



UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS
MATEMÁTICAS
E DE COMPUTAÇÃO CIENTÍFICA



Relatório de Trabalho

Diferenças entre Métodos Diretos e Métodos Iterativos para resolver sistemas lineares:
implementação, complexidade, tempo de execução e detalhes adicionais

Disciplina: Cálculo Numérico- SME0205

Aluna: Anne Kéllen de Nazaré dos Reis Dias N^o USP 10552210

Aluno: Luiz Francisco Franca de Farias N^o USP 13686542

Aluno: Luis Roberto Piva N^o USP 13687727

Professor: Antonio Castelo Filho

São Carlos- SP
Maio 2023

Conteúdo

1	Introdução	3
2	Metodologia	3
2.1	Manipulação do grafo e da componente conexa:	3
2.2	Construção das matrizes penalidades e Laplaciana:	4
2.2.1	Matriz Penalidade	4
2.2.2	Matriz Laplaciana	4
2.2.3	Relação entre a Matriz Penalidade e Laplaciana	5
2.2.4	Aplicação das técnicas propostas	5
2.3	Comparação de tempo de execução dos métodos diretos e iterativos:	6
2.3.1	Método Diretos:	6
2.3.2	Método Iterativos:	7
3	Resultados	9
4	Conclusão	11
5	Referências	12

1 Introdução

O objetivo do trabalho proposto é analisar as diferenças entre os métodos diretos e iterativos para resolver sistemas lineares aplicados a um problema específico. Utilizando a implementação em MATLAB, podemos compreender como aplicar esses métodos para resolver um problema real, avaliar o tempo de execução e a complexidade de cada método, e tirar conclusões sobre a eficiência e limitações de cada um.

Como motivação para este projeto, consideramos a cidade de Manhattan, em Nova York, EUA, e escolhemos um problema amplo, como a temperatura, para aplicar na cidade. Optamos pela temperatura devido à disponibilidade de dados e às leis naturais que permitem uma interpolação mais realista.

Nossa abordagem consiste em considerar cada rua de Manhattan como uma aresta que conecta duas esquinas, formando um grafo. A partir desse grafo, identificamos a maior componente conexa e construímos a matriz Laplaciana do grafo de ruas usando o operador de Laplace-Beltrami. Em seguida, aplicamos os métodos diretos e iterativos para resolver o sistema linear $(L + P)x = Pb$, utilizando a técnica da matriz de penalidades. Dessa forma, podemos obter uma interpolação dos dados de temperatura nos vértices próximos, com base nos dados disponíveis em alguns vértices e analisar o tempo de execução de cada método empregado.

2 Metodologia

O problema proposto foi abordado utilizando uma implementação em MATLAB. Este, foi, inicialmente, dividido em três etapas: Manipulação do grafo e da maior componente conexa; a construção das matrizes de penalidades e Laplaciana, e, por fim; a comparação de tempo de execução dos métodos de solução do sistema, tanto diretos como iterativos.

2.1 Manipulação do grafo e da componente conexa:

Definição:

Um grafo consiste em um conjunto finito e não vazio de objetos chamados vértices (também chamados de nós), juntamente com um conjunto de pares não ordenados de vértices, tais elementos são chamados de arestas.

Iniciamos o trabalho utilizando a cidade de Manhattan, NY, para criar um grafo. Cada rua da cidade foi representada como uma aresta que conecta duas esquinas, que são os vértices do grafo. Recebemos um arquivo com as coordenadas dos vértices (um total de 8837) e outro arquivo que especifica as conexões entre os vértices, totalizando 13713 conexões.

Definição:

As componentes conexas de um grafo são seus pedaços que são isoladamente conexos. Assim sendo, as componentes conexas de um dado são subgrafos dele próprio que são conexos.

Para encontrar a maior componente conexa do grafo conforme exigido pelo enunciado, utilizamos os dados fornecidos. Primeiramente, construímos a matriz de adjacência do grafo. A matriz de adjacência é uma representação comum de um grafo, onde para o



Figura 1: Manhattan-NY

nosso caso com 8837 vértices, criamos uma matriz quadrada $\mathcal{M}(n \times n)$ com $n = 8837$. A matriz adjacência $A = [a_{ij}]$ é preenchida de acordo com a presença de conexões entre os vértices. Se dois vértices, x e y , são adjacentes, atribuímos o valor 1 às coordenadas (x, y) e (y, x) da matriz adjacência. Caso contrário, mantemos o valor 0 para indicar a ausência de conexão entre os vértices. Com base nessa matriz, identificamos a maior componente conexa do grafo para prosseguir com a análise.

2.2 Construção das matrizes penalidades e Laplaciana:

Após obter a maior componente conexa do grafo de Manhattan, avançamos para a segunda etapa do trabalho, utilizando as técnicas das matrizes de penalidades e de Laplace. Primeiro, vamos compreender a definição e o propósito destas, bem como a diferença que elas apresentam ao serem utilizadas em um sistema linear.

2.2.1 Matriz Penalidade

A matriz de penalidade é utilizada na resolução de sistemas lineares como uma técnica para lidar com problemas mal-condicionados ou instáveis, pois pequenas perturbações nos valores dos coeficientes podem levar a grandes variações nas soluções encontradas. A matriz de penalidade é introduzida, então, para regularizar o sistema, adicionando um termo de penalização que ajuda a estabilizar a solução.

Portanto, a matriz de penalidade é uma abordagem utilizada para melhorar a estabilidade numérica e a confiabilidade das soluções em sistemas lineares mal-condicionados. Além disso, a matriz de penalidade é uma matriz que é frequentemente utilizada em conjunto com a matriz Laplaciana na montagem de sistemas lineares em problemas de elementos finitos.

2.2.2 Matriz Laplaciana

A matriz Laplaciana utilizada na resolução de sistemas lineares é uma matriz quadrada definida pela seguinte fórmula:

$$L = D - A \quad (1)$$

onde:

- L é a matriz Laplaciana;
- D é uma matriz diagonal que contém os graus dos vértices do grafo associado ao sistema linear;
- A é a matriz de adjacência do grafo, que representa as conexões entre os vértices.

A matriz Laplaciana obtida dessa forma possui algumas propriedades úteis, como ser simétrica e não negativa. Essas propriedades são exploradas em métodos eficientes de resolução de sistemas lineares, como o método dos gradientes conjugados. Além disso, a matriz laplaciana é uma matriz especial que descreve a relação entre os nós ou pontos em uma rede. Ela é formada pela diferença entre o grau de cada nó (o número de arestas conectadas a ele) e as conexões entre os nós.

A matriz de penalidade é uma ferramenta complementar à matriz Laplaciana e é usada para incorporar restrições e condições específicas do problema em sistemas lineares formulados pelo método dos elementos finitos. Como faremos aqui

2.2.3 Relação entre a Matriz Penalidade e Laplaciana

A matriz penalidade e a matriz laplaciana são conceitos diferentes, mas podem estar relacionados quando se trata da resolução de sistemas lineares em determinados contextos.

A relação entre a matriz penalidade e a matriz laplaciana surge quando a matriz laplaciana é utilizada como a matriz original na resolução de um sistema linear. Nesse caso, a adição da matriz de penalidade à matriz laplaciana pode ajudar a estabilizar a solução. Essa abordagem é frequentemente utilizada em métodos de regularização, como o método de Laplace-Beltrami que estamos utilizando, onde a matriz laplaciana é modificada com uma matriz de penalidade para melhorar a estabilidade e a precisão das soluções.

Por sua vez, em resumo, o método de Laplace-Beltrami, para resolução de sistemas lineares em grafos utiliza a matriz Laplaciana do grafo como uma representação da estrutura do grafo e suas propriedades na formulação e solução do sistema linear associado. Isso permite levar em consideração a topologia do grafo e obter soluções mais adequadas para problemas em grafos.

2.2.4 Aplicação das técnicas propostas

Definido o método a ser utilizado, explicaremos nossas ações conforme solicitado no enunciado do trabalho.

Nesta etapa, selecionamos $k \ll n$ (sendo n o número de vértices na maior componente do grafo e k um valor significativamente menor que n). Aleatoriamente, escolhemos 10% dos vértices da maior componente para atribuir valores no intervalo $(0, 10]$ visando uma melhor interpolação. Ao mesmo tempo, criamos a matriz de penalidades P , que é uma matriz diagonal em que a entrada $P_{jj} = \alpha$ se j corresponde ao índice de algum dos vértices selecionados anteriormente, e $P_{ii} = 0$ caso contrário. O valor escolhido para α foi $\alpha = 1.0e7$, conforme o enunciado propunha. Também criamos um vetor com os dados de temperatura inicial contendo os valores atribuídos a cada vértice em seus respectivos índices.

Na etapa seguinte desta segunda parte, procedemos à construção da matriz Laplaciana referente a maior componente. A matriz adjacência da maior componente foi construída

na primeira parte deste trabalho. A matriz de graus, por sua vez, pode ser facilmente obtida a partir desta matriz. Para isso, basta somar todos os valores 1 em cada linha do vetor que representa cada vértice da componente. Em outras palavras, somamos as ocorrências de conexões de um vértice com outros. Com a matriz de graus da componente calculada, a matriz laplaciana pode ser obtida de forma direta.

2.3 Comparação de tempo de execução dos métodos diretos e iterativos:

Combinando todas as informações obtidas nas últimas duas partes do trabalho para produzir e resolver o sistema linear $(L + P)x = Pb$ e obter a interpolação desejada. Vamos definir A como a matriz resultante da soma das matrizes Laplaciana e Penalidade, e b como o vetor inicial obtido multiplicando a matriz de penalidade pelo vetor independente encontrado na etapa anterior. Dessa forma, temos um sistema linear montado e pronto para ser solucionado utilizando métodos diretos e iterativos. Começaremos pelos métodos diretos, seguidos pelos métodos iterativos.

2.3.1 Método Diretos:

Métodos diretos para solução de sistemas lineares são algoritmos que fornecem a solução exata para um sistema de equações lineares. Esses métodos envolvem operações matemáticas diretas sobre os coeficientes do sistema para encontrar a solução. Podem ser muito eficazes para sistemas de tamanho moderado, mas podem se tornar computacionalmente caros para sistemas grandes ou esparsos, onde métodos iterativos podem ser mais eficientes. Alguns exemplos comuns de métodos diretos incluem:

- Métodos da Decomposição LU

A decomposição LU é um método que consiste em decompor a matriz A em dois fatores: uma matriz triangular inferior L e uma matriz triangular superior U .

Definição (matriz triangular inferior):

Uma matriz $L \in \mathcal{M}(n, n)$ é dita triangular inferior se $\ell_{ij} = 0, \forall j > i$.

Definição (matriz triangular superior):

Uma matriz $U \in \mathcal{M}(n, n)$ é dita triangular superior se $u_{ij} = 0, \forall i > j$.

O objetivo da decomposição LU é simplificar a resolução do sistema linear $Ax = b$, dividindo o problema em duas etapas. Consideremos, então, o sistema:

$$Ax = b \tag{2}$$

Se $A = LU$, pelo método da decomposição LU , então $(LU)x = b \Leftrightarrow L(Ux) = b$.

Vamos definir $y = Ux$ como um vetor intermediário e resolver o sistema $Ly = b$ utilizando o método das substituições progressivas. Em seguida, resolveremos o sistema $Ux = y$ utilizando o método das substituições regressivas. (É importante destacar que o método da barra do MATLAB utiliza a decomposição LU como parte do seu processo de resolução.)

- Método da Decomposição de Cholesky

Definição (matriz simétrica positiva definida):
Uma matriz simétrica $A \in \mathcal{M}(n, n)$ ($A = A^T$) é dita simétrica positiva definida (SPD), se $v^* Av > 0$, para todo vetor não-nulo $v \in \mathbb{R}^n$.

Podemos simplificar o método de decomposição LU quando a matriz é simétrica e positiva definida (SPD). Para uma matriz A SPD, desejamos obter a decomposição única $A = H^* H^T$, onde H é uma matriz triangular inferior com elementos positivos na diagonal. Com a matriz H em mãos, aplicamos as funções progressivas e regressivas, semelhantes ao método LU e achamos a resolução do sistema linear.

2.3.2 Método Iterativos:

Por outro lado, métodos iterativos para solução de sistemas lineares são algoritmos que aproximam a solução de um sistema de equações lineares por meio de iterações sucessivas. Em vez de fornecer uma solução exata, esses métodos convergem gradualmente para a solução, refinando a estimativa a cada iteração. Tais métodos geralmente começam com uma estimativa inicial da solução e, em cada iteração, atualizam essa estimativa com base nos resíduos do sistema. O processo continua até que uma condição de convergência seja alcançada, como uma tolerância pré-definida para o erro.

Os métodos iterativos podem ser mais eficientes do que os métodos diretos para sistemas grandes ou esparsos, pois geralmente requerem menos operações aritméticas. No entanto, a convergência nem sempre é garantida e pode ser necessário ajustar parâmetros ou utilizar estratégias avançadas para obter resultados precisos.

Uma possível estratégia para melhorar a eficiência dos métodos iterativos é ajustar os critérios de parada utilizados, com base em uma precisão desejada ou em um número máximo de iterações. Ao reduzir o critério de precisão, é possível obter soluções mais precisas, mas ao custo de um maior número de iterações. Por outro lado, aumentar o critério de precisão pode diminuir o número de iterações, porém, resultar em soluções menos precisas. Encontrar um equilíbrio adequado é essencial.

Alguns exemplos comuns de métodos iterativos incluem:

- Método de Gauss-Jacobi:

O método de Gauss-Jacobi é aplicável a sistemas de equações lineares em que a matriz de coeficientes é diagonalmente dominante ou estritamente diagonalmente dominante. O método de Gauss-Jacobi é baseado na ideia de iterativamente atualizar os valores das incógnitas até que a solução convirja para um valor satisfatório.

Para este método e para o próximo, transformamos o sistema linear $Ax = b$ em um sistema equivalente da forma $x = Cx + g$, onde $C \in \mathcal{M}(n, n)$ e $g \in \mathbb{R}^n$. A forma como esse método transforma $Ax = b$ em $x = Cx + g$ é feita isolando cada coordenada x_i do vetor x na i -ésima equação do sistema. O processo ocorre da seguinte forma: seja D uma matriz diagonal formada pela diagonal de A , queremos obter $x^{(k+1)} = Cx^{(k)} + g$. Assim,

$$Ax = b \Leftrightarrow (A - D + D)x = b \Leftrightarrow (A - D)x + Dx = b$$

Dessa forma,

$$(A - D)x^{(k)} + Dx^{(k+1)} = b \Leftrightarrow Dx^{(k+1)} = (D - A)x^{(k)} + b$$

Portanto,

$$x^{(k+1)} = (I - D^{-1}A)x^{(k)} + D^{-1}b \quad (3)$$

Se fizermos $C = (I - D^{-1}A)$ e $g = D^{-1}b$ então temos o sistema desejado.

O processo iterativo continua até que a solução convirja para um valor aceitável, geralmente quando a diferença entre as iterações sucessivas se torna pequena o suficiente.

- Método de Gauss-Seidel:

O método de Gauss-Seidel é considerado uma aceleração do Método de Gauss-Jacobi, pois utiliza uma atualização das incógnitas de forma mais direta. Queremos então obter $x = Cx + g$ a partir de $Ax = b$, como anteriormente em Gauss-Jacobi. Considere $A = L + R$, em que L é a matriz triangular inferior de A e R é a matriz triangular superior de A sem diagonal. Assim,

$$Ax = b \Leftrightarrow (L + R)x = b \Leftrightarrow Lx + Rx = b$$

Dessa forma,

$$Lx^{(k+1)} + Rx^{(k)} = b \Leftrightarrow Lx^{(k+1)} = -Rx^{(k)} + b$$

Portanto,

$$x^{(k+1)} = (-L^{-1}R)x^{(k)} + L^{-1}b \quad (4)$$

Se fizermos, então, $C = (-L^{-1}R)$ e $g = L^{-1}b$, temos o sistema desejado.

- Método dos Gradientes Conjugados:

O método dos gradientes conjugados é um método iterativo utilizado para resolver sistemas lineares de grande porte e simétricos. Este é um método iterativo, que parte da mesma ideia do Método dos Gradiente (minimizar uma função quadrática). No entanto, há um limitante para o número de iterações necessárias para que o Método dos Gradientes Conjugados convirja à solução do sistema linear.

Assim como no Método dos Gradientes, trocaremos o problema de encontrar uma solução para o sistema $Ax = b$ pelo problema de encontrar um minimizador de $\frac{1}{2}x^T Ax - b^T x$, com A simétrica e definida positiva.

Dada a função $f(x) = \frac{1}{2}x^T Ax - b^T x$, temos que:

$$\nabla f(x) = Ax - b \quad (5)$$

$$\nabla^2 f(x) = A \quad (6)$$

Encontrar a solução do sistema linear $Ax = b$ é equivalente a encontrar o ponto x que satisfaz $\nabla f(x) = Ax - b = 0$, ou seja, o minimizador da função f . O método dos gradientes conjugados tem a vantagem de convergir para a solução exata em um número finito de iterações quando aplicado a sistemas lineares com matrizes simétricas e positivas definidas.

3 Resultados

Começamos utilizando Manhattan, NY, como referência para criar um grafo. Recebemos um arquivo com as coordenadas dos vértices (um total de 8837) e outro com as conexões entre eles, totalizando 13713. Com base nesses dados, construímos a matriz de adjacência e, a partir dela, geramos o seguinte grafo:

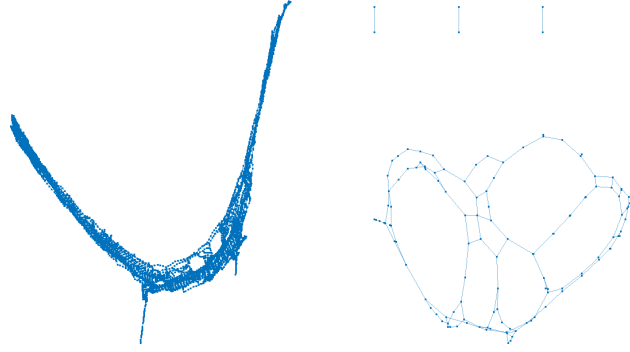


Figura 2: Grafo de ruas de Manhattan gerado pelo MATLAB

Ainda por meio da matriz adjacência do grafo e a função `Split_Edges.m` fornecida no enunciado do trabalho, foram obtidos os seguintes resultados: o número total de componentes conexas encontradas no grafo é 5; o número de vértices em cada componente que estão representado pelo vetor $[8708, 123, 2, 2, 2]$; e uma matriz 5×8708 que contém todos os vértices presentes em cada uma das componente conexa. Além disso, utilizando esses resultados e a matriz de adjacência da maior componente, conseguimos gerar uma representação visual da maior componente conexa:

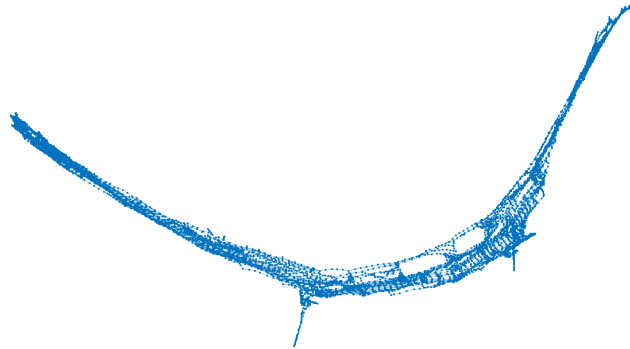


Figura 3: Maior componente conexa do grafo de Manhattan

Seguindo o cronograma do trabalho, avançamos para a construção da matriz de penalidades e de um vetor solução incompleto. Para esse propósito, selecionamos aleatoriamente 80 vértices do conjunto de arestas da maior componente conexa e atribuímos valores randômicos dentro do intervalo de $(0, 10]$. Assim, quando um vértice v_1 foi escolhido aleatoriamente, atribuímos o valor de $\alpha = 1.0e7$ na diagonal da matriz de penalidades $P(8708 \times 8708)$, especificamente no índice (v_1, v_1) . Simultaneamente, construímos um vetor solução inicial. Como o vértice v_1 foi escolhido, atribuímos um valor de temperatura

a ele dentro do intervalo de $(0, 10]$. Portanto, no nosso vetor solução inicial $b^{(1)}(8708 \times 1)$, atribuímos esse valor ao índice $(v_1, 1)$.

Após a criação dessas duas matrizes, prosseguimos para a construção da matriz de graus $Dg(8708 \times 8708)$, que armazena os dados dos graus de cada vértice. Em seguida, com base na matriz de adjacência da maior componente e na matriz de graus recém-criada, procedemos à formação da matriz laplaciana $L(8708 \times 8708)$.

Assim, podemos representar o sistema linear a seguir, utilizando os dados encontrados até o momento, para resolvê-lo por meio de métodos diretos e iterativos:

$$Ax = b \quad (7)$$

onde:

- $A = L + P$;
- $b = P * b^{(1)}$.

Os métodos diretos e iterativos são duas abordagens diferentes para resolver sistemas lineares.

Todos os métodos empregados, sejam eles diretos ou iterativos, convergiram para uma solução satisfatória e consistente do problema em questão. Além disso, os resultados obtidos demonstraram uma notável similaridade entre si em relação aos valores, exibindo apenas uma pequena margem de discrepância. Por outro lado, as diferenças de tempo de execução entre eles dependem de vários fatores, incluindo o tamanho do sistema, a esparsidade da matriz e a precisão desejada.

A seguir, são apresentados os tempos de execução obtidos para cada método utilizado na resolução do sistema (7):

Método	tempo
Decomposição LU	3,8625
Decomposição LU com barra	3,1599
Cholesky	3,2326
Cholesky com barra	1,6573
Jacobi	899,5014
Gauss-Seidel	460,6127
Gradientes conjugados	146,7786

Tabela 1: Valores de tempo de execução dos métodos diretos e iterativos em segundos

Era de se esperar que os métodos diretos fossem mais rápidos na solução desse problema, como demonstrado na tabela, onde os tempos de execução dos métodos diretos diferem significativamente dos tempos dos métodos iterativos.

Para comparar os resultados e tempos de execução dos métodos diretos, incluímos no trabalho o método da barra do próprio MATLAB, que obteve um tempo de execução de 2,058636 segundos. É importante ressaltar que o método da barra do MATLAB utiliza a decomposição LU como parte de seu processo de resolução.

Também calculamos os tempos de execução para os métodos de decomposição LU e decomposição de Cholesky, a fim de comparar a diferença de tempo ao usar as funções de substituições progressivas e regressivas (primeiro teste) para resolver os subsistemas resultantes de ambas as decomposições, e ao usar a função da barra nativa do MATLAB (segundo teste).

Para o método de decomposição LU , utilizamos a função "lu" nativa do MATLAB para realizar a fatoração de A em L e U para usarmos nos dois testes mencionados

anteriormente. No primeiro teste, o tempo de execução foi de 3,862542 segundos. Já no segundo teste, o tempo de execução foi de 3,159898 segundos.

Para o método de decomposição de Cholesky, utilizamos a função "chol" do MATLAB para obter a matriz H , tal que $A = H * H^T$, e realizamos os dois testes mencionados. No primeiro teste, o tempo de execução foi de 3,232626 segundos. Já no segundo teste, obtivemos um tempo de execução de 1,657287 segundos.

Os métodos iterativos apresentam diferenças significativas nos tempos de execução, ao contrário dos métodos diretos. Por exemplo, os tempos de execução registrados foram: 899,5014 segundos para o método de Gauss-Jacobi; 460,6127 segundos para o método de Gauss-Seidel e,; 146,7786 segundos para o método dos Gradientes Conjugados.

4 Conclusão

Com base nos resultados apresentados na seção anterior, é possível obter algumas conclusões acerca das diferenças entre Métodos Diretos e Métodos Iterativos na resolução de sistemas lineares. Essas conclusões consideram fatores como implementação, complexidade, tempo de execução e detalhes adicionais de cada método implementado.

A eficiência relativa dos métodos pode ser influenciada por diversos fatores, tais como a estrutura da matriz do sistema linear, a taxa de convergência do método, o número de iterações necessárias para atingir a solução desejada e a implementação específica em MATLAB.

Vamos considerar os resultados de tempo de execução obtidos para resolver um sistema linear usando diferentes métodos diretos. Na tabela, observamos os seguintes tempos: 2,0586 segundos para o método da barra do MATLAB; 3,8625 segundos e 3,2326 segundos para o primeiro teste com o método da decomposição LU e e com a descomposição de Cholesky, respectivamente; e 3,1599 segundos e 1,6573 segundos para o segundo teste com o método da decomposição LU e e com a descomposição de Cholesky, respectivamente.

Fiica evidente, pois, que o Método da Decomposição de Cholesky foi o mais eficiente e rápido dentre os métodos diretos utilizados. Essa conclusão baseia-se na consideração de que o tempo de execução é um critério importante para avaliar a eficiência dos métodos diretos. No entanto, há outras razões que tornam esse método particularmente vantajoso para nosso trabalho:

- A matriz de coeficientes considerada é simétrica e positiva definida (SPD);
- O método requer menos operações aritméticas em comparação com outros métodos diretos, como a decomposição LU , pois envolve apenas metade do número de operações em relação à este último;
- Possui uma complexidade computacional menor do que outros métodos diretos implementados;
- Sua implementação é menos sensível a erros de arredondamento em comparação com a decomposição LU , pois a decomposição de Cholesky envolve raízes quadradas.

Vamos agora analisar os resultados de tempo de execução dos métodos iterativos. Na tabela, podemos observar os seguintes tempos: 899,5014 segundos para o Método de Gauss-Jacobi, 460,6127 segundos para o Método de Gauss-Seidel e 146,7786 segundos para o Método dos Gradientes Conjugados. O tempo de execução do método de Jacobi

está relacionado a um número fixo de iterações que convergem para a solução. Cada iteração envolve operações de multiplicação de matriz e adição/subtração de vetores. Os resultados obtidos com o Método de Gauss-Seidel, por sua vez, eram esperados, uma vez que esse método foi desenvolvido como uma "melhoria" em relação ao Método de Gauss-Jacobi.

Com base nos tempos de execução observados, fica claro que o Método dos Gradientes Conjugados se destacou como o mais rápido entre os métodos iterativos utilizados. Essa conclusão também leva em consideração o fato de que o tempo de execução é um critério importante para avaliar a eficiência dos métodos diretos.

Além disso, o Método dos Gradientes Conjugados possui a vantagem de poder convergir para a solução exata em um número de iterações igual ao número de variáveis do sistema. Portanto, a variação nos tempos de execução está diretamente relacionada à convergência dos métodos e ao número de iterações necessárias para atingir a solução desejada. Isso significa que, em geral, o Método dos Gradientes Conjugados requer um número menor de iterações em comparação com outros métodos iterativos, o que resulta em um tempo de execução mais rápido.

Há ainda outras razões que tornam o Método dos Gradientes Conjugados particularmente eficiente para essa situação:

- O método possui uma propriedade chamada "conjugação de direções", que permite que ele encontre a solução em um número relativamente pequeno de iterações;
- O método utiliza principalmente produtos matriciais para atualizar a solução iterativamente. Essa abordagem pode ser computacionalmente eficiente, especialmente quando a matriz do sistema tem uma estrutura especial, como ser simétrica e positiva definida, como a nossa;
- Sua versatilidade torna-o uma escolha eficiente em várias áreas de aplicação.

Em resumo, a escolha entre Métodos Diretos e Métodos Iterativos depende das características do sistema linear, da precisão desejada, da complexidade computacional tolerável e dos recursos disponíveis. Cada método possui vantagens e desvantagens, e é importante considerar cuidadosamente esses aspectos ao selecionar a abordagem mais adequada para resolver sistemas lineares em um determinado contexto.

5 Referências

PAIVA, Afonso. Sistema Lineares: Métodos Diretos. Disponível em: https://ae4.tidia-ae.usp.br/access/content/group/ff1f4693-e871-413f-ac2f-6c4ab44f10b5/Slides/linsis_diretos.pdf. Acesso em 20/05/2023.

PAIVA, Afonso. Sistema Lineares: Métodos Iterativos. Disponível em: https://ae4.tidia-ae.usp.br/access/content/group/ff1f4693-e871-413f-ac2f-6c4ab44f10b5/Slides/linsis_iterativo.pdf. Acesso em 20/05/2023.

ANDRETTA, Marina. TOLEDO, Franklina. Resolução de sistemas de equações lineares: Métodos dos Gradientes Conjugados. Disponível em: <https://sites.icmc.usp.br/andretta/ensino/aulas/sme0100-2-12/aula6-gradconj.pdf>. Acesso em 20/05/2023.

SCHEWCHUK, Jonathan Richard. Classroom figures for the Conjugate Gradient Method Without the Agonizing Pain. Disponível em: <https://ae4.tidia-ae.usp.br/access/content/group/ff1f4693-e871-413f-ac2f-6c4ab44f10b5/Slides/painless-conjugate-gradient.pdf>. Acesso em 19/05/2023.