



**COMAT – Coordenadoria do Curso de Licenciatura
em Matemática**

TCC – Trabalho de Conclusão de Curso

**Elaboração de um Código
Computacional para Resolução de
Sistemas Lineares de Grande Porte**

**Trabalho de Conclusão de
Curso elaborado como um
dos requisitos para conclusão
do Curso de Licenciatura em
Matemática da UFSJ.**

Orientador: Prof. Dr. Jorge Andrés Julca Avila

Orientanda: Rosângela Assis Pires

São João del-Rei, Dezembro de 2014

Sumário

1. Introdução.....	3
2. Resultados Preliminares.....	5
3. Método de Subespaço de Krylov.....	7
4. Método de Arnoldi.....	9
5. Método de Resíduo Mínimo Generalizado – GMRES.....	12
6. Precondicionadores.....	14
7. Testes Computacionais.....	16
8. Conclusão.....	18
Agradecimentos.....	19
Referências Bibliográficas.....	19

1 Introdução

A modelagem de muitos problemas das ciências e engenharias envolve Equações Diferenciais Parciais – EDP. Quando os problemas são mais próximos ao mundo real, a teoria matemática é mais difícil, pois são modelados por sistemas de EDP não lineares, e resolvê-los por métodos analíticos é quase impossível. Na procura de métodos de solução de EDP encontramos os métodos numéricos, os quais nos fornecem soluções aproximadas destas EDP. Também, procuramos métodos numéricos de acordo a nossa necessidade, por exemplo, a geometria do domínio de nosso problema é muitas vezes um fator importante para a escolha do método numérico. Os métodos numéricos mais usados são: Diferenças Finitas, Elementos Finitos e Volumes Finitos.

Problemas que envolvem variáveis temporais são, ainda, mais complexos porque estudam a evolução do estado ao longo do tempo. Já os problemas estacionários, os quais atingiram seu regime permanente e não dependem da variável temporal são mais “fáceis” resolvê-los e, por tal, seu custo computacional é mais barato.

Ao resolver numericamente um problema estacionário percebemos que o resultado final da discretização é um sistema algébrico. Se o problema for linear o sistema algébrico, também, será linear e, se o problema for não linear, o sistema algébrico também, será não linear. Quando não é linear deve-se linearizar-se. Uns dos métodos mais populares para linearizar sistemas algébricos não lineares é o Método de Newton (o qual transforma um sistema não linear em um número finito de sistemas lineares) desse modo sempre estaremos resolvendo sistemas lineares.

Para resolver numericamente um problema não estacionário ou temporal, ou ainda, transiente procede-se do seguinte modo: primeiro, se discretiza a variável espacial, desse modo obtemos um sistema de Equações Diferenciais Ordinárias – EDO. Segundo, se discretiza a variável temporal, isto é, a discretização da EDO. O resultado desta discretização é, novamente, um sistema algébrico (linear ou não linear, dependendo do problema) e como já foi explicado para o caso estacionário, tudo aponta à solução de sistemas lineares.

Desse modo existem muitas linhas de pesquisas direcionadas ao estudo de soluções de sistemas lineares e empresas investindo em códigos computacionais mais eficientes, robustos e com baixo custo computacional.

Sistemas Algébricos Lineares, matematicamente, podem ser representados por

$$Ax = b$$

onde $A \in \mathbb{R}^{n \times n}$ é uma matriz não singular, $x \in \mathbb{R}^{n \times 1}$ e $b \in \mathbb{R}^{n \times 1}$. Existem dois métodos de solução: métodos diretos e métodos iterativos.

Os *Métodos diretos* são quando a solução é obtida analiticamente através de manipulações algébricas e/ou um número finito de operações aritméticas. Estes métodos são aplicados a sistemas lineares que resultam de problemas não muito complicados, onde o número de variáveis envolvidas é pouco. Por exemplo, Eliminação Gaussiana, Matrizes Inversas, Fatoração de matrizes, etc.

Os *Métodos iterativos* são quando a solução é obtida numericamente através de aproximações sucessivas ou sequências. Estes são aplicados a sistemas que resultam de problemas reais de grande porte, isto é, o número de variáveis envolvidas no modelo é muito alto. Além disso, muitas vezes a matriz A é mal condicionada, em blocos e esparsa, ou seja, não possui uma estrutura bem definida.

Para inicialização dos métodos iterativos precisa-se de um valor inicial aproximado ou chute inicial, isto é,

$$x_0 \in \mathbb{R}^{n \times 1}$$

Podemos citar como exemplos desses métodos, os métodos de Jacobi, Gauss-Seidel e SOR (sobre relaxação). Assim, como também, os métodos de Gradiente Conjugado (onde a matriz é simétrica e definida positiva), e outros, mais modernos, como são os Métodos de Subespaços de Krylov (MSK) e suas variantes.

Um dos MSK é o método do Resíduo Mínimo Generalizado – GMRES que são aplicados a sistemas lineares não simétricos. Foi apresentado pela primeira vez por [SAAD e SCHULTZ \(1986\)](#). Do ponto de vista computacional, este método apresenta problemas de armazenamento, então se sugere o estudo de métodos com reinicialização – GMRES (m), veja [\[3\]](#). Quando a matriz é mal condicionada, algumas vezes os MSK podem se tornar muito lentos, até mesmo não convergir, nesse caso torna-se necessário o uso de pré-condicionadores, isto é, matrizes que transformem o sistema linear original em outro equivalente ao primeiro, mas com propriedades favoráveis a convergência.

Dentre as técnicas de pré-condicionamento podemos citar a *Fatoração LU Incompleta* (*ILU*) que consiste em transformar a matriz A num produto de uma matriz triangular inferior por uma matriz triangular superior, ou seja, $A = \tilde{L}\tilde{U}$, onde \tilde{L} é uma matriz triangular inferior e \tilde{U} é uma matriz triangular superior.

Neste trabalho deduziremos, analisaremos e implementaremos o Método de Resíduo Mínimo Generalizado – GMRES (m) com preconditionador do tipo Fatoração *LU* incompleta sem preenchimento – *ILU*. Para validação do código computacional consideraremos 16 testes com matrizes que resultam de resolver numericamente problemas de grande porte das ciências e da engenharia.

2 Resultados Preliminares

Definiremos alguns conceitos dos tópicos de álgebra linear.

Definição 1. (Sistema Algébrico Linear) Um sistema algébrico linear é definido da seguinte forma

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

onde $x \in \mathbb{R}^{n \times 1}$ é o vetor incógnita, $A \in \mathbb{R}^{n \times n}$ é a matriz não singular, $b \in \mathbb{R}^{n \times 1}$ é o vetor do lado direito.

Definição 2. (Matriz de Hessenberg Superior) Uma matriz $H = (h_{ij})$ é Hessenberg superior se $h_{ij} = 0$ para todo i, j tal que $i > j + 1$, isto é, todas as entradas abaixo da primeira subdiagonal são zeros.

Exemplo:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & \cdots & h_{1(m-2)} & h_{1(m-1)} & h_{1m} \\ h_{21} & h_{22} & h_{23} & h_{24} & \cdots & h_{2(m-2)} & h_{2(m-1)} & h_{2m} \\ 0 & h_{32} & h_{33} & h_{34} & \cdots & h_{3(m-2)} & h_{3(m-1)} & h_{3m} \\ 0 & 0 & h_{43} & h_{44} & \cdots & h_{4(m-2)} & h_{4(m-1)} & h_{4m} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & h_{(m-2)(m-2)} & h_{(m-2)(m-1)} & h_{(m-2)m} \\ 0 & 0 & 0 & 0 & \cdots & h_{(m-1)(m-2)} & h_{(m-1)(m-1)} & h_{(m-1)m} \\ 0 & 0 & 0 & 0 & \cdots & 0 & h_{m(m-1)} & h_{mm} \end{pmatrix}_{m \times m}$$

Definição 3. (Norma Euclidiana ou Norma-2) Para $x \in \mathbb{R}^n$ definimos a norma euclidiana:

$$\|x\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{1/2} \quad (2)$$

Definição 4. (Solução exata) Chamamos $y \in \mathbb{R}^n$ uma solução exata do sistema linear (1) se $Ay = b$.

Definição 5. (Solução aproximada) Chamamos $\tilde{x} \in \mathbb{R}^n$ uma solução aproximada do sistema linear (1) se

$$A\tilde{x} \approx b \quad (3)$$

Definição 6. (Método iterativo) Um método iterativo do sistema linear (1) é qualquer sequência $\{x_k\}_{k=1}^{\infty}$ em W (subespaço vetorial de \mathbb{R}^n), tal que, $Ax_k \approx b$.

Note que x_k é uma solução aproximada do sistema (1). Neste caso, x_k é chamada a k -ésima solução aproximada.

Alguns métodos iterativos são da forma

$$x_k = f(x_{k-1}), \quad k = 1, 2, 3, \dots \quad (4)$$

onde f é uma função. A equação (4) é chamada, algumas vezes, de esquema numérico do sistema (1).

Definição 7. (Resíduo) O resíduo do sistema linear (1) é definido por,

$$r := b - A\tilde{x} \quad (5)$$

onde \tilde{x} é a solução aproximada de (1).

Em métodos iterativos, o *resíduo inicial* de (1), é definido por

$$r_0 = b - Ax_0 \quad (6)$$

onde $x_0 \in \mathbb{R}^n$ é o *valor inicial* ou chute inicial de (1). O k -ésimo *resíduo* de (1) é definido por,

$$r_k = b - Ax_k \quad (7)$$

3. Método de Subespaço de Krylov

Apresentaremos o método de Subespaço de Krylov como um caso particular do Método de Projeção.

Definição 8. (Método de Projeção) Sejam \mathcal{K} e \mathcal{L} subespaços de \mathbb{R}^n , m -dimensionais, e \tilde{x} uma solução aproximada de (1). O método de Projeção sobre o subespaço \mathcal{K} e ortogonal a \mathcal{L} consiste em: Encontrar

$$\tilde{x} \in \mathcal{K}, \quad (8)$$

tal que,

$$b - A\tilde{x} \perp \mathcal{L} \quad (9)$$

onde \mathcal{L} é o *espaço das restrições* e, a condição de restrição (9) é chamada de *Condição de Petrov-Galerkin*.

No caso que exista um chute inicial x_0 a condição (8) será $\tilde{x} \in x_0 + \mathcal{K}$.

Note que \mathcal{K} e \mathcal{L} são subespaços arbitrários de \mathbb{R}^n por isso que o desafio consiste em construir apropriadamente estes espaços.

Definição 9. (Espaço de Krylov) Um espaço de Krylov é definido por

$$\mathcal{K}(A, r_0) = \text{Span}\{r_0, Ar_0, A^2r_0, \dots\} \quad (10)$$

Note que $\mathcal{K}(A, r_0)$ é um subespaço de \mathbb{R}^n .

Um *subespaço de Krylov* é definido por

$$\mathcal{K}_k(A, r_0) = \text{Span}\{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\} \quad (11)$$

Uma *matriz de Krylov* é definida por

$$K_k = \begin{bmatrix} r_0 & Ar_0 & A^2r_0 & \dots & A^{k-1}r_0 \end{bmatrix}_{m \times k} \quad (12)$$

Uma base natural para o subespaço de Krylov é dada por

$$\beta = \{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\} \quad (13)$$

Assim, qualquer $z \in \mathcal{K}_k(A, r_0)$ pode ser escrito como,

$$z = \left(\sum_{i=0}^{k-1} \alpha_i A^i \right) r_0 = (\alpha_{k-1} A^{k-1} + \dots + \alpha_1 A^1 + \alpha_0 I) r_0 = q_{k-1}(A) r_0 \quad (14)$$

onde q_{k-1} é um polinômio de grau, não maior que, $k-1$.

Por outro lado, se A é uma matriz invertível, então, de (3) e (6), temos que

$$\tilde{x} \approx x_0 + A^{-1}r_0 \quad (15)$$

Em (15), podemos obter A^{-1} aplicando o Teorema de Cayley-Hamilton à matriz A . Assim,

$$A^{-1}r_0 = (\alpha_{n-1}A^{n-1} + \dots + \alpha_1A^1 + \alpha_0I)r_0 \quad (16)$$

onde, os coeficientes podem ser obtidos pela equação característica de A . Logo, de (14),

$$A^{-1}r_0 \in \mathcal{K}_k(A, r_0) \quad (17)$$

E, portanto, de (15) e (16), temos que

$$\tilde{x} \in x_0 + \mathcal{K}_k(A, r_0) \quad (18)$$

Isso sugere a introdução dos espaços de Krylov na solução de sistema lineares.

O método de Subespaço de Krylov é um método de Projeção que procura uma solução aproximada x_k do sistema linear (1), no subespaço afim $x_0 + \mathcal{K}_k(A, r_0)$, impondo a condição de Petrov-Galerkin

$$b - Ax_k \perp \mathcal{L}_k \quad (19)$$

onde \mathcal{L}_k é o espaço de restrições de dimensão k . Existem vários métodos de subespaço de Krylov, que surgem através da escolha de \mathcal{L}_k . Por exemplo, quando $\mathcal{L}_k = \mathcal{K}_k(A, r_0)$

o método se subespaço de Krylov é chamado método dos *Resíduos Mínimos* – MINRES.

4. Método de Arnoldi

Antes de iniciar a busca pela solução aproximada x_k do sistema linear (1) no subespaço de Krylov é necessário construir uma base apropriada para este subespaço. A princípio poderíamos considerar a base β , dada em (13), porém a escolha não é adequada porque essa sequência de vetores converge para o autovetor associado ao maior autovalor, em módulo, da matriz A e, por isso, β tende a tornar-se linearmente dependente quando o número de operações de cálculo e precisão é alto.

Por outro lado, se tornarmos cada vetor de β como sendo ortonormal, teremos uma base mais estável e sem o problema de tornar-se linearmente independente. Para a ortonormalidade de β usaremos o *método de Ortonormalização de Arnoldi*, também conhecido como o *processo de Gram-Schmidt modificado*, que a seguir, descreveremos.

A partir de um vetor inicial e unitário

$$v_1 = \frac{r_0}{\|r_0\|_2} \quad (20)$$

o Método de Arnoldi gera vetores v_k através da ortonormalização de Av_{k-1} com relação aos vetores gerados previamente. O Algoritmo 1 de ortonormalização de Arnoldi é apresentado na Tabela 1.

Tabela 1. Algoritmo 1: Processo de ortonormalização de Arnoldi

Entrada: A , b e x_0

1. $r_0 = b - Ax_0$
 2. $v_1 = \frac{r_0}{\|r_0\|_2}$
 3. Para $j = 1, 2, 3, \dots$ fazer
 4. $v_{j+1} = Av_j$
 5. Para $i = 1, \dots, j$ fazer
-

-
6. $\eta(i, j) = \langle v_i, v_{j+1} \rangle$
 7. $v_{j+1} = v_{j+1} - \eta(i, j)v_i$
 8. Fim
 9. $\eta(j+1, j) = \|v_{j+1}\|_2$
 10. Se $\eta(j+1, j) = 0$. Então, pare
 11. $v_{j+1} = \frac{v_{j+1}}{\eta(j+1, j)}$
 12. Fim
-

Seja a matriz $H_k \in \mathbb{C}^{k+1 \times k}$, a matriz de Hessenberg superior que contém os coeficientes $\eta(i, j)$, onde k é o número de iterações feitas durante o processo de Arnoldi. Numa observação mais detalhada pode-se concluir o algoritmo atualiza v_{k+1} da seguinte forma:

Do Passo 11 temos que:

$$\eta(j+1, j)v_{j+1} = v_{j+1}$$

Utilizando o Passo 6 segue que $v_{j+1} = v_{j+1} - \eta(i, j)v_i$, logo:

$$\eta(j+1, j)v_{j+1} = v_{j+1} - \eta(i, j)v_i$$

Pelo o Passo 3 concluímos que:

$$\eta(j+1, j)v_{j+1} = Av_j - \eta(i, j)v_i$$

Então, após k iterações a fórmula acima será:

$$\eta(k+1, k)v_{k+1} = Av_k - \eta(1, k)v_1 - \eta(2, k)v_2 - \dots - \eta(k, k)v_k \quad (21)$$

ou,

$$\eta(1, k)v_1 + \eta(2, k)v_2 + \dots + \eta(k, k)v_k + \eta(k+1, k)v_{k+1} = Av_k \quad (22)$$

Logo,

$$Av_k = \sum_{i=1}^{k+1} \eta(i, k)v_i = V_{k+1}h_k \quad (23)$$

onde h_k é a k -ésima coluna da matriz H_k e V_{k+1} constitui uma base ortonormal formada pelo processo de Arnoldi para o subespaço de Krylov $\mathcal{K}_k(A, r_0)$. Generalizando para todos os vetores de V_k obtemos uma importante relação:

$$AV_k = V_{k+1}H_k \quad (24)$$

Proposição 1. *Seja $b \in \mathbb{C}^n$, então $\dim(\mathcal{K}_j)$ será menor ou igual a n , e, portanto existirá um primeiro valor k tal que $A^k b$ será linearmente dependente com os vetores anteriores da sequência de Krylov, isto é:*

$$A^k b = \sum_{i=0}^{k-1} \alpha_i A^i b \quad (25)$$

Como já foi dito, a sequência de vetores que constituem o subespaço de Krylov converge para um autovetor da matriz A , o que torna natural a ideia de que em alguma iteração, o vetor gerado pelo método de Arnoldi seja uma combinação linear dos gerados anteriormente e, conseqüentemente, ao ortonormalizarmos este novo vetor com os anteriores teremos o vetor nulo.

De acordo com a Proposição 1 e o Algoritmo 1, isso ocorrerá na iteração k , onde $k-1$ é o grau do polinômio mínimo do vetor b com relação a matriz A . Na iteração k teremos encontrado uma base $V_k = [v_1, v_2, \dots, v_k]$ para o subespaço de Krylov $\mathcal{K}_k(A, r_0)$. Como $\eta(j+1, j) = \|v_{j+1}\|_2$ e para algum valor k , $k \in \mathbb{N}$, temos que $\eta(k+1, k) = 0$ então $\|v_{k+1}\|_2 = 0$. Assim, chega-se à conclusão que o critério de parada estabelecido pelo Algoritmo 1 será satisfeito apenas quando obtiver uma base para o subespaço $\mathcal{K}_k(A, r_0)$, onde se encontra uma solução x_k muito próxima da solução exata do sistema (1) sendo capaz de satisfazer o teste de convergência do método iterativo, que utiliza o processo de Arnoldi na construção de uma base ortonormal.

5. Método do Resíduo Mínimo Generalizado – GMRES

O método do Resíduo Mínimo Generalizado (GMRES) veja [3], é um método de projeção *ortogonal* baseado em subespaços de Krylov, que apresenta as seguintes características:

Considere o sistema linear (1). Para resolver esse sistema o GMRES parte de um valor inicial x_0 e calcula o resíduo inicial $r_0 = b - Ax_0$. \mathcal{K}_k será o subespaço de Krylov:

$$\mathcal{K}_k(A, r_0) = \text{Span} \{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\} \quad (26)$$

onde $(x_k - x_0) \in \mathcal{K}_k(A, r_0)$, sendo x a solução exata do sistema (1) e x_k a solução aproximada do mesmo sistema obtida na k -ésima iteração feita pelo GMRES. O espaço de restrição será $\mathcal{L}_k = A\mathcal{K}_k(A, r_0)$, ou seja,

$$\begin{cases} x_k \in x_0 + \mathcal{K}_k \\ r_k \perp \mathcal{L}_k \end{cases} \Leftrightarrow \|r_k\|_2 = \min_{z_k \in \mathcal{K}_k} \|r_0 - Az_k\|_2 \quad (27)$$

Considere a matriz $V_k = [v_1 \ v_2 \ \dots \ v_k]$ como sendo uma base ortonormal do subespaço de Krylov \mathcal{K}_k construída a partir do processo de ortonormalização de Arnoldi, veja a Tabela 1. Temos que $x_k \in \mathcal{K}_k(A, r_0)$, então x_k pode ser escrito como $x_k = x_0 + c_k$, onde $c_k \in \mathcal{K}_k(A, r_0)$ e sendo V_k uma base desse subespaço c_k é da seguinte forma:

$$c_k = \sum_{i=1}^k y_i v_i \quad (28)$$

onde os y_i , $i = 1, \dots, k$, são os escalares ainda desconhecidos que tornam a equação (28) verdadeira. Também podemos escrever a igualdade acima na forma de matriz:

$$c_k = V_k y_k \quad (29)$$

Neste caso, y_k será o vetor cujos elementos são os escalares y_i ($i = 1, \dots, k$).

Logo, $x_k = x_0 + c_k$ pode ser representado por:

$$x_k = x_0 + V_k y_k \quad (30)$$

O GMRES utiliza-se da equação (30) para calcular a solução aproximada e da equação $r_k = b - Ax_k$ para calcular o resíduo em cada iteração.

Sem perda de generalidade vamos definir x_0 como sendo o vetor nulo. Depois de calcular a matriz V_k , a segunda etapa será encontrar os valores de y_k para fazer uso da equação (30). Para isso, considere que em cada iteração k do GMRES o valor esperado da norma euclidiana do resíduo seja mínimo.

$$\|r_k\|_2 = \min_{x_k \in \mathcal{K}_k} \|b - Ax_k\|_2$$

Utilizando as seguintes equações:

$$r_0 = b - Ax_0 \Rightarrow r_0 = b$$

$$v_1 = \frac{r_0}{\|r_0\|_2} \Rightarrow \|r_0\|_2 v_1 = r_0 = V_{k+1} (\|r_0\|_2 e_1)$$

Sendo e_1 o primeiro vetor da base canônica e $V_{k+1} = [v_1 \ v_2 \ \dots \ v_k \ v_{k+1}]$ é a matriz ortonormal, cujas colunas são formadas pelos vetores ortonormalizados pelo método de Arnoldi e, representa uma base para o subespaço de Krylov $\mathcal{K}_{k+1}(A, r_0)$. De (30),

$$x_k = x_0 + V_k y_k \Rightarrow x_k = V_k y_k$$

De (24), temos que:

$$\|r_k\|_2 = \min_{x_k \in \mathcal{K}_k} \|b - Ax_k\|_2 = \min_{y_k \in \mathbb{R}^k} \|r_0 - AV_k y_k\|_2 = \min_{y_k \in \mathbb{R}^k} \|V_{k+1} (\|r_0\|_2 e_1) - V_{k+1} H_k y_k\|_2 \quad (31)$$

Para que o valor de $\|r_k\|_2$ seja mínimo é necessário que $\|V_{k+1} (\|r_0\|_2 e_1) - V_{k+1} H_k y_k\|_2$ seja muito próximo de zero. E como V_{k+1} é uma matriz ortonormal segue que $V_{k+1}^T V_{k+1} = V_{k+1} V_{k+1}^T = I$, onde I é a matriz identidade, então a matriz V_{k+1} possui inversa e sua inversa será V_{k+1}^T .

$$\begin{aligned} V_{k+1} H_k y_k - V_{k+1} (\|r_0\|_2 e_1) &\cong 0 \Rightarrow V_{k+1} H_k y_k \cong V_{k+1} (\|r_0\|_2 e_1) \\ &\Rightarrow V_{k+1}^{-1} V_{k+1} H_k y_k \cong V_{k+1}^{-1} V_{k+1} (\|r_0\|_2 e_1) \end{aligned} \quad (32)$$

Assim,

$$H_k y_k \cong \|r_0\|_2 e_1 \quad (33)$$

A equação (33) é, também, um sistema linear e pode ser resolvida através de uma série de rotações de Givens, sendo um problema de fácil resolução comparado ao sistema (1) devido à estrutura da matriz de Hessenberg superior H_k .

Uma vez determinado o vetor y_k , a terceira e última etapa é usar a equação (30) para determinar a solução aproximada x_k e calcular o seu respectivo resíduo r_k em cada iteração.

As três etapas são repetidas em cada iteração k dentro de um laço que termina quando o teste de convergência for satisfeito.

É importante ressaltar que o GMRES não pode fazer infinitas iterações, por isso deve-se estabelecer uma determinada tolerância para o resíduo, isto é, escolher um número muito próximo de zero para que o método faça uma série de iterações até que o resíduo seja menor ou igual à tolerância.

Quando a dimensão da matriz A é muito alta o GMRES ocupa muito espaço de armazenamento, então para amenizar tal problema foi criado o GMRES com reinicialização, denotado por GMRES (m), e consiste na escolha de um m ($m < n$), com o qual será gerada uma base ortonormal para encontrar a solução aproximada de (1). Quanto à escolha de um m ótimo podemos citar *Saad*.

6. Precondicionadores

O preconditionamento é o processo de transformação do sistema linear (1) em outro equivalente, porém mais simples. Sendo, muitas vezes, indispensável no sentido de melhorar a taxa de convergência do método.

Ao aplicar-se o preconditionamento em um sistema linear espera-se que a matriz M , chamada preconditionador, seja uma boa aproximação da matriz A e, desta forma, AM^{-1} deve ser uma boa aproximação da matriz identidade. Resolver um sistema do tipo $y = M^{-1}c$ costuma não ser viável, uma vez que o cálculo de inversa de matrizes pode ser muito trabalhoso, ocupar um grande espaço de armazenamento e o custo

computacional seja alto. Desse modo o mais vantajoso seria a resolução do sistema $My = c$.

Como não é possível construir um preconditionador que resolva um problema de forma genérica, esta área de pesquisa está aberta ao desenvolvimento de novas técnicas, uma vez que a qualquer momento podem surgir aplicações que necessitam de preconditionadores mais eficientes para o tipo de problema em questão.

A construção de um preconditionador está associada ao fato de poder reutilizá-lo em outros problemas, bem como a necessidade de acelerar a convergência do método utilizado, para compensar o fato de que ao trabalhar com preconditionadores o custo e o armazenamento é maior se comparado com métodos iterativos sem preconditionadores.

Um tipo de preconditionador muito utilizado é a Fatoração LU Incompleta (ILU) que consiste em aproximar a matriz A de um produto de uma matriz triangular inferior por uma matriz triangular superior, ou seja, $A = \tilde{L}\tilde{U}$, onde \tilde{L} é uma matriz triangular inferior e \tilde{U} é uma matriz triangular superior, fazendo assim $M = \tilde{L}\tilde{U}$, veja [6]. Dentre as técnicas de fatoração ILU , podemos destacar a Fatoração $ILU(0)$, neste caso, as posições nulas da matriz A , isto é, a posição dos elementos nulos da matriz A , são preservadas nas matrizes triangulares. A fatoração $ILU(0)$ também é conhecida como fatoração sem preenchimento.

Na Tabela 2, encontrar-se o pseudocódigo do GMRES (m) com a versão do preconditionador $ILU(0)$, baseado principalmente nas referências [5] e [6].

Tabela 2. Pseudocódigo do GMRES (m), com o preconditionador *ILU* (0).

<p>GMRES (m)</p> <p>Entrada : A, b, x_0, M, Saída : x</p> <p>Faça: $r = b - Ax_0, \beta = \ r\ _2$</p> <p>Faça enquanto: $j < \text{iterações}$ e $\beta > \varepsilon$</p> <p>$v_1 = r / \beta, \hat{b} = \beta e_1$</p> <p>Para $i = 1, m$</p> <p style="padding-left: 20px;">$Mz_i = v_i (z_i = M^{-1}v_i), w = Az_i$</p> <p>Para $k = 1, i$</p> <p style="padding-left: 20px;">$h_{k,i} = v_k^T w, w = w - h_{k,i}v_k$</p> <p style="padding-left: 20px;">$h_{i+1,i} = \ w\ _2, v_{i+1} = w / h_{i+1,i}$</p> <p style="padding-left: 20px;">$r_{1,i} = h_{1,i}$</p> <p>Para $k = 2, i$</p> <p style="padding-left: 20px;">$\gamma = c_{k-1}r_{k-1,i} + s_{k-1}h_{k,i}$</p> <p style="padding-left: 20px;">$r_{k,i} = -s_{k-1}r_{k-1,i} + c_{k-1}h_{k,i}$</p> <p style="padding-left: 20px;">$r_{k-1,i} = \gamma, \delta = \sqrt{r_{i,i}^2 + h_{i+1,i}^2}$</p> <p style="padding-left: 20px;">$c_i = r_{i,i} / \delta, s_i = h_{i+1,i} / \delta,$</p> <p style="padding-left: 20px;">$r_{i,i} = c_i r_{i,i} + s_i h_{i+1,i}$</p> <p style="padding-left: 20px;">$\hat{b}_{i+1} = -s_i \hat{b}_i, \hat{b}_i = c_i \hat{b}_i,$</p> <p style="padding-left: 20px;">$\wp = \hat{b}_{i+1}$</p> <p>Se $\wp \leq \varepsilon_1$ então</p> <p style="padding-left: 20px;">$nr = i, \text{Vá para } 10$</p> <p style="padding-left: 20px;">$nr = m$</p> <p style="padding-left: 20px;">$y_{nr} = \hat{b}_{nr} / r_{nr,nr}$</p>	<p>10 Para $k = nr - 1, 1$</p> <p style="padding-left: 20px;">$y_k = (\hat{b}_k - \sum_{i=k+1}^{nr} r_{k,i}y_i) / r_{k,k}$</p> <p style="padding-left: 20px;">$x = x + \sum_{i=1}^{nr} y_i z_i$</p> <p style="padding-left: 20px;">$r = b - Ax, \beta = \ r\ _2$</p> <p>Fim</p> <p>Precondicionador <i>ILU</i>(0)</p> <p>Entrada: A, Saída: $M = \tilde{L}\tilde{U}$</p> <p>Para $i = 2, n$</p> <p style="padding-left: 20px;">Para $k = 1, i - 1$</p> <p style="padding-left: 40px;">Se $a_{i,k} \neq 0$ e $a_{k,k} \neq 0$ faça</p> <p style="padding-left: 60px;">$a_{i,k} = a_{i,k} / a_{k,k}$</p> <p style="padding-left: 20px;">Para $j = k + 1, n$ faça</p> <p style="padding-left: 40px;">$a_{i,j} = a_{i,j} - a_{i,k}a_{k,j}$</p> <p>Para $i = 1, n$</p> <p style="padding-left: 20px;">Para $j = 1, n$</p> <p style="padding-left: 40px;">Se $i \leq j$ então $\tilde{u}_{i,j} = a_{i,j}$</p> <p style="padding-left: 40px;">Caso contrário, $\tilde{u}_{i,j} = 0$</p> <p>Para $i = 1, n$</p> <p style="padding-left: 20px;">Para $j = 1, n$</p> <p style="padding-left: 40px;">Se $i > j$ então $\tilde{l}_{i,j} = a_{i,j}$</p> <p style="padding-left: 40px;">Se $i = j$ então $\tilde{l}_{i,j} = 1$</p> <p style="padding-left: 40px;">Caso contrário, $\tilde{l}_{i,j} = 0$</p> <p>Fim</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

7. Testes Computacionais

Os testes realizados para resolução de sistemas lineares utilizam matrizes extraídas do “Matrix Market”, [4]. Estas matrizes resultam de modelar problemas de grande porte, como por exemplo: problemas de convecção-difusão 2D, elementos finitos na resolução de operadores bi-harmônicos em placas retangulares, simulação de reservatórios, problemas de reação-difusão, reatores tubulares, etc. para outros testes veja [2].

O método GMRES (m) e o método GMRES com o preconditionador $ILU(0)$, apresentado na Tabela 2, foram implementados na Linguagem Fortran.

A execução dos programas foram feitos em um computador Intel Core i5, 2.20 GHz de CPU, 4 GB de memória RAM e sistema operacional Windows 8.

Foram considerados sistemas lineares cuja solução era conhecida. O Número de iterações máximo foi igual a 1000 e a Tolerância $\varepsilon = \varepsilon_1 = 10^{-6}$, o valor de m escolhido em todos os testes foi $m = 36$. Na Tabela 3, encontra-se o valor do erro (isto é, $\|x - \tilde{x}\|_2$, onde x é a solução exata do sistema e \tilde{x} é a solução aproximada obtida pelo método iterativo) gerado pelo GMRES(m) com e sem preconditionador.

Tabela 3: Erro gerado pelo GMRES(m) com e sem preconditionador.

Teste	Matriz	Dimensão	GMRES(m)	GMRES(m)
				Precondicionado $ILU(0)$
01	Add20	2395×2395	$1,50 \times 10^{-2}$	$1,62 \times 10^{-6}$
02	Arc130	130×130	$8,33 \times 10^{-3}$	$4,00 \times 10^{-5}$
03	Bcsstk01	48×48	$1,22 \times 10^{-12}$	$2,08 \times 10^{-14}$
04	Bcsstm02	66×66	≈ 0	≈ 0
05	Bcsstm22	138×138	$1,71 \times 10^{-2}$	≈ 0
06	Cdde1	961×961	$4,23 \times 10^{-5}$	$2,39 \times 10^{-7}$
07	Cdde3	961×961	$4,07 \times 10^{-4}$	$2,73 \times 10^{-7}$
08	Cdde4	961×961	$3,94 \times 10^{-6}$	$2,36 \times 10^{-7}$
09	Nos3	960×960	$1,36 \times 10^{-8}$	$4,95 \times 10^{-9}$
10	Nos4	100×100	$2,51 \times 10^{-6}$	$4,04 \times 10^{-7}$
11	Nos6	675×675	$9,22 \times 10^{-7}$	$1,41 \times 10^{-8}$
12	Odep400a	400×400	$2,73 \times 10^2$	$6,76 \times 10^{-7}$
13	Pores2	1224×1224	$2,07 \times 10$	$7,97 \times 10^{-11}$
14	Pores3	532×532	$2,82 \times 10^{-6}$	$4,84 \times 10^{-8}$
15	Rdb200	200×200	$1,01 \times 10^{-8}$	$9,38 \times 10^0$
16	Tub100	100×100	$1,34 \times 10$	$4,33 \times 10^{-8}$

Na maioria dos casos, percebe-se um melhor desempenho do GMRES(m) com preconditionador $ILU(0)$, destacando-se os resultados obtidos com as matrizes dos Testes 12 e 13, cujo desempenho foi bem superior ao desempenho do GMRES(m). Há também, o caso da matriz do Teste 15, na qual o GMRES(m) apresentou uma maior proximidade da solução exata do sistema se comparado com o GMRES(m) com preconditionador $ILU(0)$. E em alguns casos, observa-se um desempenho semelhante entre os dois métodos iterativos.

Lembrando que os testes realizados fizeram uso somente de matrizes cujos sistemas lineares eram resolvidos por ambos os métodos, pois o objetivo aqui era observar o desempenho do GMRES(m) e do GMRES(m) com preconditionador $ILU(0)$.

8. Conclusão

Nesse trabalho, deduzimos, analisamos e implementamos o Método de Resíduo Mínimo Generalizado GMRES com reinicialização – GMRES (m). Quando se observou dificuldade de aceleração da convergência, estendeu-se para o estudo do GMRES (m) com preconditionador do tipo Fatoração LU incompleta sem preenchimento – $ILU(0)$. Esses métodos iterativos, baseados em Subespaços de Krylov, resolvem sistemas lineares não simétricos, esparsos e mal condicionados que resultam ao resolver numericamente problemas de grande porte da ciência e da engenharia.

O Método de Subespaço de Krylov – MSK procurou uma solução aproximada de um determinado sistema linear em um subespaço de Krylov, impondo restrições que facilitam essa busca. O preconditionador é uma forma de melhorar ou acelerar a convergência de um MSK, claro que, cada problema exige um tipo de preconditionador diferente que melhor se adapte as características da matriz dos coeficientes para que o preconditionador cumpra a sua função e, conseqüentemente seja vantajoso o gasto com a sua construção.

Agradecimentos

Agradeço a Deus pelas forças e a minha família pelo apoio constante que obtive durante todos esses anos de estudo.

Agradeço aos professores do DEMAT que souberam me ensinar e particularmente ao prof. Dr. Jorge Andrés Julca Avila, pelo apoio constante na minha vida profissional e acadêmica.

Também agradeço ao INCTmat/CNPq pelo apoio financeiro prestado durante o ano 2013.

Referências Bibliográficas

- [1] Carvalho, Luiz M. *Avanços em Métodos de Krylov para Solução de Sistemas Lineares de Grande Porte*. São Carlos, SP: SBMAC, 2009, 148 p., 20.5 cm – (Notas em Matemática Aplicada, vol. 42).
- [2] Gonzalez, Tífani T. *Algoritmos Adaptativos para o método GMRES(m)*. Porto Alegre: PPGMAp da UFRGS, 2005.
- [3] Lago, Rafael F. *Estudos Sobre os Métodos Iterativos de Krylov para Solução de Sistemas de Equações Lineares*. Rio de Janeiro: UFRJ/COPPE, 2010.
- [4] Matrix Market. Disponível em: < <http://math.nist.gov/MatrixMarket/> >. Acesso em: 05 Nov. 2014.
- [5] Saad, Y. e Schultz, M. H. *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM J. Sci. Stat. Comput., 7(3):856–869, 1986.
- [6] Van der Vorst, Henk A., *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, New York, 2003.