} \in \mathcal{K}_s$*

Demonstração. Se $h_{s+1,\,s} = 0$, pela construção do método de Arnoldi, temos que

$$||\boldsymbol{v}||_2 = 0 \quad \Leftrightarrow \quad \boldsymbol{q}_{s+1} = A\boldsymbol{q}_s = h_{is}\boldsymbol{q}_i$$

Ou seja, o vetor gerado pelo processo de Arnoldi é linearmente dependente aos outros vetores colunas da matriz Q_s e, portanto, $dim(\mathcal{K}_s) = n$. Dessa maneira, como $\boldsymbol{x}_s \in \mathbb{R}^n$ e por (3), a solução será dada por:

$$\boldsymbol{\beta} = ||\boldsymbol{b}||_2 H^{-1}\boldsymbol{e}_1 \quad \Rightarrow \quad \boldsymbol{x} = Q_s\boldsymbol{\beta}$$

□

Podemos concluir que, se $A \in R^{n\times n}$ é não-singular, o método GMRES irá terminar em n passos e, portanto, o método GMRES(m) nunca irá colapsar. No entanto, o método irá gerar uma sequência dada por $\{x_n\}_{n\in\mathbb{N}}$, onde $x_n = ||\boldsymbol{r}_n||_2$, tal que essa sequência seja não-crescente. Consequentemente, podemos ter casos em que o método irá estagnar. (para uma análise mais profunda sobre a convergência do GMRES(m) veja [7]).

1.3 Testes Computacionais com GMRES

A seguir, iremos realizar alguns testes computacionais utilizando o método GMRES. Para isso, contamos com a função $gmres(A, b)$ disponibilizada no MATLAB. A matriz A possui valores aleatórios gerados por uma distribuição uniforme no intervalo $[30, -30]$ e o vetor b idem.

Primeiramente, realizamos testes com matrizes e vetores aleatórios de diferentes dimensões e, em seguida, verificamos a minimização da norma-2 do resíduo com o decorrer das iterações. Finalmente, fizemos testes com a matriz de Hilbert para estudar o comportamento do método frente à matrizes mal-condicionadas.

É importante ressaltar que os testes foram feitos com tolerância 0.1, pois em seguida, iremos estudar o Método de Newton-Inexato, o qual utilizará o GMRES com tolerâncias não muito baixas. Os resultados estão dispostos nas tabelas a seguir. A coluna **cond** representa o número de condição da matriz gerada e a coluna **error** representa a norma-2 da diferença dos vetores solução original e da solução aproximada com GMRES.

dim	**flag**	**cond**	**error**	**iter**
20	0	269.543	8.869	3
50	0	4577.886	17.597	27
100	0	18181.708	25.715	48
300	0	22102.192	45.387	203

Tabela 1: Resultados do Método GMRES aplicado em sistemas aleatórios

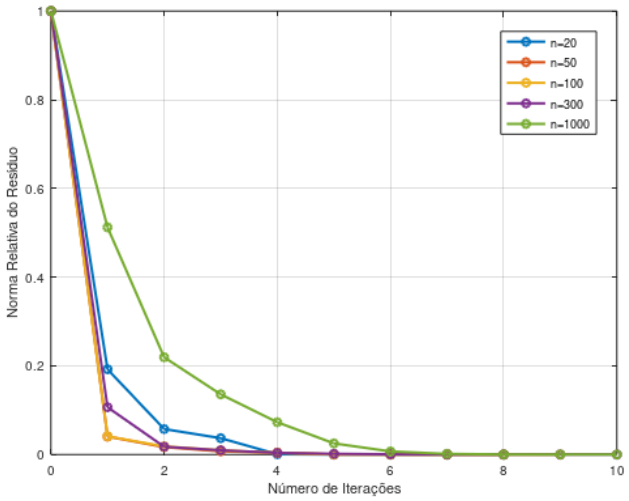

Figura 1: Testes do método GMRES com matrizes de Hilbert de dimensão n

dim	**flag**	**cond**	**error**	**iter**
20	0	1.355e+18	8.273	8
50	0	2.312e+19	18.114	8
100	0	1.991e+19	25.652	9
300	0	9.030e+19	49.054	10
1000	0	9.657e+20	90.385	13

Tabela 2: Diferentes resultados do GMRES com a Matriz de Hilbert

2 O Método Newton-Inexato

Nesta seção será apresentado uma introdução ao método de **Newton-Inexato** para a resolução de sistemas não-lineares. Também serão enunciados alguns teoremas que dizem respeito à convergência do método. O método GMRES terá enorme importância para o desenvolvimento do algoritmo que será detalhado mais a frente.

Seja $D \subset R^n$ aberto e convexo e $F : D \to \mathbb{R}^n$. Estamos interessados em resolver

$$\begin{cases} F(\boldsymbol{x}) = 0 \\ \boldsymbol{x} \in \mathbb{R}^n \end{cases} \tag{4}$$

Supomos que exista $\bar{\boldsymbol{x}} \in D$ tal que $F(\bar{\boldsymbol{x}}) = 0$ com matriz Jacobiana $J_F(x)$ não-singular e Lipschitz contínua, ou seja, existe $\delta > 0$ tal que

$$||J_F(\boldsymbol{x}) - J_F(\bar{\boldsymbol{x}})|| \leq c\,||\boldsymbol{x} - \bar{\boldsymbol{x}}||,$$

para $\boldsymbol{x}$ pertencente à bola aberta $B_\delta(\bar{\boldsymbol{x}})$ e $c > 0$.

Primeiramente, vamos relembrar que o **Método de Newton-Exato**, um dos métodos mais empregados na resolução de sistemas não-lineares, gera uma sequência $\{\boldsymbol{x}_k\}$ tal que a partir de $\boldsymbol{x}_k$, obtemos o passo $\boldsymbol{s}_k$ por

$$J_F(\boldsymbol{x_k})\boldsymbol{s}_k = -F(\boldsymbol{x}_k) \tag{5}$$

e então

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k \tag{6}$$

Uma das propriedades mais importantes do método é sua taxa de convergência quadrática (veja [2]). Ou seja, se $J(\boldsymbol{x})$ é Lipschitz contínua em uma vizinhança de $\bar{\boldsymbol{x}}$ e $||J^{-1}|| < \beta$ então existe $\delta > 0$ tal que a partir de $\boldsymbol{x_0} \in B_\delta(\bar{\boldsymbol{x}})$ a sequência gerada converge para $\bar{\boldsymbol{x}}$ satisfazendo

$$||\boldsymbol{x_{k+1}} - \bar{\boldsymbol{x}}|| \leq c\,||\boldsymbol{x}_k - \bar{\boldsymbol{x}}||^2$$

Dessa maneira, podemos nos perguntar o seguinte: sabendo que o método de Newton-Exato possui essa ótima propriedade, por que estudar outro método para resolver (4)? Basta notar que os cálculos das derivadas da matriz Jacobiana e a resolução exata de (5) podem ser muito caros computacionalmente em se tratando de problemas de grande porte. Por exemplo, se realizássemos a fatoração LU, então aproximadamente $\frac{2}{3}n^3$ operações seriam

feitas. Além disso, em sistemas de grande porte, usualmente as matrizes Jacobianas são esparsas, e a resolução através de métodos diretos provocam preenchimentos na estrutura da matriz.

Para contornar as dificuldades com a resolução exata do sistema linear, introduzimos o método de Newton-Inexato.

2.1 Newton-Inexato e Newton-GMRES

O método Newton-Inexato consiste em utilizarmos de métodos iterativos para encontrar uma solução aproximada do sistema linear (5). Ou seja, encontramos o passo $\boldsymbol{s}_k$ que satisfaz a seguinte desigualdade:

$$||\boldsymbol{r}_k|| = ||J_F(\boldsymbol{x_k})\boldsymbol{s}_k + F(\boldsymbol{x}_k)|| \leq \eta_k \, ||F(\boldsymbol{x}_k)|| \tag{7}$$

onde $\eta_k \in [0, 1)$ é denominado **termo forçante**. A sequência $\{\eta_k\}$ tem importância fundamental para controlar a precisão e convergência do passo. Podemos observar que por (7), o resíduo relativo a cada iteração do método deverá ser menor ou igual ao termo forçante. Repare que para $\eta_k \equiv 0$, estaremos no método de Newton-Exato.

A seguir, enunciamos um teorema que assegura a convergência local para o método. A demonstração se encontra em [2].

Teorema 2.1.1. *Seja $F \in \mathcal{C}^1$ e $F : D \to \mathbb{R}^n$. Considere $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ tal que $F(\bar{\boldsymbol{x}}) = 0$, $J_F(\bar{\boldsymbol{x}})$ não-singular e a sequência $\{\eta_k\}$ com $\eta_k < 1$ para todo k. Dessa maneira, existe $\delta > 0$ tal que, se $\boldsymbol{x}_0 \in B_\delta(\bar{\boldsymbol{x}})$, a sequência $\{\boldsymbol{x}_k\}$ gerada pelo método Newton-Inexato, converge para $\bar{\boldsymbol{x}}$ com taxa de convergência dada por:*

(i) Se $\eta_k = cte.$ para todo k, temos convergência linear.

(ii) Se $\lim_{k\to\infty} \eta_k = 0$, temos convergência super-linear.

(iii) Se $\eta_k = \mathcal{O}(\, ||F(\boldsymbol{x}_k))||\,)$ e a matriz Jacobiana for Lipschitz em $\bar{\boldsymbol{x}}$, temos convergência quadrática.

Portanto, pelo teorema anterior, diferentes escolhas para o termo forçante levam à diferentes performances para o Newton-Inexato.

Neste projeto o passo $\boldsymbol{s}_k$ será calculado utilizando o método GMRES para sistemas lineares. Desse modo, estaremos utilizando o método denominado **Newton-GMRES**. Vale ressaltar que neste método, a cada iteração de Newton (iteração externa) teremos um certo número de iterações realizadas do GMRES (iterações internas) as quais estão determinadas por η_k. Pode acontecer de um número muito grande de iterações internas serem realizadas a cada iteração externa e a norma de F ter um decréscimo insignificante. Sendo assim, listamos algumas escolhas para o termo forçante podem ser utilizadas.

(i) (E1) Escolha 1: $\eta_k = \eta_0 = 0.01$;

(ii) (E2) Escolha 2: $\eta_k = \frac{1}{2^{k+1}}$;

(iii) (E3) Escolha 3: $\eta_k = \gamma \left(\frac{||F(\boldsymbol{x}_k)||}{||F(\boldsymbol{x}_{k-1})||} \right)^{\alpha}$, onde $\gamma \in$ [0,1] e $\alpha \in$ (1,2].

É importante ressaltar que pode-se associar o sistema (4) com um problema de minimização irrestrita dado por:

$$\min f(\boldsymbol{x}) = \frac{1}{2}||F(\boldsymbol{x})||_2^2\,, \quad \boldsymbol{x} \in \mathbb{R}^n$$

Sendo assim, podemos aplicar técnicas de otimização não-linear, para encontrar o mínimo da função $f(\boldsymbol{x})$. Uma delas é a chamada busca linear. Após encontrarmos uma direção de descida $\boldsymbol{s}_k$, é preciso saber quanto é necessário percorrer esta direção para que a função decresça. Aqui utilizaremos uma condição parecida com a exibida em [2] dada por:

$$||F(\boldsymbol{x}_k + t\boldsymbol{s}_k)||_2 < (1 - t\sigma)||F(\boldsymbol{x}_k)||_2\,, \tag{8}$$

onde $\sigma \in (0, 1)$ e $t \leq 1$.

A seguir vamos mostrar que a direção $\boldsymbol{s}_k$ que satisfaz a relação (7) é de descida, isto é, $\nabla f(\boldsymbol{x}_k)^T s < 0$.
Sabemos que $\nabla f(\boldsymbol{x}_k) = J_F(\boldsymbol{x}_k)^T F(\boldsymbol{x}_k)$. Dessa maneira,

$$\nabla f(\boldsymbol{x}_k)^T \boldsymbol{s}_k = J_F(\boldsymbol{x}_k)^T F(\boldsymbol{x}_k)\boldsymbol{s}_k = F(\boldsymbol{x}_k)^T \boldsymbol{r}_k - F(\boldsymbol{x}_k)^T F(\boldsymbol{x}_k)$$

onde $\boldsymbol{r}_k = J_F(\boldsymbol{x}_k)\boldsymbol{s}_k + F(\boldsymbol{x}_k)$. Agora, observe que

$$F(\boldsymbol{x}_k)^T \boldsymbol{r}_k \leq |F(\boldsymbol{x}_k)^T \boldsymbol{r}_k| \leq ||F(\boldsymbol{x}_k)||_2\, ||\boldsymbol{r}_k||_2 < ||F(\boldsymbol{x}_k)||_2^2 = F(\boldsymbol{x}_k)^T F(\boldsymbol{x}_k).$$

Nas desigualdades acima utilizamos a relação (7) e a desigualdade de Cauchy-Schwarz (veja [4]). Ao final, teremos:

$$F(\boldsymbol{x}_k)^T \boldsymbol{r}_k - F(\boldsymbol{x}_k)^T F(\boldsymbol{x}_k) < 0 \;\Rightarrow\; \nabla f(\boldsymbol{x}_k)^T \boldsymbol{s}_k < 0$$

Finalmente, descrevemos o algoritmo para o método de Newton Inexato abaixo.

Algoritmo 3: NEWTON INEXATO

Entrada: $\boldsymbol{x}_0$, tol, $\sigma \in (0, 1)$

início
  k = 0
  enquanto $||F(\boldsymbol{x}_k)||_2 > tol$ **faça**
    Escolha η_k.
    Calcule a direção $\boldsymbol{s}_k$ que satisfaz $||J_F(\boldsymbol{x}_k)\boldsymbol{s}_k + F(\boldsymbol{x}_k)||_2 \leq \eta_k\, ||F(\boldsymbol{x}_k)||_2$.
    $t = 1$
    enquanto $||F(\boldsymbol{x}_k + t\boldsymbol{s}_k)||_2 < (1 - t\sigma)||F(\boldsymbol{x}_k)||_2$ **faça**
      $t = t/2$
    fim
    Faça $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t\boldsymbol{s}_k$
    $k = k + 1$
  fim
fim

A seguir iremos resolver dois exemplos utilizando o método anterior para realizar alguns testes com as diferentes escolhas para o termo forçante. Para os parâmetros, escolhemos

$\sigma = 10^{-4}$ e $tol = 10^{-4}$. Além disso utilizamos o método GMRES sem restart para encontrarmos as direções de descida.

Exemplo 1 - Broyden Tridiagonal
Considere o seguinte sistema de equações:

$$\begin{cases} f_1(x) = (3-2x_1)x_1 - 2x_2 + 1 = 0 \\ f_i(x) = (3-2x_i)x_i - x_{i-1} - 2x_{i+1} + 1 = 0\,, \;\; i = 2, \ldots, n-1 \\ f_n(x) = (3-2x_n)x_n - x_n + 1 = 0 \end{cases}$$

onde $\boldsymbol{x}^0 = (0, 0, \ldots, 0)^T$. Dessa maneira, a matriz Jacobiana será dada por:

$$J_F(\boldsymbol{x}) = \begin{pmatrix} 3-4x_1 & -2 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 3-4x_2 & -2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 3-4x_3 & -2 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 3-4x_{n-1} & -2 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 3-4x_n \end{pmatrix}$$

Aplicando o método Newton-GMRES com $n = 10$, atingimos uma solução com os seguintes resultados distribuídos na tabela a seguir:

η_k	**iter_ex**	**iter_in**	**tempo(s)**
E1	8	35	0.0189
E2	8	25	0.0151
E3	6	21	0.0147

Tabela 3: Resultados dos testes para os termos forçantes na resolução do sistema não-linear.

Exemplo 2 - PVC Não-Linear
Considere o seguinte problema de valor de contorno não linear:

$$\begin{cases} y'' = ysen(y) + xy \\ y(0) = 1 \\ y(1) = 5 \end{cases}$$

Fixado n inteiro, consideramos espaçamentos $h = 1/n$ e dividiremos o intervalo $[0, 1]$ em $x_0 = 0, x_i = ih$ e $x_n = 1$ para $i = 1, 2, \ldots, n-1$. Assim, obtemos as incógnitas $y(x_1) = y_1$, $y(x_2) = y_2$, $\ldots$, $y(x_{n-1}) = y_{n-1}$ e poderemos utilizar a aproximação por diferenças centrais para a segunda derivada dada por

$$y''(x_i) = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}, \quad i = 1, 2, \ldots, n-1.$$

Desse modo, substituindo a aproximação na equação diferencial e organizando os termos, obtemos o seguinte sistema de equações:

$$\begin{cases} 1 - y_1(h^2 sen\, y_1 + h + 2) + y_2 = 0 \\ y_{i-1} - y_i(h^2 sen\, y_i + ih + 2) + y_{i+1} = 0, \quad i = 2, \ldots, n-2 \\ y_{n-2} - y_{n-1}(h^2 sen\, y_{n-1} + (n-1)h + 2) + 5 = 0 \end{cases} \tag{9}$$

Neste caso, a matriz Jacobiana será dada por:

$$J_{ij}(y) = \begin{cases} J_{11} = -h^2\left(\sin\left(y_1\right) + h\right) - h^2 y_1 \cos\left(y_1\right) - 2 \\ J_{12} = 1 \\ J_{ii} = -h^2\left(\sin\left(y_i\right) + ih\right) - h^2 y_i \cos\left(y_i\right) - 2, \quad i = 2, \ldots, n-2 \\ J_{i,i-1} = J_{i,i+1} = 1, \quad i = 2, \ldots, n-2 \\ J_{n-2,n-1} = 1 \\ J_{n-1,n-1} = -h^2\left(\sin\left(y_{n-1}\right) + (n-1)h\right) - h^2 y_1 \cos\left(y_{n-1}\right) - 2 \\ \text{Os outros elementos serão nulos} \end{cases}$$

Utilizando o ponto inicial como $\boldsymbol{y}^0 = (0, 0, \ldots, 0)^T$ e definindo espaçamentos de tamanho $h = 1/64$, iremos resolver o sistema não-linear para $63 \times 63 = 3969$ equações e variáveis. Sendo assim conseguimos obter os seguintes resultados:

η_k	**iter_ex**	**iter_in**	**tempo(s)**
E1	60	565	0.0584
E2	32	268	0.0459
E3	24	126	0.0223

Tabela 4: Resultados dos testes para os termos forçantes na resolução do PVC não-linear.

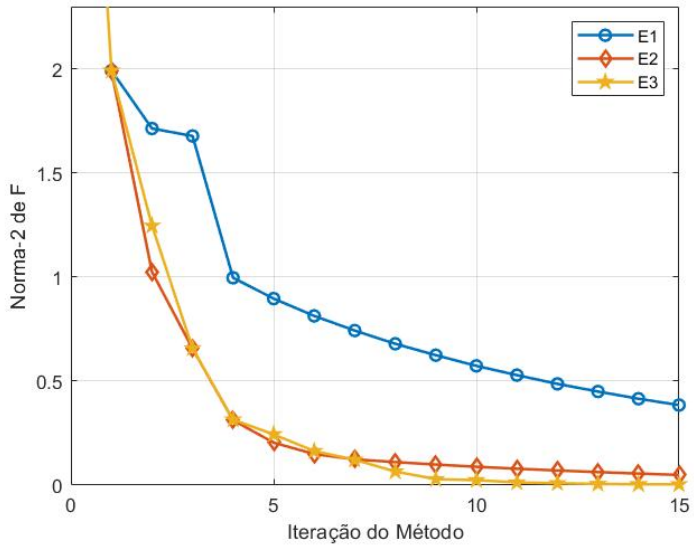

Figura 2: Testes para os termos forçantes na resolução do PVC não-linear.

Observe que, no exemplo 2, as três escolhas diminuíram de forma significativa a norma-2 de F nas primeiras iterações. Logo em seguida, com a escolha 1, o decréscimo continuou de forma constante e gradual. Na escolha 2, a função está decrescendo de forma muito lenta. Para escolha 3 o método convergiu rápido com poucas iterações internas. Podemos concluir

o seguinte: após um certo tempo, o método passou a encontrar direções de descida sobre as quais a redução da norma de F era pouco significante, ou seja, a direção encontrada estava próxima de ser perpendicular à $\nabla f(\boldsymbol{x}_k)$. Para evitar este problema, poderíamos modificar o algoritmo para que o termo forçante se comportasse de forma mais dinâmica tanto nas primeiras iterações como nas últimas, como feito em [2].

3 Problemas de Valor de Contorno

Nesta seção iremos estudar problemas de valor de contorno (PVC) e encontrar soluções aproximadas para estes através dos métodos desenvolvidos anteriormente. Os principais problemas estudados o **Problema de Bratu** e o **Problema da Convecção-Difusão**. Ambos os problemas são caracterizados por encontrar uma solução $u : \Omega = [0,1] \times [0,1] \to \mathbb{R}$ e $u \in \mathcal{C}^2$ tal que

$$\begin{cases} -\nabla^2 u + h(\lambda, u) = f(s,t) \\ u(s,t) = 0 \ \text{ em } \partial\Omega \end{cases} \tag{10}$$

onde ∇^2 é o operador Laplaciano em coordenadas cartesianas. As funções h e f são reais e juntamente com o parâmetro $\lambda \in \mathbb{R}$, definem os diversos PVC. Neste trabalho, as derivadas serão aproximadas por diferenças centrais e a malha resultante da discretização irá dispor de 63 pontos internos em cada eixo. Portanto, o sistema não-linear resultante tem $63 \times 63 = 3969$ variáveis e equações.

3.1 Problema de Bratu

Este problema é caracterizado por (10), onde h é dada por $h(\lambda, u) = -\lambda\, exp(u)$ e $f(s,t)$ é construída de forma que já conhecemos a solução exata. Consideramos duas soluções, dadas por

$$u^*(s,t) = 10st(1-s)(1-t)\, \exp(s^{4.5})$$
$$u^{**}(s,t) = (2s - s^3)(sin(3\pi\, t))$$

como mostrado em [2]. Dessa maneira, a formulação do PVC será

$$\begin{cases} -\nabla^2 u + \lambda e^u = f(s,t) \ \text{ em } \Omega \\ u(s,t) = 0 \ \text{ em } \partial\Omega \end{cases} \tag{11}$$

Para resolver o problema determinamos alguns valores positivos e negativos para o parâmetro λ. O chute inicial escolhido é $\boldsymbol{x}_0 = (0, 0, \ldots, 0)^T$. A partir de sua discretização, o problema é reescrito da seguinte maneira:

$$-\frac{1}{h^2}(-4u_{i,j} + u_{i+1,j} + u_{u-1,j} + u_{i,j+1} + u_{i,j-1}) + \lambda \exp u_{i,j} = f, \quad 1 \le i, j \le 63$$

É importante notar que, como estamos resolvendo o sistema não-linear resultante com muitas variáveis, a matriz Jacobiana associada será uma matriz de grande porte. Dessa maneira, o

método GMRES(m) terá papel fundamental, como método iterativo, quando exploramos a esparsidade da matriz mostrada na figura a seguir:

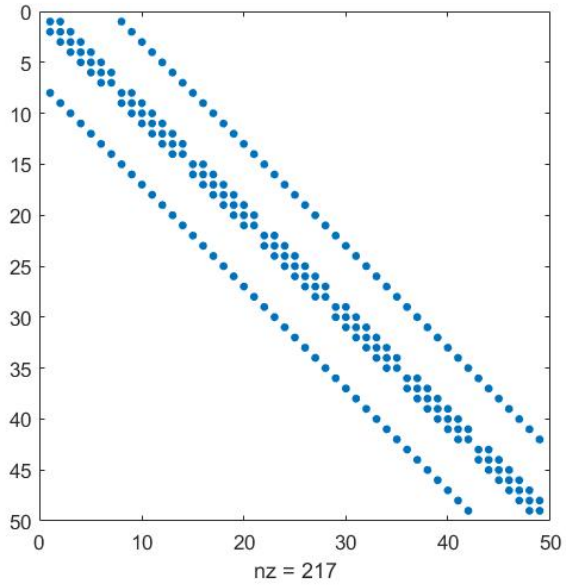

Figura 3: Exibição da esparsidade da matriz Jacobiana associada a problema de Bratu discretizado ($L = 7$) com comando *spy* do MATLAB.

Dessa maneira consideramos parâmetros como $\lambda = [-1000, -500, -100, -10, 1, 5, 10]$, $\sigma = 10^{-4}$, $tol = 10^{-4}$, $m = 30$, $\gamma = 1$ e $\alpha = \frac{1+\sqrt{5}}{2}$ e os resultados obtidos foram os seguintes:

	λ	**-1000**	**-500**	**-100**	**-10**	**1**	**5**	**10**
	iter_ex	6	6	5	5	4	5	8
E1	**iter_in**	58	83	70	240	230	510	2142
	tempo (s)	1.63	1.79	2.37	4.64	6.27	9.16	21.52
	iter_ex	7	8	8	7	7	7	9
E2	**iter_in**	53	80	119	209	312	430	1714
	tempo (s)	1.75	2.08	2.91	4.16	6.02	7.54	19.91
	iter_ex	5	6	6	4	3	4	9
E3	**iter_in**	53	78	174	246	444	477	2340
	tempo (s)	1.42	2.23	3.61	5.61	6.23	8.52	24.07

Tabela 5: Resultados para o problema de Bratu para os diferentes valores de λ utilizando a solução u^*.

Analisando os dados da tabela anterior, podemos observar que para os três primeiros valores de λ a escolha E1 teve o menor tempo que as outras duas. No entanto, para os valores seguintes de λ, a escolha E2 obteve o melhor tempo de execução e o menor número de iterações internas. A escolha E3 foi a que realizou o método de procura pela solução em menos iterações externas, ou seja, mesmo demorando para calcular a direção de descida, esta

era uma ótima direção de minimização da norma-2 da função objetivo enquanto que, para a escolha E2, foi o contrário: o método GMRES(m) calculou as direções de descida de forma mais rápida. Podemos entender que para valores cada vez maiores de λ é interessante optar pelas escolhas E2 ou E3 e para valores menores podemos escolher E1.

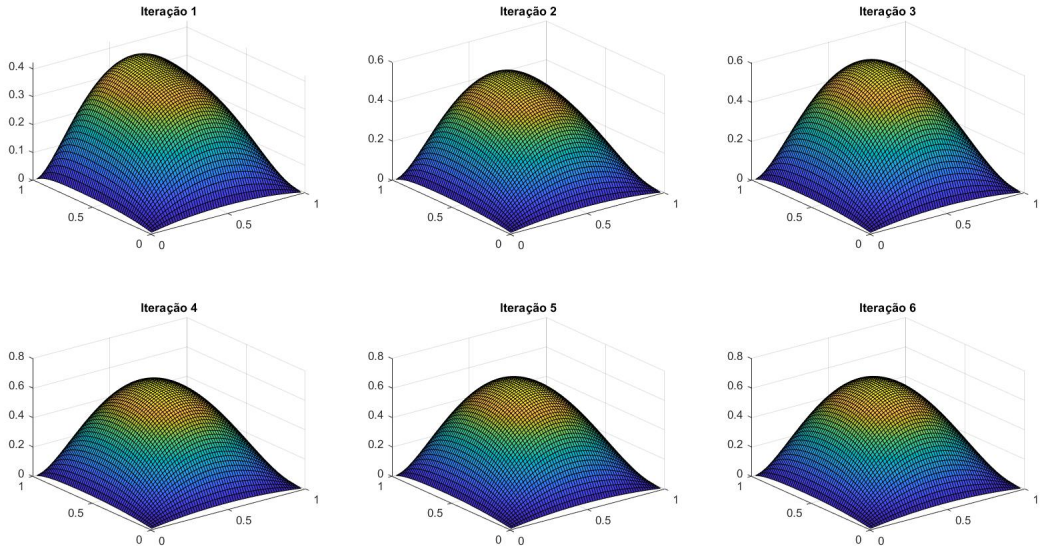

Figura 4: Comportamento das soluções aproximadas de u^* nas 6 primeiras iterações do método Newton-Inexato com $\lambda = 10$ e escolha E3.

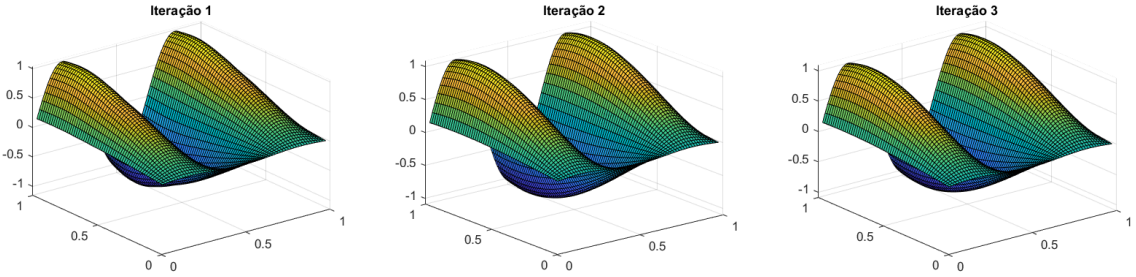

Figura 5: Comportamento das soluções aproximadas de u^{**} nas 3 primeiras iterações do método Newton-Inexato com $\lambda = 10$ e escolha E3.

3.2 Problema da Convecção-Difusão

A equação da convecção-difusão descreve fenômenos físicos onde partículas, energia, calor ou outras grandezas físicas, se comportam em meios onde há a ação dois processos físicos: convecção e difusão. Neste problema consideramos $h(\lambda, u) = \lambda\, u\left(\frac{\partial u}{\partial t} + \frac{\partial u}{\partial s}\right) = \lambda\, u(\nabla \cdot u)$.

Sendo assim, a formulação do problema será dada por

$$\begin{cases} -\nabla^2 u + \lambda\, u(\nabla \cdot u) = f(s,t) \ \text{ em } \Omega \\ u(s,t) = 0 \ \text{ em } \partial\Omega \end{cases} \tag{12}$$

Assim como no problema anterior, a função $f(s,t)$ é calculada para que as soluções exatas também sejam

$$\begin{aligned} u^*(s,t) &= 10st(1-s)(1-t)\ \exp(s^{4.5}) \\ u^{**}(s,t) &= (2s - s^3)(sin(3\pi\, t)) \\ u^{***}(s,t) &= 1000ts(1-t)(1-s)(s-0.5)(t-0.5)exp(s^{4.5}); \end{aligned}$$

Para resolver este problema determinamos alguns valores positivos de λ. O chute inicial escolhido foi $\boldsymbol{x}_0 = (0, 0, \ldots, 0)^T$. A partir da sua discretização, o problema pode ser reescrito da seguinte maneira:

$$\begin{aligned} &-\frac{1}{h^2}(-4u_{\,i,j} + u_{\,i+1,j} + u_{\,i-1,j} + u_{\,i,j+1} + u_{\,i,j-1}) \\ &+\frac{\lambda}{2h}(u_{\,i+1,j} - u_{\,i-1,j} + u_{\,i,j+1} - u_{\,i,j-1}) = f \end{aligned}$$

para $1 \leq i, j \leq 63$.

É importante ressaltar que a estrutura da matriz Jacobiana associada ao sistema não-linear é a mesma do problema anterior. Sendo assim, utilizamos os mesmos parâmetros do problema anterior, mas teremos agora $\lambda = [5, 10, 50, 75, 100, 150]$. Os resultados obtidos estão distribuídos na seguinte tabela:

	λ	**5**	**10**	**50**	**75**	**100**	**150**
	iter_ex	5	5	8	12	23	63
E1	**iter_in**	393	437	460	1937	1790	15669
	tempo (s)	5.92	6.03	8.71	23.92	24.33	155.76
	iter_ex	7	7	10	12	24	77
E2	**iter_in**	274	244	732	1264	5555	17087
	tempo (s)	4.66	4.68	12.68	23.01	39.42	180.31
	iter_ex	6	8	12	13	22	91
E3	**iter_in**	304	307	502	882	1344	2548
	tempo (s)	5.87	5.32	9.05	17.40	29.12	52.78

Tabela 6: Resultados para o problema da Convecção-Difusão para os diferentes valores de λ utilizando a solução u^*

Primeiramente, podemos notar que a escolha E1 teve a melhor performance no que diz respeito ao número de iterações externas e a escolha E3 teve ótimo desempenho quanto ao número de iterações internas e tempo. A escolha E2 desempenhou um pouco melhor que E3 nas duas primeiras escolhas de λ. Dessa maneira, podemos entender que E2 e E3 possuem termos forçantes bem flexivos inicialmente fazendo o método GMRES(m) encontrar

direções ruins e, aliado a isso, no começo do método estamos longe da zona de convergência definida no teorema anterior. Portanto, as duas escolhas possuem dificuldades em minimizar a norma-2 de F nas primeiras iterações.

Em segundo lugar, escolha E1 apresentou maior número de iterações internas e tempo para os valores menores de λ. Assim, na resolução deste problema, poderíamos adotar E3 pelo seu bom desempenho geral.

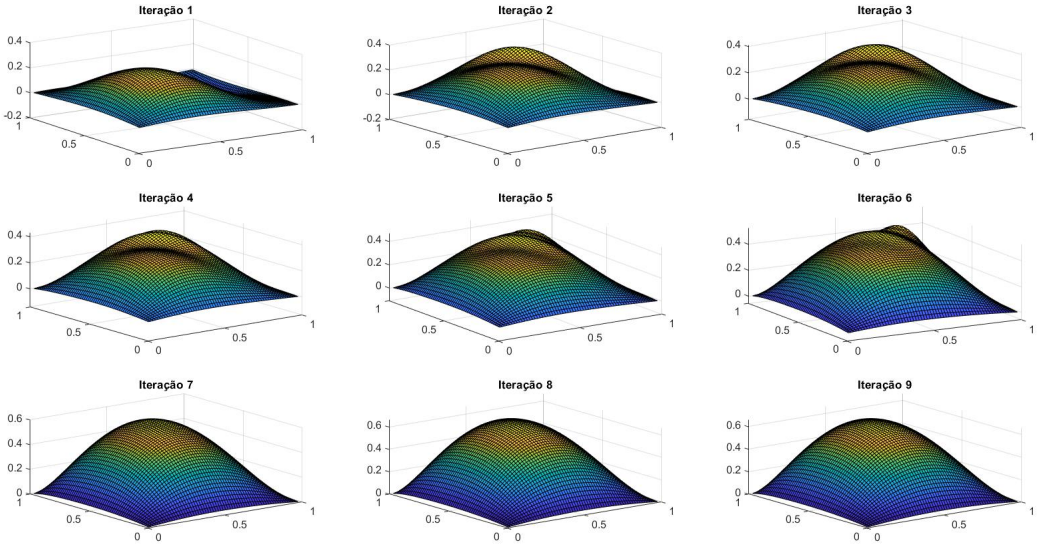

Figura 6: Comportamento das soluções aproximadas de u^* nas 9 primeiras iterações do método Newton-Inexato com $\lambda = 100$ e escolha E1.

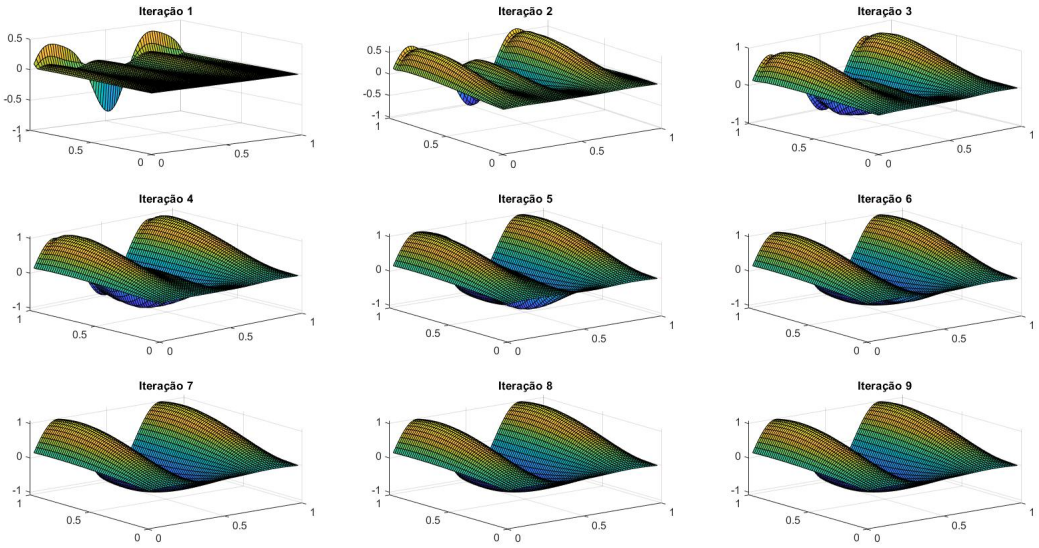

Figura 7: Comportamento das soluções aproximadas de u^{**} nas 9 primeiras iterações do método Newton-Inexato com $\lambda = 50$ e escolha E2.

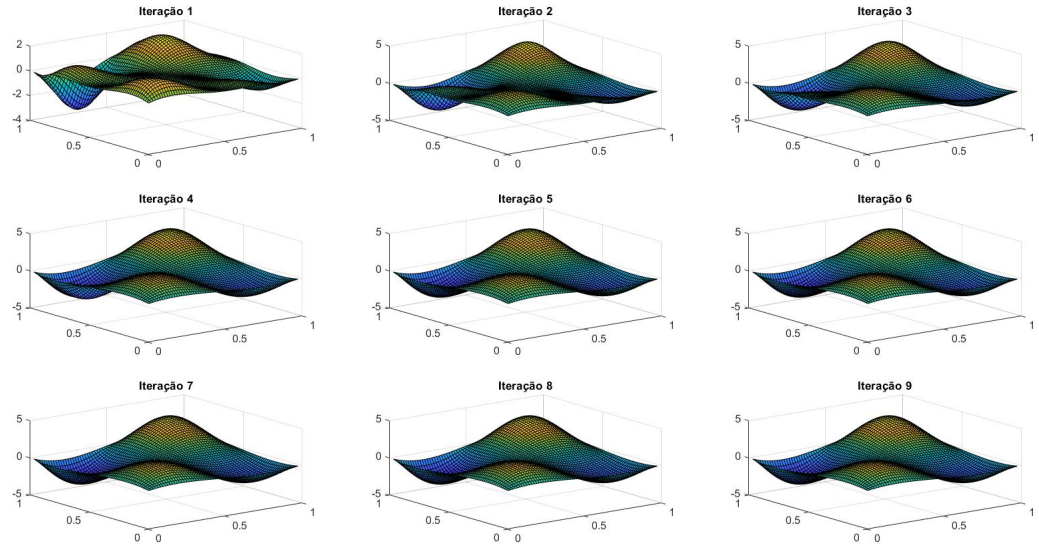

Figura 8: Comportamento das soluções aproximadas de u^{***} nas 9 primeiras iterações do método Newton-Inexato com $\lambda = 10$ e escolha E2.

4 Conclusão

Neste trabalho desenvolvemos a teoria do método Newton-GMRES e testamos sua eficiência frente à alguns problemas de grande porte com vários parâmetros. O método GMRES com restart foi essencial para explorarmos a estrutura esparsa de alguns sistemas lineares além de permitir uma diminuição do uso da memória da máquina. Além disso, podemos considerar a resolução de sistemas não-lineares como problemas de minimização onde podemos aplicar algumas estratégias de otimização.

Frente aos problemas estudados, as três escolhas utilizadas convergiram para a solução desejada (considerando a tolerância) com os parâmetros utilizados. Em alguns momentos o método GMRES(m) gerava direções de descida ruins aumentando o número de iterações realizadas. Observamos que as escolhas tiveram desempenhos diferentes em cada teste feito. No entanto, a escolha E3 apresentou ótimos resultados frente a todos os testes produzidos.

Para melhorar o desempenho do método Newton-GMRES, poderíamos considerar escolhas mais dinâmicas para o termo forçante, isto é, em vez de consideramos uma única escolha, poderíamos escolher E1 inicialmente e após um determinado tempo considerar E3 pela sua convergência quadrática numa vizinhança da solução do problema. Ou ainda, podemos utilizar outras escolhas mostradas em [2].

Ainda mais, o número de iterações até o recomeço do GMRES(m) também poderia ser mais flexível durante a aplicação do método. Também, poderíamos ter utilizado de precondicionadores para melhorar a eficiência do método.

Referências

[1] Rodolfo Gotardi BEGIATO. Um método Newton-Inexato com estratégia híbrida para globalização. *Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Cientifica, Campinas, SP.*, 2007.

[2] Julia Toledo BENAVIDES. Um método Newton-GMRES globalmente convergente com uma nova escolha para o termo forçante e algumas estrategias para melhorar o desempenho de gmres(m). *Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Cientifica, Campinas, SP.*, 2005.

[3] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in applied mathematics 16. Society for Industrial and Applied Mathematics, 1 edition, 1987.

[4] Elon Lages Lima. *Algebra Linear*. IMPA, 2014.

[5] David Bau III Lloyd N. Trefethen. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, 1997.

[6] Vera Lúcia da Rocha Lopes Márcia A. Gomes Ruggiero. *Cálculo Numérico: Aspectos Teóricos e Computacionais*. MAKRON, 2nd edition, 1996.

[7] Martin H. Saad, Youcef; Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.