



GUILHERME OLIVEIRA CHAGAS

**UMA VERIFICAÇÃO DE DESEMPENHO DO
MÉTODO DOS GRADIENTES
CONJUGADOS COM REDUÇÃO DE BANDA
PELO MÉTODO CUTHILL-MCKEE
REVERSO COM VÉRTICE
PSEUDOPERIFÉRICO PELO ALGORITMO
DE GEORGE-LIU**

LAVRAS – MG

2013

GUILHERME OLIVEIRA CHAGAS

**UMA VERIFICAÇÃO DE DESEMPENHO DO MÉTODO DOS
GRADIENTES CONJUGADOS COM REDUÇÃO DE BANDA PELO
MÉTODO CUTHILL-MCKEE REVERSO COM VÉRTICE
PSEUDOPERIFÉRICO PELO ALGORITMO DE GEORGE-LIU**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da computação para a obtenção do título de Bacharel em Ciência da Computação.

Orientador

Prof. D.Sc. Sanderson L. Gonzaga de Oliveira

LAVRAS – MG

2013

GUILHERME OLIVEIRA CHAGAS

**UMA VERIFICAÇÃO DE DESEMPENHO DO MÉTODO DOS
GRADIENTES CONJUGADOS COM REDUÇÃO DE BANDA PELO
MÉTODO CUTHILL-MCKEE REVERSO COM VÉRTICE
PSEUDOPERIFÉRICO PELO ALGORITMO DE GEORGE-LIU**

Monografia apresentada ao Colegiado do Curso de
Ciência da Computação, para a obtenção do título
de Bacharel em Ciência da Computação.

APROVADA em 08 de abril de 2013.

Prof. D.Sc. Daniel Furtado Ferreira

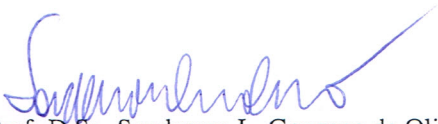
DEX/UFLA

Prof. D.Sc. Lucas Monteiro Chaves

DEX/UFLA

Prof. D.Sc. Wilian Soares Lacerda

DCC/UFLA



Prof. D.Sc. Sanderson L. Gonzaga de Oliveira
(Orientador)

LAVRAS – MG

2013

Aos meus pais, minhas inspirações para seguir em frente...

AGRADECIMENTOS

Acima de tudo, quero agradecer às pessoas que mais me apoiaram até aqui, sem eles não teria trilhado metade do caminho, ao Sr. Zaqueu e a Sra. Irene, os meus maiores incentivadores, os meus pais! Agradeço, também, aos meus familiares pelo apoio e incentivo, em especial à Vovó, Tia Dadá, Tia I, meu padrinho Dé, minha madrinha Célia e minha prima Valda. Agradeço à minha companheira Thaís, por toda amizade, carinho e amor passados durante todo esse tempo. Ao meu orientador e amigo Sanderson, por todos os ensinamentos e conselhos dados durante minha iniciação científica e a realização deste trabalho. Ao Jack e à Mentira, meus cachorros. Aos meus amigos de infância e, atualmente, segunda família, Matheus, Limão e Bozo. Aos meus grandes amigos que percorreram todo esse caminho ao meu lado, em especial ao Luiz Fernando, Igor, Bolina, Francisco e Rennan, sem eles, o caminho teria sido muito mais difícil. Aos meus amigos de laboratório, em especial à Jéssica e ao Frederico. Ainda, agradeço os apoios financeiros do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (Fapemig). Finalmente, agradeço ao meu avô e ídolo Paulo Tomé das Chagas (*in memoriam*), a pessoa mais sábia e inteligente que conheci.

RESUMO

Neste trabalho, implementou-se o método Cuthill-Mckee reverso, iniciando-se por um vértice pseudoperiférico para redução de banda de matrizes esparsas simétricas. A redução de banda consiste em realizar permutações de linhas e colunas de uma matriz inicialmente esparsa, deixando-a com uma estrutura compacta e com coeficientes não nulos próximos à diagonal principal. Analisou-se o desempenho computacional do método dos gradientes conjugados em sistemas de equações lineares modificados pelo Cuthill-Mckee reverso. Verificou-se, nos testes realizados, que, com um sistema de equações lineares alterado pelo Cuthill-Mckee reverso, houve diminuição no tempo de execução do método dos gradientes conjugados.

Palavras-Chave: Redução de banda, redução de *profile*, Cuthill-Mckee reverso, gradientes conjugados, vértice pseudoperiférico.

ABSTRACT

In this work, the reverse Cuthill-McKee method was implemented starting with a pseudoperipheral vertex to reduce the bandwidth of symmetric sparse matrices. The problem of reducing the bandwidth of a matrix consists of finding a permutation of rows and columns of a given matrix which keeps the non-zero elements in a band as close as possible to the main diagonal. Also, it was studied the computational performance of the conjugate gradients method for systems of linear equations modified by the reverse Cuthill-McKee. Clearly, the results show that the conjugate gradients method is faster after running the reverse Cuthill-McKee to reduce the bandwidth of the given linear system.

Keywords: Bandwidth and profile reduction, reverse Cuthill-McKee, conjugate gradient method, pseudoperipheral vertex.

SUMÁRIO

1	Introdução	12
1.1	Motivação	14
1.2	Objetivos	14
2	Referencial Teórico	15
2.1	Conceitos básicos	15
2.2	Método dos gradientes conjugados	19
2.3	Cuthill-Mckee e Cuthill-Mckee reverso	21
2.3.1	Descrição dos algoritmos	23
2.3.2	Análise de complexidade	25
2.3.3	Exemplo	26
2.4	Algoritmo de George-Liu para encontrar um vértice pseudoperiférico	29
3	Metodologia	34
3.1	Metodologia de pesquisa	34
3.2	Detalhes da implementação	35
3.3	Ferramentas utilizadas	37
4	Resultados	39
4.1	Simulações sem o algoritmo para redução de banda	40
4.2	Simulações com o método Cuthill-Mckee reverso	41

4.3	Simulações com CMr-pseudo	41
4.4	Exemplos de matrizes de coeficientes	43
5	Considerações finais e trabalhos futuros	46

LISTA DE FIGURAS

2.1	Representação da matriz por meio de um grafo.	16
2.2	Grafo com a numeração alterada.	16
2.3	Estrutura de nível de $\mathcal{L}(1)$ baseada no grafo da figura 2.1.	19
2.4	Grafo original para um exemplo do CM e sua representação matricial M	26
2.5	Passo inicial da ordenação pelo Cuthill-McKee.	27
2.6	Passo a passo da ordenação pelo Cuthill-McKee.	28
2.7	Grafo reordenado pelo Cuthill-McKee e sua representação matricial M_1	28
2.8	Grafo reordenado pelo Cuthill-McKee reverso e sua representação ma- tricial M_2	28
2.9	Um grafo para escolha do vértice pseudoperiférico e sua representação matricial M	32
2.10	Geração da estrutura de nível para o vértice 1.	32
2.11	Geração da estrutura de nível para o vértice 2.	33
3.1	Triangulação de Delaunay numerada com 19 vértices.	36
3.2	Diagrama de atividades do projeto computacional.	38
4.1	Representação de uma matriz de coeficientes com, aproximadamente, 1200 vértices sem a redução de banda.	44
4.2	Representação de uma matriz de coeficientes com, aproximadamente, 1200 vértices com a redução de banda realizado pelo CMr.	45

4.3	Representação de uma matriz de coeficientes com, aproximadamente, 1200 vértices com a redução de banda realizado pelo CMr iniciando-se por um vértice pseudoperiférico	45
-----	--	----

LISTA DE TABELAS

2.1	Vértices adjacentes ao vértice 9 e seus graus.	26
2.2	Largura de banda e de <i>profile</i> de matrizes ordenadas do exemplo para os métodos CM e CMr.	29
4.1	Dados referentes à execução do método do GC e ao sistema de equações lineares sem redução de banda.	40
4.2	Dados referentes à execução do método dos gradientes conjugados e ao sistema de equações lineares com redução de banda pelo CMr, comparados com os resultados mostrados na tabela 4.1.	42
4.3	Dados referentes à execução do método do método dos gradientes conjugados e ao sistema de equações lineares com redução de banda pelo CMr iniciando-se por um vértice pseudoperiférico, comparados com os resultados mostrados na tabela 4.2.	43

1 INTRODUÇÃO

Na resolução de diversos problemas científicos, como em problemas de dinâmica de fluidos computacional, por métodos numéricos, como o método dos volumes finitos, o método dos elementos finitos ou o método das diferenças finitas, geram-se sistemas de equações lineares. Na maioria dos casos, com a utilização desses métodos numéricos, geram-se matrizes de coeficientes de grande porte e esparsas. Exige-se muita memória e alto custo de processamento para armazenar e tratar essas matrizes.

Para a solução dos sistemas de equações lineares provenientes de problemas científicos, utilizam-se métodos diretos ou iterativos, em que, por exemplo, a eliminação de Gauss é um método direto e o método dos gradientes conjugados é um método iterativo. Para a solução de sistemas de equações lineares com matrizes de coeficientes esparsas, um dos métodos iterativos eficiente é o método dos gradientes conjugados (BITTENCOURT; FEIJÓO, 1997; BURDEN; FAIRES, 2003, p. 404).

Com a *redução de banda* da matriz de coeficientes, pode-se obter baixo custo computacional na resolução do sistema de equações lineares. A minimização de banda de uma matriz simétrica A é aproximar o máximo possível todos os coeficientes não nulos em relação à diagonal principal de A . Conforme demonstrado por Papademetrious (1976), o problema da minimização da banda, ou seja, encontrar a menor largura de banda de uma matriz simétrica A é um problema NP-Completo. Portanto, algoritmos que reduzem a banda de uma matriz por meio de permutações de linhas e colunas das matrizes deixando-as com uma estrutura compacta e com coeficientes não nulos próximos à diagonal principal são estudados. Realizam-se essas permutações de linhas e colunas, de uma matriz simétrica A , alterando-se a numeração do grafo correspondente a A .

Especificamente, neste trabalho, apresenta-se o algoritmo para redução de banda Cuthill-McKee reverso (CMr) de George (1971). Dado um sistema de equações lineares $Ax = b$ inicialmente esparso e a representação da matriz A de coeficientes por meio de um grafo, a redução de banda dessa matriz de coeficientes se dá por meio da mudança da ordem, em que se numeram os vértices do grafo. Para isso o CMr utiliza uma busca em largura modificada.

Implementou-se o CMr porque é um dos algoritmos mais tradicionais para o problema da redução de banda. Foi um dos primeiros métodos, junto com os algoritmos de Alway e Martin (1965) e Akyuz e Utku (1968), proposto para essa finalidade. Esse algoritmo é importante; como exemplos, pesquisas atuais sobre redução de banda utilizam esse algoritmo como referência para realizar comparações, como em Doss e Arathi (2011). Ainda, é utilizado na resolução de problemas científicos atuais, em que as matrizes de coeficientes de sistemas de equações lineares são alterados para se melhorar o desempenho da resolução por métodos diretos e iterativos, como em Kazoela, Nikolos e Synolakis (2012) e Shiming, Wei e Xiang (2013).

A escolha do vértice inicial para numeração do CMr é diretamente ligada à qualidade da redução de banda. Portanto, inicia-se a numeração por um vértice que atende a algumas características. Esse vértice é chamado de vértice pseudo-periférico e é abordado na subseção 2.4. Escolheu-se o algoritmo de George e Liu (1979) para encontrar um vértice pseudoperiférico inicial para o CMr. Utilizou-se esse algoritmo por ser um dos algoritmos clássicos para esse propósito e por apresentar melhorias em relação ao algoritmo original de Gibbs, Poole e Stockmeyer (1976).

Este texto está organizado como a seguir. A motivação deste trabalho é apresentada na subseção 1.1. Os objetivos são descritos na subseção 1.2. O referencial teórico é apresentado na seção 2. Na seção 3, Descrevem-se a metodologia de pes-

quisa utilizada e as ferramentas. Apresentam-se na seção 4, os resultados obtidos nas simulações realizadas. Finalmente, na seção 5, apresentam-se as conclusões e os trabalhos futuros.

1.1 Motivação

A principal motivação deste estudo, é tentar proporcionar melhorias em um projeto computacional de geração de malhas triangulares, tornando sua execução mais rápida. Nesse projeto computacional, geram-se sistemas de equações lineares esparsos, em que são resolvidos pelo método dos gradientes conjugados. Essas malhas triangulares são utilizadas em vários campos da ciência, como em simulações em mecânica dos fluídos.

1.2 Objetivos

O objetivo deste trabalho é verificar se haverá redução no custo computacional das resoluções, pelo método dos gradientes conjugados, de alguns sistemas de equações lineares em que houve reduções de banda pelo CMr nas matrizes de coeficientes. Especificamente, neste trabalho, o método CMr será acoplado a um projeto computacional de geração de malhas triangulares.

2 REFERENCIAL TEÓRICO

Nesta seção, apresentam-se conceitos básicos para o entendimento deste trabalho. Na subseção 2.1, abordam-se conceitos básicos sobre largura de banda e grafos. Aborda-se o método dos gradientes conjugados na subseção 2.2. Na subseção 2.3, são apresentados os métodos Cuthill-Mckee e Cuthill-Mckee reverso. Na subseção 2.4, apresenta-se o algoritmo de George e Liu (1979) para se encontrar o vértice (pseudoperiférico) inicial para o algoritmo CMr.

2.1 Conceitos básicos

Os sistemas de equações lineares considerados são da forma $Ax = b$, em que A é uma matriz $n \times n$ esparsa, simétrica e positiva-definida. Uma matriz é positiva-definida se todos os seus autovalores são positivos. Os elementos de A são representados por a_{ij} , em que i é o índice referente à linha e j é o índice da coluna.

Adiante, considere um grafo $G = (V, E)$ não orientado constituído por V , um conjunto finito de vértices e E , um conjunto de arestas, em que $V = \{v_1, v_2, v_3, \dots, v_n\}$ e as arestas de $E = \{e_1, e_2, e_3, \dots, e_m\}$ são representadas por pares não ordenados de vértices. Tem-se as seguintes definições:

- dois vértices são adjacentes quando existe uma aresta entre eles;
- o grau de um vértice v é o número de vértices adjacentes a v ;
- caminho é uma sequência de vértices que, para cada um, há uma aresta para o próximo vértice da sequência;
- o tamanho de um caminho é o seu número de arestas;
- ciclo é um caminho com o mesmo vértice inicial e final;

- em um grafo conexo, existe um caminho entre quaisquer dois de seus vértices;
- árvore é um grafo conexo acíclico.

Para mais detalhes sobre grafos, veja Oliveira (2011, capítulo 8).

Na figura 2.1, tem-se a representação de um grafo por meio de uma matriz de adjacências. Cada linha da matriz A corresponde a um vértice no grafo da figura 2.1, ou seja, para $1 \leq i \leq 10$, a linha i da matriz corresponde ao vértice i do grafo. Para o grafo da figura 2.2, utilizou-se uma ordem diferente da anterior para numerar os vértices do grafo e obteve-se uma representação matricial também diferente. Nota-se que a disposição dos elementos não nulos na representação matricial do grafo depende da ordem em que se numeram os vértices do grafo para a montagem da matriz. É da ordem dessa numeração de que trata este texto. Para isso, há algoritmos que definem essa ordem e o vértice inicial da numeração. Na subseção seguinte, abordam-se conceitos matemáticos relacionados com a escolha do vértice inicial da numeração.

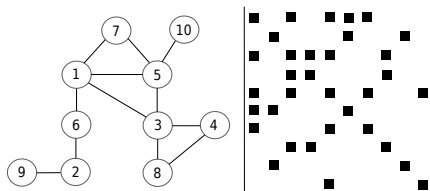


Figura 2.1: Representação da matriz por meio de um grafo.

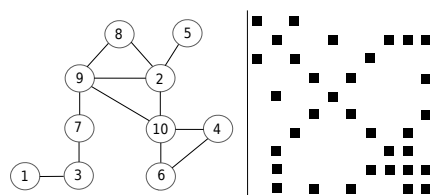


Figura 2.2: Grafo com a numeração alterada.

Definição 1 (largura de banda) *Seja A uma matriz simétrica, a largura de banda da i -ésima linha é $\beta_i(A) = i - \min\{(1 \leq j \leq n) \mid a_{ij} \neq 0\}$. A largura de banda $\beta(A)$ de A é a maior distância de elemento não nulo da matriz triangular inferior*

até a diagonal principal, considerando-se todas as n linhas da matriz, ou seja,
 $\beta(A) = \max\{(1 \leq i \leq n) \beta_i(A)\} = \max\{(1 \leq i, j \leq n) | i - j | \mid a_{ij} \neq 0\}$.

Definição 2 (banda) A banda de A é definida como $banda(A) = \{(0 < i - j \leq \beta(A)) \mid \{i, j\}\}$.

A definição 2 significa que a banda é formada pelas diagonais da matriz triangular inferior que contenham elementos não nulos. Note que a diagonal principal não compõe a banda. Ainda, note que só se considera a matriz triangular inferior porque A é simétrica.

Definição 3 (envelope) O envelope de A é definido como $Env(A) = \{(1 \leq i, j \leq n) \mid \{i, j\} \mid 0 < i - j \leq \beta_i(A)\}$.

A definição 3 significa que os elementos nulos mais à esquerda da i -ésima linha, na $banda(A)$, não compõem o envelope.

Definição 4 (profile) O profile de A , definido como $|Env(A)| = \sum_{i=1}^n \beta_i(A)$, é a soma das distâncias, em cada linha, do primeiro elemento não nulo até a diagonal principal.

Para detalhes sobre a definição 4, veja George e Liu (1981, p. 48-52).

Definição 5 (distância) Seja $G = (V, E)$. A distância $d(v, u)$ entre dois vértices v e u pertencentes a V é o tamanho do menor caminho entre eles. Considere as distâncias $(\forall u \in V \wedge u \neq v) d(v, u)$ de v para todos os demais vértices do grafo.

Definição 6 (excentricidade) A excentricidade de um vértice $v \in V$ é dada por $\ell : V \rightarrow \mathbb{N}$, em que $\ell(v) = \max\{(\forall u \in V \wedge u \neq v) d(v, u)\}$.

Definição 7 (diâmetro do grafo) O diâmetro $\delta(G) = \max\{(\forall v \in V) \ell(v)\}$ do grafo $G = (V, E)$ é a maior excentricidade encontrada em $G = (V, E)$, que também pode ser definida como $\delta(G) = \max\{(\forall v, u \in V) d(v, u)\}$.

Definição 8 (vértice periférico) Um vértice v é considerado periférico se a sua excentricidade é igual ao diâmetro do grafo, ou seja, $\ell(v) = \delta(G)$.

Para Gibbs, Poole e Stockmeyer (1976), um vértice é considerado *pseudoperiférico* se sua excentricidade for próxima ao diâmetro do grafo.

Definição 9 (estrutura de nível) Uma estrutura de nível de um vértice de um grafo conexo e simples é um particionamento do conjunto de vértices em classes de vértices com as mesmas distâncias para o vértice raiz. Seja $G = (V, E)$ conexo e simples. Dado um vértice $v \in V$, a estrutura de nível com raiz v e profundidade $\ell(v)$ é o particionamento $\mathcal{L}(v) = \{L_0(v), L_1(v), \dots, L_{\ell(v)}(v)\}$, em que $L_0(v) = \{v\}$ e $L_i(v) = \text{adj}(L_{i-1}(v)) - \bigcup_{j=0}^{i-1} L_j(v)$, para $i = 1, 2, 3, \dots, \ell(v)$ e $\text{adj}(\cdot)$ retorna os vértices adjacentes aos vértices do argumento.

Note que o número de partições (ou níveis) de $\mathcal{L}(v)$ é $\ell(v) + 1$ e $\bigcup_{i=0}^{\ell(v)} L_i(v) = V$. Considerando-se que é utilizada a busca em largura para a montagem da estrutura de nível e para a numeração, a maior largura de banda possível é igual ao grau do vértice de maior grau do grafo.

Mostra-se um exemplo de montagem de uma estrutura de nível na figura 2.3. Para isso, considere o grafo da figura 2.1. Para a estrutura de nível mostrada na figura 2.3, o vértice inicial 1 foi escolhido arbitrariamente. Note que $\ell(1) = 3$. O particionamento em níveis dos vértices, partindo de 1, é $L_0(1) = \{1\}$, $L_1(1) = \{3, 5, 6, 7\}$, $L_2(1) = \{2, 4, 8, 10\}$ e $L_3(1) = \{9\}$.

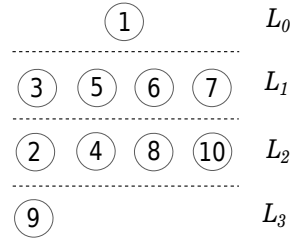


Figura 2.3: Estrutura de nível de $\mathcal{L}(1)$ baseada no grafo da figura 2.1.

Definição 10 (largura de nível) A largura de nível $b(u)$ de $\mathcal{L}(u)$ é o número de vértices do nível com mais vértices, ou seja, $b(u) = \max\{|L_i(u)|, \text{ para } i = 0, 1, \dots, \ell(u)\}$.

2.2 Método dos gradientes conjugados

Apresenta-se, nesta seção, uma breve descrição do método dos gradientes conjugados. Considere que o algoritmo descrito nesta seção, não é o algoritmo implementado no projeto computacional criado para as simulações publicadas em Oliveira e Gonzaga de Oliveira (2012) e sim apenas uma descrição didática. No projeto computacional (OLIVEIRA; Gonzaga de Oliveira, 2012), armazenam-se apenas os coeficientes não nulos da matriz A por meio de uma lista de adjacências.

O método dos gradientes conjugados foi proposto por Hestenes e Stiefel (1952). Verificou-se que o método dos gradientes conjugados é um eficiente método iterativo para solução de sistemas de equações lineares, simétricos, positivos-definidos e esparsos (BURDEN; FAIRES, 2003, p. 404). Todos os sistemas de equações lineares abordados neste trabalho são simétricos e positivos-definidos.

Seja um sistema de equações lineares na forma $Ax = b$, em que $A \in \mathbb{R}^{n \times n}$, x e $b \in \mathbb{R}^n$. A solução desse sistema de equações lineares, pode ser encontrada por

meio da minimização de $f(x)$, com $x \in \mathbb{R}^n$. Seja f , uma função quadrática,

$$f(x) = \frac{x^T A x}{2} - b^T x, \quad (2.1)$$

o vetor x^* é uma solução do sistema de equações lineares se, e somente se, x^* minimiza f . Ao derivar f da equação 2.1 em relação ao vetor x e igualar a zero, obtém-se

$$\nabla f(x^*) = Ax^* - b = 0, \quad (2.2)$$

ou seja, o método dos gradientes conjugados soluciona o sistema de equações lineares quando encontra o mínimo de f , $f'(x) = 0$.

Inicia-se o método dos gradientes conjugados escolhendo-se uma solução inicial x_0 , uma solução aproximada de $Ax^* = b$. Considere que, inicialmente, $r_0 = b - Ax_0$ e $d_1 = r_0$, em que r_k é o vetor resíduo na iteração k e d_k é o vetor de direção de busca na iteração k , com $k = 0, 1, \dots, n-1$. O método dos gradientes conjugados constrói os vetores r e d , a cada iteração, por

$$r_k = r_{k-1} - \alpha_k A d_k \quad (2.3)$$

e

$$d_{k+1} = r_k + \beta_k d_k. \quad (2.4)$$

Os escalares α da equação 2.3 e β da equação 2.4 são dados por

$$\alpha_k = \frac{r_{k-1}^T r_{k-1}}{d_k^T A d_k} \quad (2.5)$$

e

$$\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}. \quad (2.6)$$

A solução em cada iteração é dada por

$$x_k = x_{k-1} + \alpha_k d_k. \quad (2.7)$$

Os vetores r e d auxiliam o método dos gradientes conjugados na busca do mínimo de f , em que cada iteração a solução x_k aproxima-se mais de x^* . O vetor d satisfaz a condição de *A-ortogonalidade* (ou conjugada), $d_i^T A d_j = 0$, com $i \neq j$. Em Luenberger e Ye (2008, p. 263), tem-se a definição formal de *A-ortogonalidade*. O uso dos vetores de direção d *A-ortogonais* produz o método de direção conjugada (BURDEN; FAIRES, 2003, p. 409). Resíduos são ortogonais entre si, portanto, $r_i^T r_j = 0$, para $i \neq j$. Ainda, os vetores r_i e d_j , com $i \neq j$, são mutuamente ortogonais, ou seja, $r_i^T d_j = 0$.

Com as equações 2.3, 2.4, 2.5, 2.6 e 2.7, forma-se o algoritmo iterativo do método dos gradientes conjugados. Esse algoritmo é mostrado no algoritmo 1. Inicialmente, define-se a precisão numérica ε desejada. Nas linhas 3 a 5, inicializam-se os vetores x, r e d . As variáveis k e *erro* são inicializadas na linha 6, em que k é o contador para a estrutura de repetição *enquanto* e o *erro* é a variável para o critério de parada. A cada iteração atualiza-se $\text{erro} \leftarrow \frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$, em que $\|x\|_\infty = \max\{|x_1|, |x_2|, \dots, |x_n|\}$. Se $\text{erro} \geq \varepsilon$, o algoritmo para.

O método dos gradientes conjugados converge para a solução em no máximo n iterações. Para mais detalhes sobre o método método dos gradientes conjugados, veja Shewchuk (1994) e Burden e Faires (2003, p. 404-415), por exemplo.

2.3 Cuthill-McKee e Cuthill-McKee reverso

Com o método de Cuthill e McKee (1969), reduz-se o tamanho da largura de banda de uma matriz simétrica por meio da reordenação do grafo que corresponde à matriz. O algoritmo faz uso da busca em largura, em que todos os vértices do grafo são visitados. A ordem de visitação dos vértices é o que difere um algoritmo de busca em largura convencional do algoritmo de Cuthill-McKee (CM). Neste último, a visitação dos vértices se dá de forma crescente ao grau dos vértices,

Algoritmo 1: método dos gradientes conjugados.

```

1  início
    // define a precisão numérica  $\varepsilon$ , ex.  $10^{-3}$ 
2   $\varepsilon \leftarrow 10^{-3}$ ;
3   $x_0 \leftarrow 0$ ; // estimativa inicial
4   $r_0 \leftarrow b - Ax_0$ ;
5   $d_1 \leftarrow r_0$ ;
6   $k \leftarrow 1$ ;  $erro \leftarrow \varepsilon + 1$ ;
    //  $n$  é a dimensão de  $x$ 
7  enquanto (  $(k < n) \wedge (erro > \varepsilon)$  ) faça
8       $\alpha_k \leftarrow \frac{r_{k-1}^T r_{k-1}}{d_k^T A d_k}$ ; // equação 2.5
9       $x_k \leftarrow x_{k-1} + \alpha_k d_k$ ; // equação 2.7
10      $r_k \leftarrow r_{k-1} - \alpha_k A d_k$ ; // equação 2.3
11      $erro \leftarrow \frac{\|x_k - x_{k-1}\|_\infty}{\|x_k\|_\infty}$ ;
12      $k \leftarrow k + 1$ ;
13     se (  $(k < n) \wedge (erro > \varepsilon)$  ) então
14          $\beta_k \leftarrow \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$ ; // equação 2.6
15          $d_{k+1} \leftarrow r_k + \beta_k d_k$ ; // equação 2.4
16     fim-se;
17 fim-enquanto;
18 fim.
```

isto é, dado um vértice inicial, seus vértices adjacentes serão visitados em ordem crescente de grau e isso ocorrerá para os vértices adjacentes desses adjacentes e assim sucessivamente até que todos os vértices sejam visitados. A visitação dos vértices adjacentes em ordem crescente proporciona uma boa configuração da matriz correspondente em relação a sua largura de banda, pois representam-se os

elementos do grafo com maior grau para que fiquem posicionados, no possível, nas linhas centrais da matriz.

Cuthill e Mckee (1969) constataram que a escolha do vértice inicial interferia na qualidade da solução: na largura de banda e no *profile* da matriz resultante. Então, propuseram que o algoritmo começasse pelo vértice de grau mínimo do grafo. Embora apresente bons resultados, isso não garante que a largura de banda ou o *profile* sejam menores que uma solução em que o vértice inicial seja o vértice de maior excentricidade.

A seguir, na subseção 2.3.1, são apresentados os algoritmos CM e Cuthill-Mckee reverso (CMr). Na subseção 2.3.2, são apresentadas as análises de complexidade desses algoritmos. Na subseção 2.3.3, um exemplo é mostrado.

2.3.1 Descrição dos algoritmos

O método CM é mostrado no algoritmo 2 (LIU, 1976). O algoritmo recebe um grafo $G = (V, E)$ conexo, em que V é o conjunto de vértices, E é o conjunto de arestas. Outro parâmetro do algoritmo é o vértice inicial $v \in V$. Nas linhas 2 a 4, inicializam-se as variáveis, em que i recebe 1 e é o contador para se percorrerem os vértices. A primeira posição de α recebe o vértice inicial. A variável j indica a posição na ordenação. Enquanto i for menor que o número de vértices, a cada iteração, numeram-se os vértices adjacentes a $\alpha(j)$, em ordem crescente de grau, exceto aqueles que já foram numerados. Após cada w adjacente a $\alpha(j)$ ser numerado, w é colocado na posição i de α , após incremento de i . Note que não há necessidade de se usar $\leq$ na linha 5 porque o último vértice verificado é adjacente a algum outro vértice. A saída do algoritmo é a sequência de vértices no vetor α .

George (1971) descobriu que renumerar os vértices na ordem contrária em que foram visitados, em vez da numeração habitual do CM, diminui, frequentemente, o tamanho dos *profiles* das matrizes resultantes após a execução do algoritmo;

Algoritmo 2: Cuthill-McKee.

Entrada: grafo $G = (V, E)$ conexo; vértice inicial $v \in V$;

Saída: vetor α ordenado;

1 **início**

2 $i \leftarrow 1$;

3 $\alpha(1) \leftarrow v$;

4 $j \leftarrow 0$;

5 **enquanto** ($i < |V|$) **faça**

6 $j \leftarrow j + 1$;

7 **para cada** (vértice $w \in \text{adj}(\alpha(j)) - \{\alpha(1), \dots, \alpha(i)\}$, em
ordem crescente de grau) **faça**

8 $i \leftarrow i + 1$;

9 $\alpha(i) \leftarrow w$;

10 **fim-para-cada**;

11 **fim-enquanto**;

12 // A saída é a sequência no vetor α ;

13 **retorna** α ;

14 **fim.**

porém, a largura de banda é mantida. Então, George (1971) propôs o CMr, que é o CM, mas, com ordenação *reversa*. Liu (1976) provou que o CMr resulta em matrizes com *profiles*, pelo menos, tão bons quanto o CM.

Na versão original do CMr (GEORGE, 1971), o vértice inicial também é aquele com grau mínimo. Em ocorrendo mais de um vértice com grau mínimo, o vértice inicial é o que possui a menor largura de nível. As reduções de banda e de *profile* da matriz resultante depende da escolha do vértice inicial. Essa tarefa rendeu trabalhos, que são abordados na seção 2.4, na página 29.

2.3.2 Análise de complexidade

Suponha que o grafo é representado por listas de adjacências. Para encontrar o vértice de grau mínimo, necessita-se percorrer todos os vértices. Para cada vértice, percorre-se toda a lista de adjacências para se verificar o grau do vértice. De uma forma similar, utilizando-se o algoritmo busca em largura, ao percorrer a lista de adjacências de cada vértice, pode-se verificar o grau de cada vértice. Dessa forma, o custo para se encontrar o vértice de grau mínimo é $O(|V| + |E|)$. Em se utilizando uma matriz de adjacências, pode-se percorrer as $|V|^2$ entradas para se encontrar o vértice de grau mínimo.

Conforme demonstrado por Liu (1976), a complexidade, no pior caso, dos algoritmos Cuthill-McKee e Cuthill-McKee reverso é dada pelo produto entre o grau m do vértice de maior grau do grafo e o número de arestas $|E|$. Liu (1976) obteve a complexidade, no pior caso, considerando que, no passo de ordenação dos vértices na linha 7 do algoritmo 2, fosse utilizado o algoritmo por inserção. Com isso, considere que todas as ordenações na linha 7 exijam, no máximo, $\frac{n^2}{c}$ operações, para n vértices e uma constante c . Esse número de operações é menor que $\frac{1}{c} \sum_{v \in V} \text{grau}(v)^2 \leq \frac{m}{c} \sum_{v \in V} \text{grau}(v) \leq \frac{2m|E|}{c}$ porque, para um grafo não direcionado, ocorre $\sum_{v \in V} \text{grau}(v) = 2|E|$. Com isso, $\frac{2m|E|}{c}$ é o limite superior do número de operações na linha 7. O número de operações do laço de repetição na linha 5 é $\Theta(|V|)$. Com isso, a complexidade no, pior caso, desse algoritmo é $O(m \cdot |E|)$. O limite inferior desse algoritmo, considerando-se um grafo em que o vértice inicial tem ligações com todos os demais vértices e utilizando-se, por exemplo, o merge sort no passo da ordenação, é $\Omega(|V| \lg |V|)$.

2.3.3 Exemplo

A seguir, é mostrado um exemplo da execução do método Cuthill-McKee. Apresenta-se apenas o funcionamento do algoritmo de Cuthill-McKee, pois o CM e o CMr diferem apenas na numeração final. Considere o grafo da figura 2.4 e, ao seu lado, sua representação matricial M . Neste exemplo, o vértice inicial não é o de grau mínimo, como é nos algoritmos CM e CMr. Escolheu-se, arbitrariamente, o vértice 9 como o inicial. Coloca-se esse vértice na primeira posição do vetor α . Em seguida, visitam-se os vértices adjacentes ao vértice 9, colocando-os, em ordem crescente de grau, em suas respectivas posições i do vetor α . Na figura 2.5, mostram-se, em destaque, o vértice 9 e suas arestas incidentes. Na tabela 2.1, mostram-se os vértices adjacentes ao vértice 9, ordenados por seus graus. Os vértices são atribuídos a cada posição i do vetor α nessa ordem.

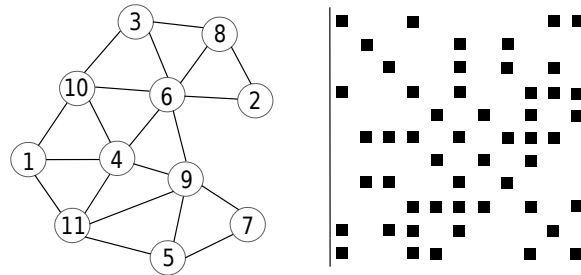


Figura 2.4: Grafo original para um exemplo do CM e sua representação matricial M .

Vértice	7	5	11	4	6
Grau	2	3	4	5	6

Tabela 2.1: Vértices adjacentes ao vértice 9 e seus graus.

O próximo vértice a ser processado é o vértice 7, que ocupa a segunda posição do vetor. Não são inseridos novos vértices no vetor porque os vértices adjacentes ao vértice 7 já foram visitados. O mesmo ocorre com o vértice 5. Ao processar

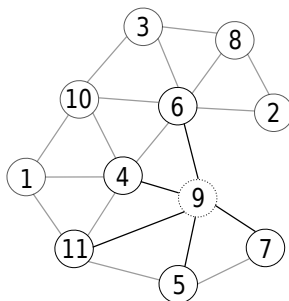


Figura 2.5: Passo inicial da ordenação pelo Cuthill-McKee.

o vértice 11, o vértice 1 é inserido na primeira posição vazia do vetor α . Isso é realizado até que todos os vértices sejam visitados. Veja a ordem de visitação na figura 2.6. Algoritmos para encontrar o vértice inicial, ou pseudoperiférico, são abordados na seção 2.4, na página 29.

O resultado da numeração é mostrado na figura 2.7, juntamente com a matriz correspondente M_1 . Na figura 2.8, representa-se o grafo com a numeração pelo Cuthill-McKee reverso e sua representação matricial M_2 . Nesse exemplo, $\beta(M) = 10$, $\beta(M_1) = \beta(M_2) = 5$, $|Env(M)| = 39$, $|Env(M_1)| = 33$ e $|Env(M_2)| = 24$. Na tabela 2.2, esses dados são sumarizados. Note que, com a numeração reversa, as colunas da matriz M_1 transformam-se nas linhas da matriz M_2 . Conforme destacado pela linha tracejada, as larguras de banda $\beta_4(M_1)$ e $\beta_5(M_1)$ não são contabilizadas no *profile* de M_2 . Os primeiros elementos dessas linhas pertencem à última linha de M_2 . Com isso, há redução no *profile* de M_2 em relação ao *profile* de M_1 . São mostradas, ainda, as numerações do CM e CMr iniciando-se por vértices pseudoperiféricos (CM-pseudo e CMr-pseudo, respectivamente).

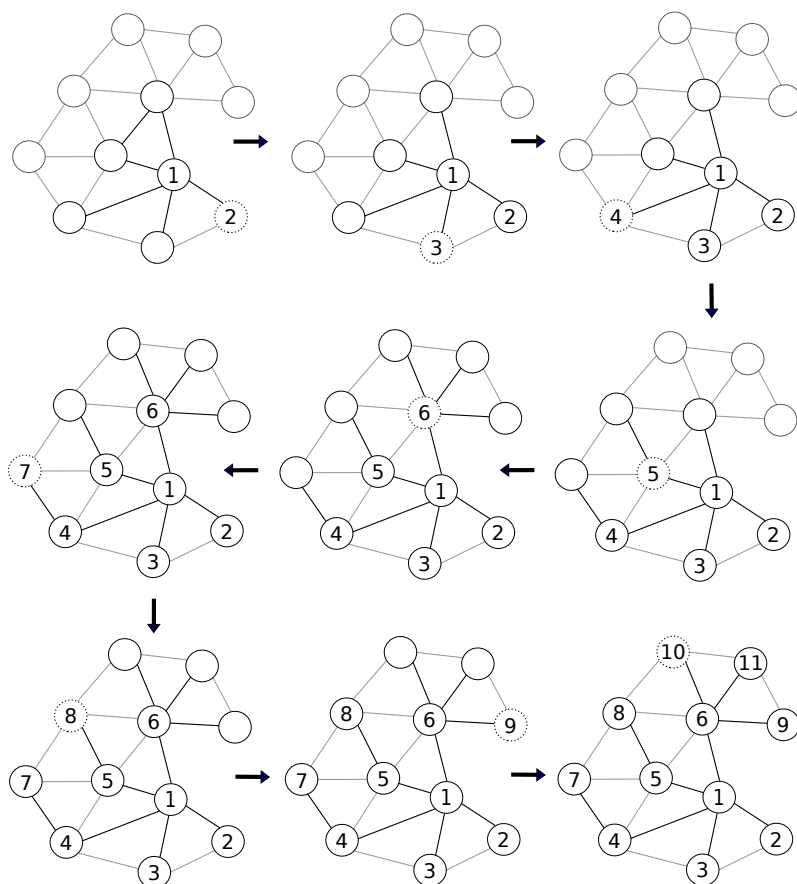


Figura 2.6: Passo a passo da ordenação pelo Cuthill-McKee.

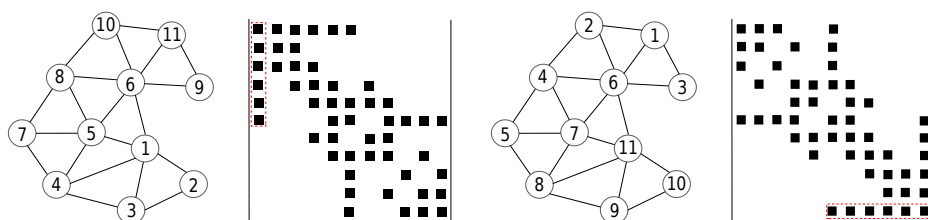


Figura 2.7: Grafo reordenado pelo Cuthill-McKee e sua representação matricial M_1 . **Figura 2.8:** Grafo reordenado pelo Cuthill-McKee reverso e sua representação matricial M_2 .

<i>Matriz</i>	Largura de banda	<i>Profile</i>
M (original)	10	39
M_1 (CM)	5	33
M_2 (CMr)	5	24
CM-pseudo	4	26
CMr-pseudo	4	25

Tabela 2.2: Largura de banda e de *profile* de matrizes ordenadas do exemplo para os métodos CM e CMr.

2.4 Algoritmo de George-Liu para encontrar um vértice pseudoperiférico

Alguns algoritmos, como o Cuthill-McKee, o Cuthill-McKee reverso e o algoritmo de Gibbs, Poole e Stockmeyer (1976), têm a qualidade da ordenação dependente da escolha do vértice inicial. Essa constatação foi realizada, inicialmente, por Cuthill e McKee (1969). Esses autores iniciaram a ordenação com o vértice de grau mínimo do grafo, que nem sempre era uma boa escolha. Após isso, houve publicações sobre algoritmos que encontram o vértice inicial, como Cheng (1973). Contudo, apenas na publicação de Gibbs, Poole e Stockmeyer (1976), foi proposto que o melhor vértice inicial seria um vértice cuja distância entre ele e outro vértice qualquer seja igual ao diâmetro do grafo, caracterizando-se assim vértices periféricos. Essa constatação deu-se pelo motivo que, como haviam Cuthill e McKee (1969) abordado em sua publicação, quanto maior a excentricidade de um vértice v , menos vértices haverá em um mesmo nível de uma estrutura de nível de v . Cuthill e McKee (1969) explicaram que a largura de banda da matriz será relacionada com o número de vértices do nível da estrutura de nível com o maior número de vértices. Um vértice periférico é ótimo vértice inicial para os algoritmos de or-

denação. Smyth e Benzi (1974) demonstraram que, para realizar essa tarefa, seria necessário, por exemplo, aplicar a busca em largura em todos os vértices do grafo. O custo dessa tarefa é $O(|V| \cdot |E|)$, em que V é o número de vértices e E é o número de arestas. Gibbs, Poole e Stockmeyer (1976) propuseram, então, um algoritmo que encontra um vértice cuja excentricidade fosse próxima ao diâmetro do grafo, caracterizando-se um vértice pseudoperiférico.

O algoritmo de George-Liu foi proposto como melhorias do algoritmo de Gibbs, Poole e Stockmeyer (1976), em que George e Liu (1979) propuseram 4 modificações do algoritmo original. A primeira modificação, citada no texto como *short circuiting*, consiste em terminar a montagem da estrutura de nível dos vértices folhas assim que encontrar um nível com largura de nível maior do que a já encontrada. A segunda modificação é iniciar o algoritmo com um vértice qualquer em vez de um com grau mínimo. A terceira e quarta modificações são denominadas *shrinking*, pois ambas são estratégias de qual vértice escolher do último nível da estrutura de nível. A terceira melhoria é montar a estrutura de nível apenas para o vértice folha que possuir o menor grau. A última proposta de modificação é escolher um vértice folha qualquer para a montagem da estrutura de nível. Após experimentos, o algoritmo final proposto por George e Liu (1979), é a combinação da segunda e terceira propostas de melhoria. Esse algoritmo é descrito nesta subseção.

Como mostrado por George e Liu (1979), o algoritmo pode ser dividido em alguns passos simples. Na parte da inicialização, um vértice qualquer $v \in V$ é escolhido para ser o vértice inicial. Posteriormente, a estrutura de nível $\mathcal{L}(v)$ é gerada. Em seguida, escolhe-se um vértice de grau mínimo $r \in L_{\ell(v)}(v)$. Gera-se $\mathcal{L}(r)$ e verifica-se se $\ell(v) < \ell(r)$. Se essa condição for satisfeita, então, o vértice r passa a ser o vértice v e o processo é repetido. Caso contrário, o vértice pseudoperiférico será v .

Mostra-se o pseudocódigo no algoritmo 3. Inicializa-se a variável v com um vértice arbitrário. Em seguida, é montada a estrutura de nível $\mathcal{L}(v)$, utilizando a busca em largura. Na estrutura de repetição *enquanto* nas linhas 5-11, constrói-se a estrutura de nível do nó folha r de grau mínimo de $\mathcal{L}(v)$. Se a excentricidade de r for maior que a excentricidade de v , então, atribui-se r a v e repete-se o processo. O algoritmo retorna o nó v quando $\ell(v)$ for maior que a excentricidade do seu nó folha de grau mínimo.

Algoritmo 3: George-Liu.

Entrada: um grafo $G = (V, E)$;

Saída: vértice pseudoperiférico $v \in V$;

1 início

2 $v \leftarrow \text{escolheVertice}(V)$;

3 $\text{buscaEmLargura}(v)$; // constrói-se estrutura de nível

4 **repita**

 // u é utilizada como controle do laço de
 repetição

5 $u \leftarrow v$;

6 $r \leftarrow \text{verticeComMinimoGrau}(\mathcal{L}_{\ell(v)}(v))$;

7 $\text{buscaEmLargura}(r)$; // constrói-se estrutura de
 nível

8 **se** ($\ell(r) > \ell(v)$) **então**

9 $v \leftarrow r$;

10 **fim-se;**

11 **até que** $u = v$;

12 **retorna** v ;

13 fim.

Como um exemplo de funcionamento do algoritmo de George e Liu (1979), considere o grafo representado na figura 2.9. O vértice v escolhido arbitrariamente para a inicialização do algoritmo foi o vértice 1. Na figura 2.10, mostra-se a montagem da estrutura de nível iniciando-se no vértice 1. Note que o número de níveis de $\mathcal{L}(1)$ é 4 e $\ell(1) = 3$.

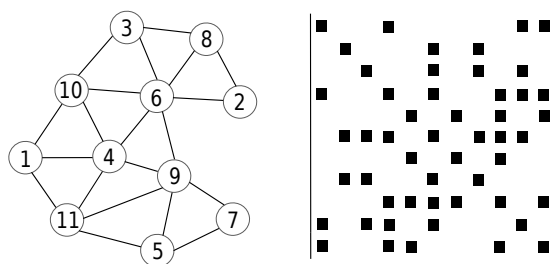


Figura 2.9: Um grafo para escolha do vértice pseudoperiférico e sua representação matricial M .

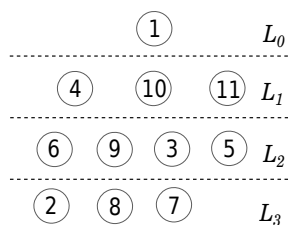


Figura 2.10: Geração da estrutura de nível para o vértice 1.

Atribui-se à variável r o vértice do último nível com grau mínimo. Como os vértices 2 e 7 apresentam grau 2, indifere a escolha de um ou outro. Nesse exemplo, optou-se pelo vértice 2. Como passo seguinte do algoritmo, a estrutura de nível do vértice 2 é construída. Isso pode ser visto na figura 2.11. Como $\ell(v) = \ell(r)$, o algoritmo para, determinando-se o vértice $v = 1$ como o vértice pseudoperiférico. Contudo, pode ser observado que, nesse exemplo, o algoritmo de fato encontrou o vértice periférico.

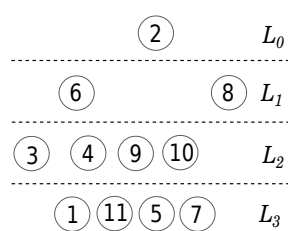


Figura 2.11: Geração da estrutura de nível para o vértice 2.

3 METODOLOGIA

Nesta seção, descrevem-se as metodologias de pesquisa e ferramentas utilizadas durante o trabalho. Na subseção 3.1, descrevem-se as metodologias de pesquisa. Apresentam-se na subseção 3.2, detalhes da implementação do projeto computacional. Descrevem-se na subseção 3.3, as ferramentas utilizadas durante a realização deste trabalho.

3.1 Metodologia de pesquisa

As metodologias de pesquisa utilizadas nesse trabalho foram descritas conforme definições de [Zambalde, Pádua e Alves \(2008\)](#). O conhecimento necessário adquirido para a realização deste trabalho foi obtido por meio da leitura de artigos e livros. Muitos desses artigos foram encontrados pela base de periódicos da CAPES.

O trabalho foi desenvolvido em laboratório de pesquisas do DCC-UFLA. A natureza de pesquisa desse trabalho, classifica-se como aplicada, pois os conhecimentos adquiridos foram utilizados para implementar os algoritmos estudados. O procedimento de pesquisa utilizado nesse trabalho classifica-se como operacional, pois deseja-se reduzir o custo computacional do método dos gradientes conjugados (GC). Este estudo tem caráter de pesquisa quantitativa, porque será verificado se a resolução de um sistema de equações lineares pelo gradiente conjugado será mais rápida com a redução de banda.

3.2 Detalhes da implementação

O objetivo do projeto computacional criado para as simulações publicadas em Oliveira e Gonzaga de Oliveira (2012) é solucionar a equação de Laplace no domínio computacional. O projeto computacional utilizado para acoplar o método CMr foi desenvolvido em C++ durante o programa de iniciação científica (OLIVEIRA; Gonzaga de Oliveira, 2012). Nesse projeto computacional, geram-se malhas triangulares em um domínio quadrado por meio da triangulação de Delaunay, para a solução da equação de Laplace com as condições de contorno: $T(a, 0) = 0$ para $0 < a < 1$ e $T(a, 1) = T(0, a) = T(1, a) = 10$ para $0 \leq a \leq 1$.

Para a discretização da equação diferencial parcial, utilizou-se o método dos volumes finitos. Polígonos de Voronoi foram utilizados como volumes de controle.

Cada vértice da triangulação, exceto os vértices da borda, recebem um número. Por meio da ordem dessa numeração, gera-se o sistema de equações lineares. Soluciona-se o sistema de equações lineares pelo método dos gradientes conjugados. Refina-se a malha pelo algoritmo de Ruppert (1995). Malhas de qualidade são geradas por *off-centers* de Üngör (2004, 2009). Para detalhes sobre a equação de Laplace, veja Iorio (2007), por exemplo. A triangulação de Delaunay e o diagrama de Voronoi são detalhados em Edelsbrunner (2001), por exemplo. Para detalhes sobre o método dos volumes finitos, veja Maliska (2004), por exemplo.

No projeto computacional, a malha é um grafo e os vértices desse grafo são ligados por uma lista encadeada. Gera-se o sistema de equações lineares conforme a ordem dessa lista encadeada. No projeto computacional sem redução de banda pelo método CMr ou pelo CMr-pseudo, os novos vértices gerados pelo refinamento são inseridos no final dessa lista encadeada. Isso gera um sistema de equações lineares com matriz de coeficientes com largura de banda grande. No

projeto computacional, para reduzir a banda e o *profile* do sistema de equações lineares, permuta-se a ordem dos vértices nessa lista encadeada pelo método CMr.

Um exemplo de triangulação de Delaunay, com 19 vértices, é mostrado na figura 3.1. Nessa malha, não se utilizou algoritmo para redução de banda. Há vértices com numeração baixa adjacentes a vértices com numeração alta. Um exemplo disso é o vértice 1, que é adjacente ao vértice 18. Isso proporciona uma largura de banda grande na matriz de coeficientes que representa esse grafo. Na malha da figura 3.1, os vértices da borda recebem o número -1. Claramente, não fazem parte do sistema de equações lineares.

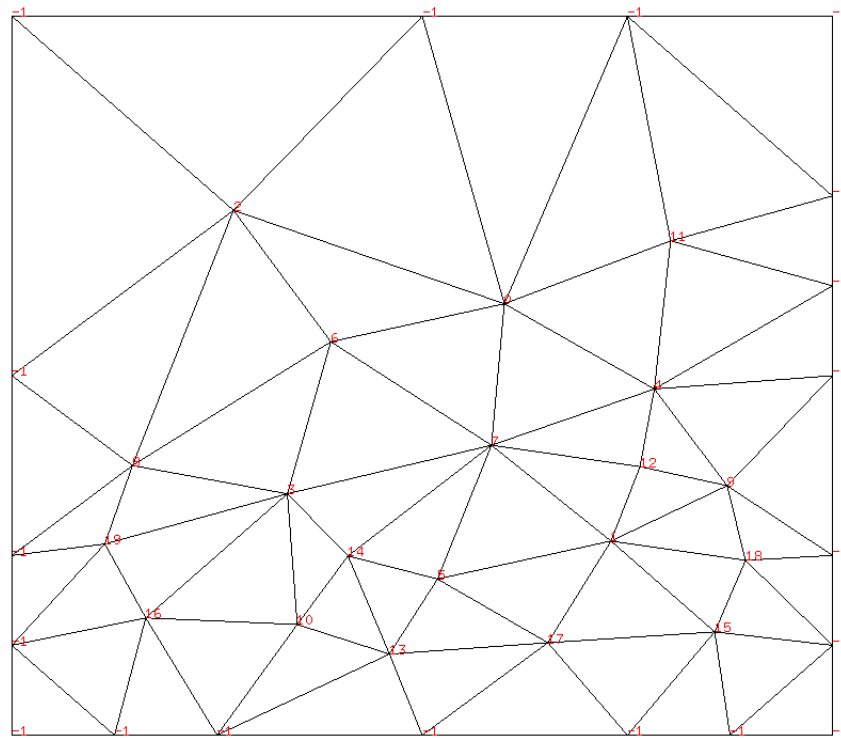


Figura 3.1: Triangulação de Delaunay numerada com 19 vértices.

Mostra-se, na figura 3.2, o diagrama de atividades do projeto computacional com o método CMr. O programa começa com uma malha inicial e com configurações iniciais. Em seguida, gera-se o sistema de equações lineares referente

à malha inicial pelo método dos volumes finitos. Claramente, o método CMr foi inserido entre a montagem do sistema de equações lineares e a resolução pelo método dos gradientes conjugados, conforme é mostrado na figura 3.2. Depois da resolução do sistema de equações lineares pelo método dos gradientes conjugados, se os critérios de refinamento não forem atendidos, o programa para. Se os critérios de refinamento forem atendidos, a malha é refinada pelo algoritmo de Ruppert (1995). Em seguida, refina-se a malha por *off-centers*, gerando-se uma malha de qualidade. No projeto computacional criado para as simulações publicadas em Oliveira e Gonzaga de Oliveira (2012), utiliza-se a medida de qualidade de triângulos *circunradius-to-shortest edge ratio*. Isso significa que uma malha é dita de qualidade se todos os ângulos de todos os triângulos são menores que $\arcsin \frac{1}{2B}$. O limite B deve ser maior que a razão do raio do circuncírculo do triângulo pelo comprimento da menor aresta do triângulo. Finalmente, retorna-se ao passo de montagem do sistema de equações lineares.

3.3 Ferramentas utilizadas

Nas simulações, utilizou-se um computador IBM[®] x3400 M3 com processador Intel[®] Xeon[®] E5620 2.40GHz com 12Mb de memória *cache*, 16Gb de memória RAM DDR3 1333MHz. O sistema operacional utilizado foi o Ubuntu 12.04 LTS 64bits com *kernel* versão 3.2.0-38-generic. Utilizou-se a IDE NetBeans 7.2.1 com *plugin* para C/C++ para a implementação dos projetos computacionais.

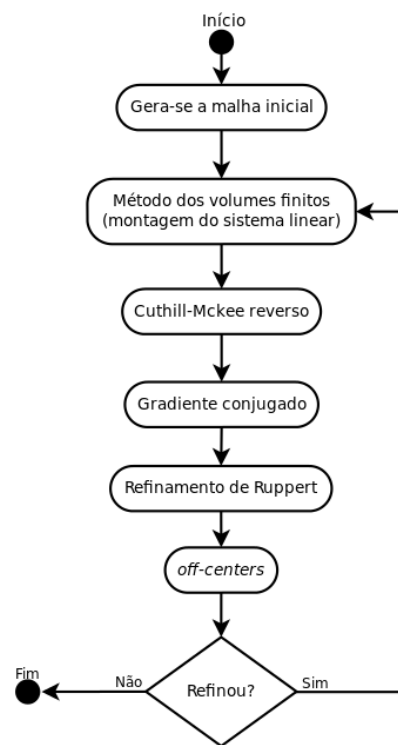


Figura 3.2: Diagrama de atividades do projeto computacional.

4 RESULTADOS

Os resultados das simulações são apresentados nesta subseção. Utilizou-se a mesma malha em todos os testes realizados. Porém, a cada iteração, conforme pode ser observado na figura 3.2, refina-se a malha. Com isso, a cada iteração, um sistema de equações lineares maior do que o da iteração anterior é gerado. Como o intuito é avaliar o desempenho do método dos gradientes conjugados, aplicou-se o método dos gradientes conjugados 10 vezes na mesma iteração e mediu-se o tempo em cada execução.

Fez-se a média, em segundos, dessas 10 execuções. Como o tempo de execução não variou muito (desvio padrão, em média, menor que 0,04), considerou-se que dez execuções foram suficientes nessas simulações. Os tempos de execuções do CMr e CMr-pseudo não foram sumarizados, pois não foram considerados significativos. Por exemplo, na simulação com 279557 os métodos CMr e CMr-pseudo apresentaram tempo médio de execução menor que 1s.

Os detalhes das simulações são mostrados nas subseções seguintes. Na subseção 4.1, mostram-se os dados referentes à simulação do projeto computacional (OLIVEIRA; Gonzaga de Oliveira, 2012) sem o método CMr. Apresentam-se os dados da execução do projeto computacional (OLIVEIRA; Gonzaga de Oliveira, 2012), com o método CMr e sem o algoritmo que encontra um vértice pseudoperiférico, na subseção 4.2. Os dados da execução do projeto computacional (OLIVEIRA; Gonzaga de Oliveira, 2012), com o método CMr iniciando por um vértice pseudoperiférico, são apresentados na subseção 4.3.

4.1 Simulações sem o algoritmo para redução de banda

Mostram-se os dados da execução do método dos gradientes conjugados no projeto computacional (OLIVEIRA; Gonzaga de Oliveira, 2012) sem a utilização do método CMr na tabela 4.1. Na primeira coluna, mostram-se o número de vértices da malha. Mostram-se, na segunda coluna, em segundos, o tempo médio do método dos gradientes conjugados para solucionar o sistema de equações lineares com tamanho referente ao número de vértices. São mostrados, na terceira coluna, os números de iterações que o método dos gradientes conjugados utilizou para solucionar os sistemas de equações lineares. Mostram-se as larguras de banda e os *profiles* referentes aos sistemas de equações lineares na quarta e quinta coluna, respectivamente.

Resolução de sistemas de equações lineares sem redução de banda				
Vértices	GC: tempo	GC: iterações	β	<i>profile</i>
4885	0,546s	402	4460	7741207
12203	3,039s	657	11726	52697023
32522	22,172s	1076	19224	142986662
54174	56,824s	1456	31725	393777694
89343	138,832s	1893	53166	1111666737
130863	260,146s	2296	88211	3079787252
178907	428,391s	2686	129448	12173473282
224016	603,047s	2967	221821	18578615675
256113	753,879s	3189	253919	23535750631
279557	819,093s	3142	277230	26846000209

Tabela 4.1: Dados referentes à execução do método do GC e ao sistema de equações lineares sem redução de banda.

Como em todas as dez execuções do método dos gradientes conjugados utilizou-se o mesmo sistema de equações lineares, não houve variação no número de iterações. O mesmo ocorre com a largura de banda e o *profile*.

4.2 Simulações com o método Cuthill-McKee reverso

Mostram-se os dados da execução do método dos gradientes conjugados no projeto computacional (OLIVEIRA; Gonzaga de Oliveira, 2012) com a utilização do método CMr para redução sem a utilização do algoritmo que encontra um vértice pseudoperiférico, na tabela 4.2. Inicialmente, esperava-se que reduzir a banda da matriz de coeficientes de um sistema de equações lineares diminuiria o número de iterações do método dos gradientes conjugados e, conseqüentemente, houvesse diminuição do tempo de execução. Entretanto, ao se analisar os números de iterações mostrados nas tabelas 4.1, 4.2 e 4.3, nota-se que, praticamente, não houve variação no número de iterações. Porém, houve redução no tempo médio de execução do método dos gradientes conjugados com a utilização do método CMr, em relação às simulações do projeto computacional sem a utilização do CMr. Isso pode ser observado nas tabelas 4.1 e 4.2. É possível que as iterações do método dos gradientes conjugados sejam mais rápidas após a redução de banda da matriz de coeficientes do que sem a redução. Na tabela 4.2, a terceira, a sexta e a oitava colunas são as porcentagens das reduções do tempo de execução do método dos gradientes conjugados, largura de banda e de *profile*, respectivamente, em relação à tabela 4.1.

4.3 Simulações com CMr-pseudo

Mostram-se os dados da execução do método dos gradientes conjugados no projeto computacional (OLIVEIRA; Gonzaga de Oliveira, 2012) com a utilização

Resolução de sistemas de equações lineares com redução de banda pelo CMr							
Vértices	GC: tempo	Redução	GC: iterações	β	Redução	<i>profile</i>	Redução
4885	0,494s	9,52%	402	165	96,30%	525882	93,20%
12203	2,878s	5,29%	654	281	97,60%	2147050	95,92%
32522	20,288s	8,49%	1084	457	97,62%	9549110	93,22%
54174	47,586s	16,25%	1383	590	98,14%	20386818	94,82%
89343	123,600s	10,97%	1888	747	98,59%	43027960	96,12%
130863	233,804s	10,12%	2307	937	98,93%	81879952	97,34%
178907	383,404s	10,50%	2687	1139	99,12%	137745953	98,86%
224016	541,292s	10,24%	2963	1288	99,41%	199617546	98,92%
256113	679,310s	9,89%	3192	1387	99,45%	245402391	98,95%
279557	743,309s	9,25%	3143	1462	99,47%	275907946	98,97%

Tabela 4.2: Dados referentes à execução do método dos gradientes conjugados e ao sistema de equações lineares com redução de banda pelo CMr, comparados com os resultados mostrados na tabela 4.1.

do método CMr para redução de banda iniciando-se por um vértice pseudoperiférico, na tabela 4.3. Note que o número de iterações do método dos gradientes conjugados continuou praticamente o mesmo, conforme mostrado nas tabelas 4.1, 4.2 e 4.2.

Nas simulações com o CMr-pseudo, houve redução no tempo médio em todas as execuções do método dos gradientes conjugados em relação ao tempo das execuções do método dos gradientes conjugados, mostradas na tabela 4.1. Em relação aos resultados do tempo médio de execução do método dos gradientes conjugados das simulações do CMr, o CMr-pseudo apresentou redução no tempo na maioria dos testes, exceto com 54174 e 224016 vértices. Como pode ser observado na tabela 4.3, com 54174 e 224016 vértices, o CMr-pseudo apresentou tempo médio

superior ao CMr. Comparando-se os resultados referentes à largura de banda e ao *profile*, em todas as simulações, houve reduções dos resultados do CMr-pseudo em relação ao CMr. Mostram-se, nas colunas 4, 5 e 7 da tabela 4.3, as porcentagens das reduções do tempo de execução do método dos gradientes conjugados, largura de banda e de *profile* em relação aos dados mostrados na tabela 4.2.

Resolução de sistemas lineares com redução de banda pelo CMr-pseudo							
Vértices	GC: tempo	Redução	GC: iterações	β	Redução	<i>profile</i>	Redução
4885	0,489s	1,01%	407	104	36,96%	316492	39,81%
12203	2,802s	2,64%	650	141	49,82%	1164292	45,77%
32522	19,886s	1,69%	1080	249	45,51%	5295436	44,54%
54174	49,212s	-3,41%	1441	337	42,88%	11684275	42,68%
89343	122,556s	0,89%	1888	409	45,24%	24888681	42,15%
130863	231,196s	1,11%	2292	533	43,11%	47695096	41,74%
178907	380,474s	0,76%	2685	640	43,81%	73159699	39,90%
224016	543,786s	-0,46%	2961	775	39,82%	131844302	33,95%
256113	675,790s	0,57%	3166	967	30,28%	171531053	30,50%
279557	737,620s	0,76%	3125	1052	28,04%	264427621	29,53%

Tabela 4.3: Dados referentes à execução do método do método dos gradientes conjugados e ao sistema de equações lineares com redução de banda pelo CMr iniciando-se por um vértice pseudo-periférico, comparados com os resultados mostrados na tabela 4.2.

4.4 Exemplos de matrizes de coeficientes

Mostram-se alguns exemplos de matrizes de coeficientes dos sistemas de equações lineares do projeto computacional. Na figura 4.1, mostra-se a matriz de coeficientes do sistema de equações lineares, com, aproximadamente, 1200 vértices, sem a redução de banda. Na figura 4.2, mostra-se a matriz de coeficientes do

sistema de equações lineares, com, aproximadamente, 1200 vértices, com redução de banda pelo método CMr. Na figura 4.3, mostra-se a matriz de coeficientes do sistema de equações lineares, com, aproximadamente, 1200 vértices, com redução de banda pelo método CMr, iniciando-se por um vértice pseudoperiférico.

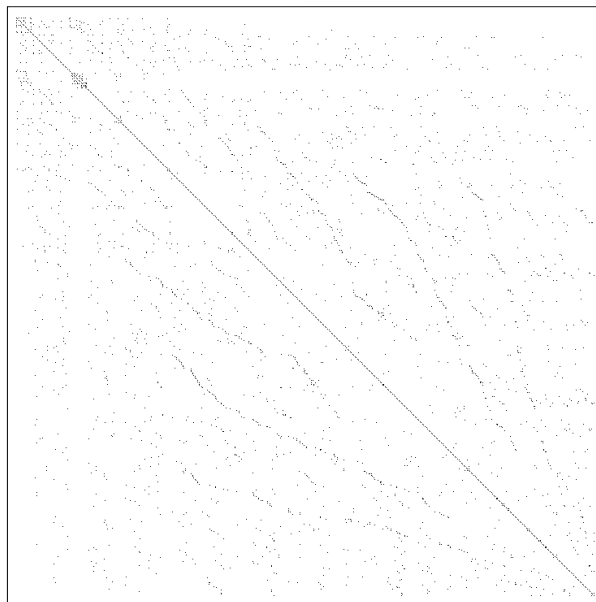


Figura 4.1: Representação de uma matriz de coeficientes com, aproximadamente, 1200 vértices sem a redução de banda.

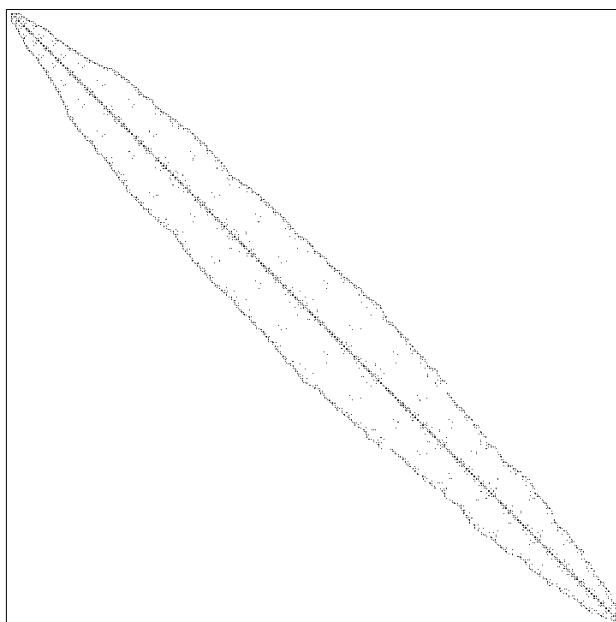


Figura 4.2: Representação de uma matriz de coeficientes com, aproximadamente, 1200 vértices com a redução de banda realizado pelo CMr.

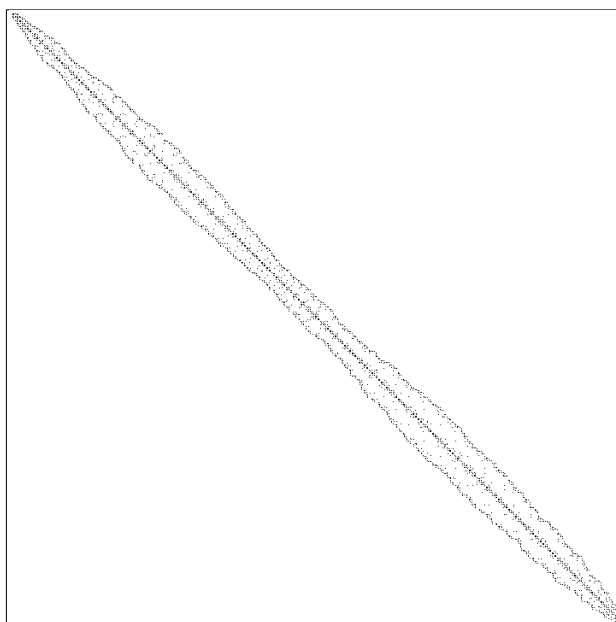


Figura 4.3: Representação de uma matriz de coeficientes com, aproximadamente, 1200 vértices com a redução de banda realizado pelo CMr iniciando-se por um vértice pseudoperiférico

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Neste trabalho implementou-se o método Cuthill-Mckee reverso (CMr) de George (1971) e o algoritmo de George e Liu (1979), que encontra um vértice pseudoperiférico. Verificou-se o desempenho do método dos gradientes conjugados com redução de banda, após a utilização desses métodos, em matrizes de coeficientes de sistemas de equações lineares.

Com os resultados apresentados na seção 4, verificou-se que, ao se utilizar o CMr para se reduzir a banda da matriz de coeficientes de um sistema de equações lineares, houve redução no tempo de execução do método dos gradientes conjugados nas simulações deste trabalho. Verificou-se que a redução de tempo de execução do método dos gradientes conjugados, após a execução do CMr, não é proporcional à redução da banda da matriz do sistema linear.

Ainda, com a utilização do CMr-pseudo, não se obteve redução significativa no tempo de execução do método dos gradientes conjugados em relação aos resultados obtidos com o CMr. O ganho em tempo computacional, com a utilização do CMr, proporciona uma melhoria em relação ao tempo de execução no projeto computacional criado para as simulações publicadas em Oliveira e Gonzaga de Oliveira (2012). Em trabalhos futuros, pretende-se verificar por que o número de iterações do método dos gradientes conjugados não diminuiu com a utilização de um método para reduções de banda e de *profile*. Um análise do condicionamento das matrizes deve ser realizada com e sem reduções de banda e de *profile*.

Em trabalhos futuros, pretende-se verificar por que o número de iterações do método dos gradientes conjugados, não diminuiu com a utilização de um método para reduções de banda e de *profile*. Um análise do condicionamento das matrizes deve ser realizada com e sem reduções de banda e de *profile*. Pretende-se, tam-

bém, realizar uma revisão dos métodos para redução de banda de matrizes. Ainda, pretende-se identificar qual melhor algoritmo para reduções de banda e de *profile* de matrizes. Ainda, deseja-se implementar esse algoritmo e incorporá-lo no projeto computacional de Oliveira e Gonzaga de Oliveira (2012).

REFERÊNCIAS BIBLIOGRÁFICAS

AKYUZ, F. A.; UTKU, S. An automatic relabeling scheme for bandwidth minimization of stiffness matrices. *American Institute of Aeronautics and Astronautics*, v. 6, p. 728–730, 1968.

ALWAY, G. G.; MARTIN, D. W. An algorithm for reducing the bandwidth of a matrix of symmetrical configuration. v. 8, n. 3, p. 264–272, out. 1965. ISSN 0010-4620 (print), 1460-2067 (electronic).

BITTENCOURT, M. L.; FEIJÓO, R. A. Análise comparativa de métodos diretos e iterativos para soluções de sistema de equações. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, v. 13, n. 2, p. 123–148, 1997.

BURDEN, R. L.; FAIRES, J. D. *Análise Numérica*. Boston, USA: Thomson Learning, 2003.

CHENG, K. Note on minimizing the bandwidth of sparse, symmetric matrices. *Computing*, Springer Wien, v. 11, p. 27–30, 1973.

CUTHILL, E.; MCKEE, J. Reducing the bandwidth of sparse symmetric matrices. In: 24TH NATIONAL CONFERENCE ACM. New York, 1969. p. 157–172.

DOSS, L. J. T.; ARATHI, P. A bandwidth reduction algorithm for L-shaped and Z-shaped grid structured graphs. *Operations Research Letters*, v. 39, p. 441–446, 2011.

EDELSBRUNNER, H. *Geometry and Topology for Mesh Generation*. Cambridge, New York: Cambridge University Press, 2001.

GEORGE, A. *Computer implementation of the finite element method*. Tese (Doutorado) — Stanford University, Stanford, USA, 1971.

GEORGE, A.; LIU, J. W. H. An implementation of a pseudoperipheral node finder. *ACM Transactions on Mathematical Software*, n. 5, p. 284–295, 1979.

GEORGE, A.; LIU, J. W. H. *Computer solution of large sparse positive definite systems*. Englewood Cliffs: Prentice-Hall, 1981.

GIBBS, N. E.; POOLE, W. G.; STOCKMEYER, P. K. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis*, v. 2, n. 13, p. 236–250, 1976. Disponível em: <http://doi.acm.org/10.1145/355705.355707>.

HESTENES, M. R.; STIEFEL, E. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, v. 49, n. 36, 1952.

IORIO, V. d. M. *EDP: um curso de graduação*. Rio de Janeiro: IMPA, 2007.

KAZOELA, M.; NIKOLOS, A. L.; SYNOLAKIS, C. E. An unstructured finite volume numerical scheme for extended 2D Boussinesq-type equations. *Coastal Engineering*, v. 69, p. 42–66, 2012.

LIU, J. W. H. *On reducing the profile of sparse symmetric matrices*. Tese (Doutorado) — University of Waterloo, fev. 1976.

LUENBERGER, D.; YE, Y. *Linear and Nonlinear Programming*. Stanford, USA: Springer, 2008.

MALISKA, C. R. *Transferência de Calor e Mecânica dos Fluidos Computacional*. Rio de Janeiro: LTC, 2004.

OLIVEIRA, S. L. G. d. Introdução a grafos . In: _____. *Algoritmos e seus fundamentos*. Lavras, Brazil: Editora UFLA, 2011. cap. 8, p. 215–226.

OLIVEIRA, T. H.; Gonzaga de Oliveira, S. L. Uma resolução da equação de laplace por volumes finitos e diagrama de voronoi com refinamento adaptativo e

de ruppert. In: *Proceedings of the SIBGRAPI 2012 - International Conference on Graphics, Patterns and Images - Workshop of Undergraduate Works (WUW)*. Ouro Preto: SBC, 2012.

PAPADEMETRIOUS, H. C. The NP-completeness of bandwidth minimization problem. *Computing Journal*, v. 16, p. 177–192, 1976.

RUPPERT, J. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *Journal of Algorithms*, v. 18, n. 3, 1995.

SHEWCHUK, J. R. *An introduction to the conjugate gradient method without the agonizing pain*. 1. ed. Pittsburgh: School of Computer Science Carnegie Mellon University, 1994.

SHIMING, X.; WEI, X.; XIANG, L. Performance modeling and optimization of sparse matrix-vector multiplication on NVIDIA CUDA platform. *The Journal of Supercomputing*, v. 63, n. 3, p. 710–721, 2013.

SMYTH, W. F.; BENZI, W. M. L. An algorithm for finding the diameter of a grap. In: IFIP CONGRESS. Stockholm, Sweden, 1974. p. 500–503.

ÜNGÖR, A. Off-Centers: a new type of steiner points for computing size-optimal quality-guaranteed delaunay triangulations. In: *6th Latin American Symposium*. Buenos Aires, Argentina: Springer, 2004. v. 2976, p. 152–161.

ÜNGÖR, A. Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. *Computational Geometry*, v. 42, n. 2, p. 109–118, 2009.

ZAMBALDE, A. L.; PÁDUA, S. P. I. C.; ALVES, M. R. O documento científico em Ciência da computação e Sistemas de Informação. DCC- UFLA. Lavras, MG. Notas de aula – rascunho – texto em construção sem revisão de português e citações. 2008. Disponível em: <http://algol.dcc.ufla.br/~zambalde/aulas-/Apostila_PDF.pdf>.