

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Departamento de Informática

André Barreto

Igor Ventrini

**Estudo Sobre a Influência do
Reordenamento e Precondicionamento
aplicados a Sistemas Esparsos de Grande
Porte Utilizando Métodos Iterativos Não
Estacionários**

Trabalho 2 de Algoritmos Numéricos 2

Vitória

2016

1 Introdução

Frequentemente os processos de solução de problemas das mais diversas áreas do conhecimento recaem na necessidade de resolver sistemas lineares. Na maioria das vezes esses sistemas são esparsos e de grande porte. Nesse contexto, o uso de métodos diretos torna-se desaconselhável, por limitações de tempo e memória. Em detrimento, as abordagens iterativas tornam-se mais atraentes.

Neste trabalho, será estudado o GMRES, método iterativo não estacionário, que caracteriza-se por buscar uma aproximação da solução dos sistemas lineares a partir de um subespaço. Este método também é conhecido como livre de matriz.

Os métodos iterativos não estacionários, apesar de bem fundamentados teoricamente, podem sofrer convergência lenta para problemas que surgem de aplicações típicas. Para resolver tal questão, deve-se considerar o uso de bons preconditionadores, que melhoram o condicionamento de uma matriz, acelerando a convergência. Além disto, deve-se considerar o uso de processos de reordenamento, que são simples trocas de linhas, reduzindo o número de operações com ponto flutuante durante a montagem das matrizes preconditionadoras.

Em suma, este trabalho tem como objetivo realizar um estudo sobre a utilização de métodos iterativos não estacionários no contexto de matrizes esparsas de grande porte e verificar a influência do ordenamento e o preconditionamento aplicado a essas matrizes.

Neste contexto, realizar tal estudo requer experimentos práticos com resultados visuais. Para tanto, na seção 3 serão apresentados experimentos numéricos para tal análise. Antes desta, na seção 2, encontram-se resumos dos métodos e técnicas abordados. E por fim, a seção 4 são discutidos brevemente os resultados obtidos.

2 Referencial Teórico

2.1 Sistemas lineares de grande porte

Sistemas lineares de grande porte aparecem como resultado da modelagem de vários problemas nas engenharias. A busca de métodos para resolução destes sistemas é muito abordada pela Ciência da Computação. Neste trabalho, resolveremos sistemas lineares $Ax = b$ esparsos de grande porte.

2.2 Técnica de Compactação de matriz CSR

O método de armazenamento de matrizes esparsas Compressed Sparse Row(CSR) é um método de armazenamento de matrizes esparsas que utiliza três vetores para armazenar os valores não nulos da matriz assim denominados *valA*, *rowPtrA* e *colIndA*. O vetor *AA* armazena os valores não nulos da matriz, o vetor *IA* armazena os índices identificando a posição no vetor *AA* onde se inicia uma nova linha da matriz e o vetor *JA* armazena os índices referentes às colunas em que os elementos não nulos estão posicionados na matriz.

Exemplo:

$$\begin{aligned}AA &= [1, 1, 4, 2, 4, 1, 2, 1, 8, 2, 4, 1, 3, 6, 2, 1] \\JA &= [0, 1, 4, 1, 2, 4, 0, 1, 2, 3, 2, 3, 0, 1, 2, 4] \\IA &= [0, 3, 6, 10, 13, 17]\end{aligned}$$

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 4 \\ 0 & 2 & 4 & 0 & 1 \\ 2 & 1 & 8 & 2 & 0 \\ 0 & 0 & 4 & 1 & 0 \\ 3 & 6 & 2 & 0 & 1 \end{bmatrix}$$

Utilizaremos o método CSR uma vez que os sistemas propostos são esparsos. O armazenamento de forma otimizada é crucial na eficiência do método numérico.

2.3 Precondicionamento

O objetivo dos pré condicionadores é acelerar a convergência dos métodos iterativos, de forma que dado um sistema $Ax = b$, podemos encontrar uma matriz M tal que $M^{-1}Ax = M^{-1}b$ onde M será o pré condicionador. Deve-se levar em conta que seja uma boa aproximação para A e seja fácil de implementar. Caso essas características não sejam satisfeitas, acaba se tornando inviável o uso de pré condicionadores.

Existem vários tipos de implementação de pré condicionadores. Neste trabalho foi utilizado o pré condicionador baseado da fatoração LU incompleta (ILU) tal que $M = LU$ onde L e U são fatorações LU aproximadas de A . Para pré condicionadores baseados na fatoração incompleta (ILU) é necessário armazenamento extra para L e U , a operação do pré condicionador se dá na operação produto matriz-vetor do método iterativo.

Para aplicação dos pré condicionadores no trabalho foram criadas três funções principais a partir de que se tenha as Matrizes L e U.

Esta função é responsável por gerenciar as funções que realizam as operações necessárias para aplicação no método GMRES, resolvendo a matriz triangular superior e inferior.

```

1 void preconditionLU(MAT* L, MAT* U, double* vet_out, double* vet, int
    N)
2 {
3     // Precondicionamento:
4     //  $Ax = b \longrightarrow MAx = Mb$ 
5     //  $Mb' = b$ 
6     //
7     // Resolver:
8     //  $LUvet\_out = vet$ 
9
10    double *q = createVector(N); // Vetor auxiliar
11
12    //  $Lq = vet$ 
13    MATRIX_forward(L, q, vet);
14
15    //  $Uvet\_out = q$ 
16    MATRIX_backward(U, vet_out, q);
17
18    free(q);
19 }

```

Listing 1: Aplicação Pré Condicionamento LU

Função que resolve a matriz triangular superior armazenada com CSR.

```

1 // -----
2 // * Solve CSR matrix by successive substitutions
3 // * -----
4 void MATRIX_forward(MAT* L, double* x, double *b)
5 {
6     int i, j, N;
7     double soma = 0;
8
9     N = L->n;
10    x[0] = b[0]/L->D[0];
11    for(i = 1; i < N; i++){
12        soma = 0;
13        for(j = L->IA[i]; j < L->IA[i+1]; j++){
14            soma += x[L->JA[j]] * L->AA[j];
15        }
16        x[i] = (b[i] - soma)/L->D[i];
17    }
18 }

```

Listing 2: Matrix Forward

Função que resolve a matriz triangular inferior armazenada com CSR.

```

1 // -----
2 // * Solve CSR matrix by retro substitutions
3 // -----
4 void MATRIX_backward(MAT* U, double* x, double *b)
5 {
6     int i, j, N;
7     double soma;
8
9     N = U->n;
10    x[N-1] = b[N-1]/U->D[N-1];
11    for (i = N-2; i >=0; i--){
12        soma = 0;
13        for (j = U->IA[i]; j < U->IA[i+1]; j++){
14            soma += x[U->JA[j]] * U->AA[j];
15        }
16        x[i] = (b[i] - soma)/U->D[i];
17    }
18 }

```

Listing 3: Matrix Backward

2.4 Reordenamento

O pré condicionamento de matrizes pode ocasionar em preenchimento, ou seja elementos nulos em A se tornam não nulo em M . Uma solução é o uso de reordenamento para reduzir o preenchimento causado pelo pré condicionador baseado na fatoração ILU, reduzindo o preenchimento consequentemente reduziremos o tempo de computação. Após o uso do reordenamento e encontrada uma solução do sistema, é necessário reordenar novamente para que se encontre o vetor solução original.

Existem várias técnicas de reordenamento para a redução de *fill-ins* (preenchimentos). A técnica proposta neste trabalho é a RCM, que apresenta uma boa qualidade de solução, baixo tempo de execução e fácil implementação. O RCM trabalha tentando aproximar os elementos não nulos da diagonal principal, o que resulta em uma redução de *fill-ins*. Para saber quão próximo os elementos não nulos estão da diagonal principal foram estabelecidas duas métricas: a largura de banda e o envelope. A largura de banda pode ser definida como a maior distância entre o primeiro elemento não-nulo da linha i até a diagonal principal. Já o envelope como a soma das distâncias entre o primeiro elemento não-nulo de cada linha i até a diagonal principal.

2.5 Métodos Iterativos Não Estacionários

Os métodos iterativos não estacionários diferem dos métodos estacionários pois os operadores matriciais que atuam sobre a aproximação da solução x^{k-1} para produzir x^k não são constantes ao longo do processo iterativo.

A matriz dos coeficientes A e o vetor dos termos independentes b não são alterados durante o processo iterativo e dependem de critérios de convergência relacionados a matriz dos coeficientes A , sendo seu objetivo transformar o sistema $Ax = b$ em um problema de minimização do resíduo $\|r_k\| = \min\|b - Ax_k\|$.

2.5.1 GMRES

O GMRES (Generalized Minimal Residual) foi proposto por Saad e Schultz, em 1986. Ele se diferencia do método dos gradientes conjugados por não ser restrito a matrizes simétricas definidas positivas. Porém, com isso temos que o uso dos polinômios residuais torna mais difícil, assim como não temos garantia da aplicação do teorema espectral. Tem como objetivo minimizar a norma do resíduo do sistema $x = \min\|b - Ax_k\|$. Considerando x_0 solução inicial, uma aproximada x_k é dada por $x_k = x_0 + z$ onde z é um vetor do espaço de Krylov $K^k = \text{span}(r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0)$

$$\min(x \in x_0 + K^k)(\|b - Ax\|)_2$$

Cada iteração do método GMRES é muito cara. Para torná-lo viável computacionalmente é adotado o processo de reinicializar a formação dos vetores da base a cada k iterações.

Supondo $k = 10$

$$\begin{aligned} \text{Dados : } & A, b, x_0 \\ & V = [u_1, u_2, \dots, u_{10}] \\ & y = (y_1, y_2, \dots, y_{10}) \\ & x_{10} = x_0 + \sum_{j=1}^{10} y_j u_j \end{aligned}$$

O Algoritmo com esta modificação é chamado de GMRES com Restart, o qual foi utilizado para a realização do trabalho.

Entrada: $A, b, lmax, x = 0, kn$

Saída: x

```

1 início
2    $\epsilon = tol(||b||)_2;$ 
3    $l = 1;$ 
4   repeat
5      $i = 1, u_i = b - Ax, e_i = ||u_i||, u_i = u_i/e_i, p = e_i;$ 
6     while  $p > \epsilon$  e  $i < k$  do
7        $u_{i+1} = Au_i;$ 
8       for  $j \leftarrow 1$  to  $i$  do
9          $h_{ji} = u_{i+1}^t u_j;$ 
10         $u_{i+1} = u_{i+1} - h_{ji} u_j;$ 
11         $h_{i+1,i} = (||u_{i+1}||)_2;$ 
12         $u_{i+1} = u_{i+1}/h_{i+1,i};$ 
13        for  $j \leftarrow 1$  to  $i - 1$  do
14           $h_{ji} = c_j h_{ji} + s_j h_{j+1,i};$ 
15           $h_{j+1,i} = -s_j h_{ji} + c_j h_{j+1,i};$ 
16           $r = \sqrt{h_{ii}^2 + h_{i+1,i}^2};$ 
17           $c_i = h_{ii}/r;$ 
18           $s_i = h_{i+1,i}/r;$ 
19           $h_{ii} = r(h_{i+1,i} = 0);$ 
20           $e_{i+1} = -s_i e_i;$ 
21           $e_i = c_i e_i;$ 
22           $p = |e_{i+1}|;$ 
23           $i = i + 1;$ 
24         $i = i - 1;$ 
25        for  $j \leftarrow i$  to  $1$  do
26           $y_j = (e_j - \sum_{l=j+1}^i h_{jl} y_l)/h_{jj};$ 
27           $x = \sum_{j=1}^i y_j u_j;$ 
28           $l = l + 1;$ 
29  until  $p < \epsilon$  e  $l \geq lmax;$ 

```

Algoritmo 1: GMRES c/Restart

Como as matrizes que estão sendo trabalhadas tem tratamentos bem particulares e foram armazenadas com a técnica de compactação de matriz CSR, é necessário que tenham funções específicas para realizar algumas operações. Uma operação de grande relevância no método GMRES é a operação produto matriz vetor que pode ser vista nas linhas 5 e 7 do algoritmo.

Segue a baixo a implementação do algoritmo produto matriz vetor em

linguagem C para uma matriz armazenada com o método CSR:

```
1 //  


---

  
2 // Compute CSR matrix vector product  
3 //  


---

  
4 void MATRIX_matvec(MAT *A, double* x, double *b)  
5 {  
6     int i, j;  
7  
8     for( i = 0; i < A->n; i++)  
9     {  
10         for( j = A->IA[i]; j < A->IA[i+1]; j++)  
11         {  
12             x[i] += A->AA[j] * b[A->JA[j]];  
13         }  
14     }  
15 }
```

Listing 4: Produto Matriz Vetor (CSR)

Observação: para alterar a tolerância do GMRES, o número máximo de ciclos e o parâmetro do pré-condicionador ILU, deve-se alterar o código do arquivo *trab2.c*, nas linhas indicadas a seguir:

```
1 #define P      0          // Parametro de ILU(P)  
2 #define TOL    0.000001   // Tolerancia do GMRES  
3 #define LMAX 1000        // Num. maximo de ciclos do GMRES
```

Listing 5: Definições de constantes

3 Experimentos Numéricos

Nesta seção será feito um estudo do método GMRES, com e sem preconditionamento e com e sem ordenamento RCM, ao avaliar os tempos e números de iterações consumidos ao resolver sistemas lineares esparsos de grande porte. Para tanto, serão apresentados 8 tabelas e 4 gráficos resultantes dos experimentos para realizar o comparativo.

3.1 Hardware utilizado

Os testes nesta seção foram executados em uma máquina com as seguintes configurações:

Ubuntu 16.04 64-bit
Intel Core i7-3770 CPU @ 3.40GHz x 4
8GB de memória

3.2 Matrizes utilizadas

Na tabela 1 estão descritas as matrizes esparsas utilizadas nos experimentos desta seção.

Matriz	Área de Aplicação	n	nnz
rail_5177	Transferência de Calor	5177	35185
aft01	Problema de Acústica	8205	125567
FEM_3D_thermal1	Problema Térmico	17880	430740
Dubcova2	Problema 2D/3D	65025	1030225

Tabela 1: Matrizes utilizadas nos experimentos

3.3 Experimentos

Para cada uma das matrizes apresentadas na tabela 1, foi resolvido o sistema linear utilizando o método GMRES com e sem preconditionamento ILU(p), e com e sem ordenamento RCM. Serão apresentadas a seguir duas tabelas para cada matriz, relacionando o número na base de Krylov k , o parâmetro p do método ILU, a quantidade de iterações e o tempo computacional gasto para encontrar a solução através do método GMRES, com as considerações sobre reordenamento e preconditionamento.

Para uma melhor compreensão do estudo da convergência, para cada matriz da tabela 1 foram selecionados os valores de k e p que resultaram em um menor tempo de processamento utilizando o método GMRES. E com estes, foram elaborados gráficos do número de iterações em função do logaritmo do resíduo da solução ($iter \times \log(||r||)$). Nos gráficos, as iterações vão na sequência 1, 10, 20, 30, ..., 100.

Os testes foram executados com uma tolerância de $tol = 10^{-6}$. Este valor foi escolhido empiricamente pois o método converge em um tempo viável e a solução é suficientemente boa. Além disto, o número máximo de iterações do GMRES é $max = 1000$. Quando um caso de teste não convergir e atingir o máximo, será indicado max em “Iterações” e ∞ em “Tempo”.

3.3.1 rail_5177

A matriz rail_5177 é a de menor ordem ($n = 5177$), porém é a matriz que menos convergiu nos experimentos, de acordo com as tabelas 2 e 3. A solução do sistema só foi alcançada dentro do limite utilizando o GMRES com ILU(p).

O parâmetro do ILU(p) que melhor condicionou a matriz quando sem ordenamento RCM foi $p = 4$. E com o RCM, $p = 3$.

	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
k	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>max</i>	∞	<i>max</i>	∞	<i>max</i>	∞
50	<i>max</i>	∞	<i>max</i>	∞	549	1.158005
100	<i>max</i>	∞	<i>max</i>	∞	68	0.171907

Tabela 2: Matriz rail_5177 - Método GMRES sem ordenamento RCM

	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
k	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>max</i>	∞	<i>max</i>	∞	<i>max</i>	∞
50	<i>max</i>	∞	<i>max</i>	∞	2600	3.934766
100	<i>max</i>	∞	<i>max</i>	∞	76	0.182442

Tabela 3: Matriz rail_5177 - Método GMRES com ordenamento RCM

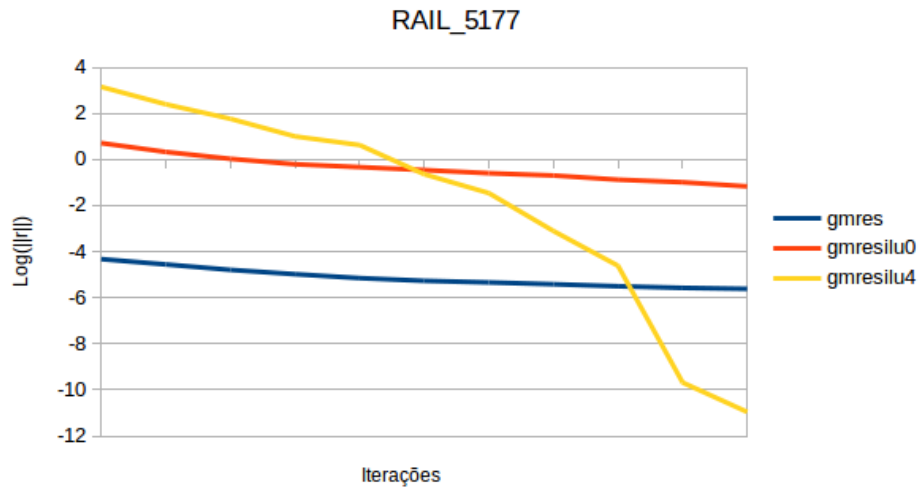


Figura 1: Matriz rail_5177 - Estudo da convergência do método GMRES

3.3.2 aft01

aft01 é uma matriz oriunda de problemas de acústica, de ordem $n = 8205$ e $nnz = 125567$ valores não nulos. Esta matriz também se mostrou um tanto complexa, assim como a rail_5177, com poucos casos de convergência. O diferencial desta matriz é que o GMRES sem preconditionamento gera resul-

tados incorretos, convergindo rapidamente mas apresentando uma solução errônea. Nestes casos, foi marcado *erro* nas tabelas 4 e 5.

O melhor valor de p do ILU encontrado, tanto com e sem ordenamento RCM, foi tal que $p = 10$.

k	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>erro</i>	—	<i>max</i>	∞	12	0.054555
50	<i>erro</i>	—	<i>max</i>	∞	12	0.054520
100	<i>erro</i>	—	85	0.313931	12	0.054545

Tabela 4: Matriz aft01 - Método GMRES sem ordenamento RCM

k	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>erro</i>	—	<i>max</i>	∞	13	0.051558
50	<i>erro</i>	—	<i>max</i>	∞	13	0.050558
100	<i>erro</i>	—	92	0.338417	13	0.050448

Tabela 5: Matriz aft01 - Método GMRES com ordenamento RCM

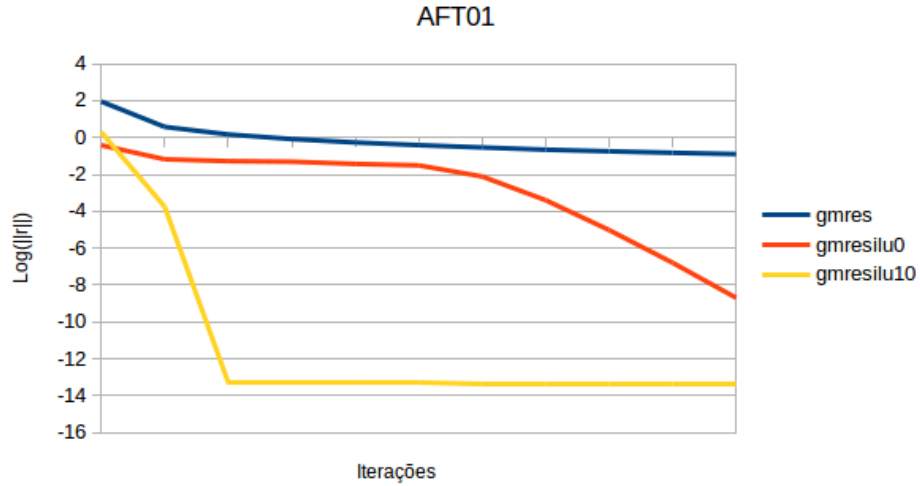


Figura 2: Matriz aft01 - Estudo da convergência do método GMRES

3.3.3 FEM_3D_thermal1

A matriz FEM_3D_thermal1, de ordem $n = 17880$ e $nnz = 430740$ mostrou-se uma matriz de fácil resolução, apesar de ordem elevada. Esta não foi capaz de convergir apenas quando utilizando GMRES sem condicionamento e $k < 100$, de acordo com as tabelas 6 e 7.

Melhor valor para ILU(p): $p = 1$

	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
k	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>max</i>	∞	7	0.039565	5	0.056208
50	<i>max</i>	∞	7	0.043184	5	0.042400
100	188	1.446595	7	0.039589	5	0.046340

Tabela 6: Matriz FEM_3D_thermal1 - Método GMRES sem ordenamento RCM

	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
k	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>max</i>	∞	9	0.047200	6	0.050042
50	<i>max</i>	∞	9	0.046716	6	0.050021
100	188	1.394231	9	0.048806	6	0.050120

Tabela 7: Matriz FEM_3D_thermal1 - Método GMRES com ordenamento RCM

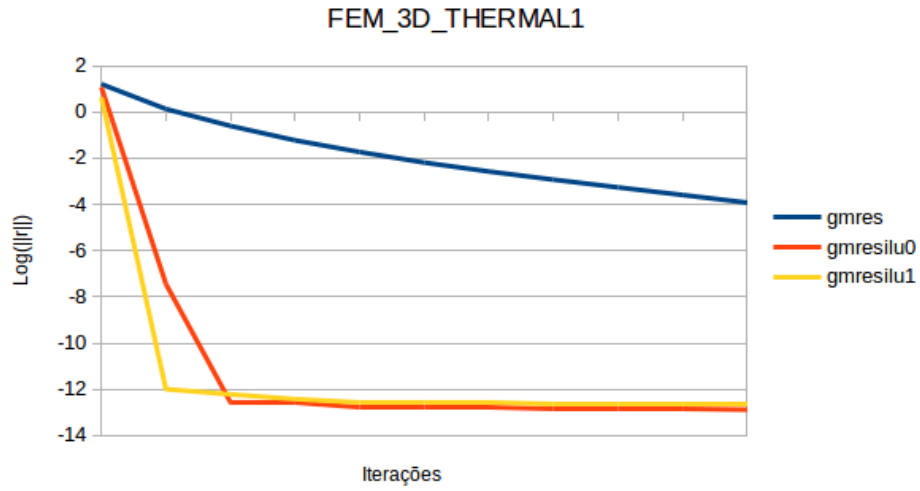


Figura 3: Matriz FEM_3D_thermal1 - Estudo da convergência do método GMRES

3.3.4 Dubcova2

Dubcova2 é a maior matriz utilizada nestes experimentos, com ordem $n = 65025$ e $nnz = 1030225$ valores não nulos. E ainda assim, esta converge mais facilmente que as menores matrizes estudadas até então. Veja nas tabelas 8 e 9 os testes realizados nesta matriz.

Para o condicionamento ILU(p), foram encontrados $p = 1$ como o melhor caso não seja realizado o ordenamento RCM e $p = 5$ caso contrário.

k	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>max</i>	∞	<i>max</i>	∞	880	22.371890
50	999	15.694150	141	2.765995	37	1.005747
100	252	5.957986	74	1.845755	37	1.015742

Tabela 8: Matriz Dubcova2 - Método GMRES sem ordenamento RCM

k	GMRES (s/ Precond.)		GMRES + ILU(0)		GMRES + ILU($p > 0$)	
	Iterações	Tempo	Iterações	Tempo	Iter.	Tempo
20	<i>max</i>	∞	198	2.913908	15	0.356196
50	999	16.022811	49	1.027727	15	0.360827
100	252	6.092857	49	0.995358	15	0.363706

Tabela 9: Matriz Dubcova2 - Método GMRES com ordenamento RCM

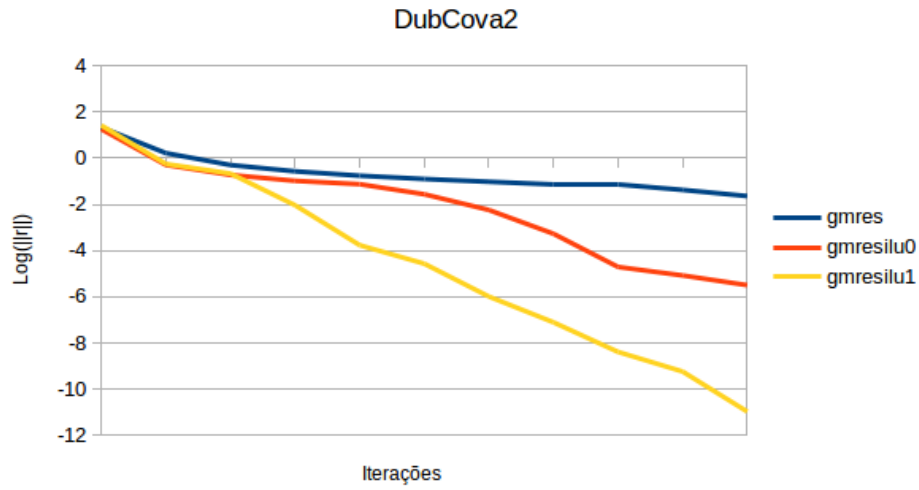


Figura 4: Matriz Dubcova2 - Estudo da convergência do método GMRES

4 Conclusão

Da série de experimentos realizados na seção 3, podemos avaliar o método não estacionário GMRES e o uso de reordenamento e preconditionadores no mesmo.

É notável perceber que o GMRES possui bons resultados para sistemas esparsos de grande porte, uma vez que foi pensado para isto. Mas vemos que, simplesmente utilizado, o método pode sofrer de uma convergência lenta, tornando-se inviável, ou até mesmo produzir resultados incorretos. Este último pode ser observado nos testes da matriz aft01 (seção 3.3.2). Quanto a velocidade de convergência, em todas as matrizes houve ao menos um caso em que o GMRES puro não obteve a solução no limite de iterações. Em destaque, para a matriz rail_5177, que não convergiu para qualquer teste (seção 3.3.1).

Como esperado, o GMRES puro não garante, em muitos casos, bons

resultados, principalmente em questão de tempo. Agora, quando adicionado a ele o uso do preconditionador $ILU(p)$, observamos resultados muito melhores. Por exemplo, no experimento da matriz Dubcova2 (seção 3.3.4), na tabela 8, e $k = 50$, ocorre um ganho muito grande de tempo e iterações. De 999 para 141 iterações e de aproximadamente 15 segundos para 3. Uma redução significativa devido ao preconditionamento da matriz.

Ainda sobre o preconditionamento $ILU(p)$, vemos que o parâmetro p é bastante relevante no método. Isto pode ser ilustrado nos testes da matriz rail_5177 (tabelas 2 e 3), onde para $p = 0$, não houve ganho algum se comparado com o GMRES simples, pois ambos não convergiram. Em contrapartida, para $p = 4$, foi alcançado uma resposta em tempo extremamente viável. E se observados os outros experimentos, vemos que o melhor valor de p não é o maior nem possui um padrão entre os casos. O melhor valor depende da matriz proposta, sendo definido, portanto, empiricamente.

Quanto ao ordenamento RCM, podemos observar sua influência positiva claramente no experimento da matriz Dubcova2 (3.3.4). Sem ordenamento, para $k = 20$, o método converge apenas quando aplicado $ILU(1)$, e ainda converge lentamente, com 880 iterações em 22 segundos. Agora, quando aplicado o RCM, este tempo cai para 0.35 segundos em apenas 15 iterações, utilizando $p = 5$, o novo melhor parâmetro do ILU . Além disto, também passa a convergir para $ILU(0)$, o que não era verdade quando sem o ordenamento.

Mas o processo RCM nem sempre surte efeitos positivos, mantendo os tempos e iteração praticamente iguais. E em alguns poucos casos, aparentemente pode até piorar a convergência do GMRES. Assim como para o valor de p , a efetividade do ordenamento RCM não é fácil de ser prevista. Logo, deve-se determinar o seu uso de acordo com a matriz a ser resolvida do sistema.

Com estas análises, vemos que resolver sistemas lineares esparsos de grande porte não é trivial. Devem ser considerados todos estes métodos - armazenamento, reordenamento e pré-condicionamento da matriz e o método de resolução do sistema - para que seja viável resolver muitos problemas que surgem de aplicações reais.