Eliane Maria Dias

Métodos Numéricos para Resolução de Sistemas Lineares Esparsos de Grande Porte

unesp

MÉTODOS NUMÉRICOS PARA RESOLUÇÃO DE SISTEMAS LINEARES ESPARSOS DE GRANDE PORTE

ELIANE MARIA DIAS

Orientador Profª Drª Sandra Maria Venturelli Ferreira Dias

Dissertação apresentada ao DCCE/IBILCE/UNESP como parte dos requisitos para obtenção do título de Mestre em Ciências Matemáticas - Área de Concentração Matemática Aplicada e Computacional.

Inst. de Biociências, Letras e Ciências Exatas - UNESP
BIBLIOTECA
S. J. Rio Preto

654/94

São José do Rio Preto - 1994

À Antonio, Neide, Joaquim e Generosa por sua dedicação e carinho. Tadeu, Elisete, Carlos e Rosângela pelo apoio.

AGRADECIMENTOS

Primeiramente à minha orientadora Sandra Maria Venturelli Ferreira Dias, pelo apoio, dedicação e incentivo durante todas as fases deste trabalho.

Ao Professor Odelar Linhares por seu incentivo e apoio na fase inicial deste trabalho.

À meus amigos Miriam Delmont Zacca, Luís Antonio de Vasconcelos, Júlio Ricardo Sambrano, Luzinete Felix Nascimento, Luiz Antonio do Carmo e Míriam Yumi Hamada companheiros durante toda a realização deste trabalho.

Aos meus queridos Carlos Emannuel, Layane e Flávia, que apesar de nada entenderem, sempre me estimularam.

Aos colegas do Departamento de Matemática da Unesp - Campus de Bauru e do Departamento de Computação e Estatística do IBILCE - Unesp Campus de São José do Rio Preto, pelo apoio recebido.

Às pessoas que tiveram participação indireta neste trabalho.

RESUMO

Neste trabalho apresentamos métodos para resolução numérica de grandes sistemas lineares esparsos. Mostramos como as matrizes associadas aos vários passos do processo de eliminação de Gauss podem ser utilizados para expressar a inversa da matriz dos coeficientes do sistema linear numa forma fatorada (EFI). Apresentamos métodos para minimizar o número de elementos nulos que podem se tornar não nulos (fill-in) no cálculo da EFI e métodos que transformam a matriz dos coeficientes em várias formas desejáveis de matriz esparsa.

Descrevemos métodos diretos para resolução de sistemas lineares esparsos especiais (simétricos e positivos definidos; tridiagonais e tridiagonais de blocos) e um método denominado Partição PAQ que determina as permutações P e Q que particionam a matriz dos coeficientes em suas componentes fortes.

Apresentamos também métodos iterativos clássicos e bloco-iterativo e algumas considerações sobre a convergência destes métodos.

ABSTRACT

In this work we present methods for the numerical solution of large systems of linear equations with a sparse matrix. We show how the matrices associated with the various steps of the process of the Gaussian elimination can be utilized to express the inverse of the coefficient matrix of the linear system in a factored form (EFI). We present methods for minimize the number of elements zeros that can became nonzero (Fill-in) in the EFI and methods which transform the matrix of the coefficients into various desirable forms of sparse matrix.

We describe direct methods for solution of special sparse linear systems (symmetric, positive definite; symmetric and positive definite band; tridiagonal and block tridiagonal) and a method called Partition PAQ which determine the permutations P and Q to partition the matrix of coefficients into its irreducible components.

We also present basic methods and block-iteratives methods and some considerations about the convergence of this methods.

ÍNDICE

INTRODUÇÃO I

CAPÍTULO 1

ARMAZENAMENTO DE MATRIZES ESPARSAS

1.1. Introdução 1

1.2. Listas Acopladas (Linked lists) 3

1.3. Observações sobre os Esquemas de Compactação 11

CAPÍTULO 2

O MÉTODO DE ELIMINAÇÃO DE GAUSS PARA A RESOLUÇÃO DE SISTEMAS LINEARES ESPARSOS

2.1. Introdução 13

2.2. Descrição do Método 13

2.3. Pivotamento e Erros de Arredondamento 23

2.4. A Eliminação na Forma de Inversa (EFI) 25

2.5. Minimização do Número Total de Elementos não nulos na EFI 27

2.6. Estratégia de Markowitz 39

2.7. Armazenamento e Uso da EFI 44

CAPÍTULO 3

MÉTODOS ADICIONAIS PARA A MINIMIZAÇÃO DO ARMAZENAMENTO DA EFI

3.1. Introdução 47

3.2. Formas Desejáveis para a Eliminação de Gauss 48

3.3. Forma Diagonal de Blocos (FDB) 56

3.4. Forma Triangular de Blocos (FTB) 60

unesp

3.5. Forma Triangular de Banda (FTBa) 74

3.6. Forma de Banda (FBa) 76

3.7. Outras Formas Desejáveis 92

3.8. Método de Remoção de Arcos do Digrafo para Quebrar Grandes Ciclos 93

3.9. Inversa das Matrizes na Forma Triangular de Blocos e Forma Borda Triangular de Blocos 95

CAPÍTULO 4

OUTROS MÉTODOS EXATOS DE RESOLUÇÃO DE SISTEMAS LINEARES ESPARSOS ESPECIAIS

4.1. Introdução 101

4.2. Sistemas Simétricos e Positivos Definidos 101

4.3. Sistemas de Banda, Simétricos e Positivos Definidos 105

4.4. Sistemas Tridiagonais de Blocos 113

4.5. Partição PAQ 122

CAPÍTULO 5

MÉTODOS ITERATIVOS PARA RESOLUÇÃO DE SISTEMAS LINEARES ESPARSOS

5.1. Introdução 134

5.2. Métodos Iterativos Clássicos 135

5.3. Métodos Bloco-Iterativos 139

INTRODUÇÃO

A necessidade de se resolver sistemas de equações lineares ocorre numa grande quantidade de problemas científicos. Existem estimativas de que a cada quatro problemas de simulação matemática, três convertem-se em solução de sistemas de equações. Por exemplo, a solução de equações diferenciais através de métodos de discretização implica na resolução de sistemas de equações lineares cuja matriz dos coeficientes é, em geral, de grande porte e tem características de esparsidade (muitos elementos nulos).

Matrizes esparsas ocorrem também na solução de outros problemas práticos como, por exemplo, na análise estrutural, teoria de redes, sistemas de distribuição, teoria dos grafos, teoria genética, teoria do comportamento, etc.

Este trabalho tem como objetivo apresentar alguns dos principais métodos de resolução numérica de sistemas lineares onde a matriz do sistema é grande e esparsa.

Porque trabalharemos com matrizes esparsas de grande porte, no Capítulo 1 mostraremos técnicas para o armazenamento destas matrizes numa forma compactada ("packed form"). Este tipo de armazenamento é vantajoso pois possibilita armazenar na memória interna do computador de maiores dimensões que as possíveis na forma convencional.

Em geral, os métodos de resolução de sistemas lineares podem ser divididos em dois grupos: Métodos Diretos que são os que conduzem à solução exata do sistema, a menos de erros de arredondamento, após um número finito de passos; e, Métodos Iterativos que fornecem uma seqüência de aproximantes da solução, cada um deles obtido dos anteriores pela repetição do mesmo tipo de processo.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

Observamos que quando se aplica o método de eliminação de Gauss (que é um método direto) para a resolução de grandes sistemas esparsos, elementos que eram nulos podem tornarem-se não nulos. Este preenchimento (fill-in) de elementos nulos pode comprometer a utilização dos métodos diretos. Neste sentido, os métodos iterativos podem ser mais eficientes uma vez que eles não introduzem nenhum elemento não nulo. Por isto, estes métodos são muito utilizados na resolução de grandes sistemas esparsos. Mas nem tudo é perfeito: os métodos iterativos exigem um estudo da convergência.

Contudo, na prática, como os programas computacionais não analisam a convergência do processo, a aplicação de um método iterativo pode levar à perda de tempo e trabalho computacional sem que a solução desejada seja encontrada.

Por isso, no Capítulo 2 deste trabalho mostraremos como a eliminação de Gauss pode ser usada para expressar a inversa da matriz dos coeficientes de um sistema linear numa forma fatorada denominada eliminação na forma da inversa (EFI).

Algumas técnicas serão apresentadas para a obtenção, quando possível, da EFI de matrizes esparsas. Também serão descritos métodos que minimizam o número total de operações aritméticas no cálculo da EFI; bem como, o armazenamento e o uso da EFI na prática computacional.

No Capítulo 3, descreveremos métodos que tornam o fill-in pequeno ou nulo e que ao mesmo tempo não requerem grande esforço computacional. Descreveremos métodos que envolvem permutações de linhas e colunas da matriz dos coeficientes para transformá-la numa das várias formas desejáveis para a eliminação de Gauss (por exemplo, forma de banda).

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

No Capítulo 4 descreveremos métodos exatos e respectivos algoritmos para a resolução de sistemas lineares especiais. Apresentaremos métodos usados quando a matriz dos coeficientes é simétrica e positiva definida, de banda simétrica e positiva definida; tridiagonal; e, tridiagonal por blocos. Também apresentamos o método denominado PAQ que determina as permutações P e Q que particionam a matriz dos coeficientes em suas componentes fortes facilitando a resolução de grandes sistemas.

Embora neste trabalho tenhamos nos dedicado principalmente à solução numérica de grandes sistemas esparsos pelo uso dos métodos diretos, apresentaremos no Capítulo 5 em breve relato dos métodos iterativos clássicos e bloco-iterativos para a resolução de sistemas lineares e alguns critérios de convergência destes métodos.

CAPÍTULO 1

ARMAZENAMENTO DE MATRIZES ESPARSAS

1.1. INTRODUÇÃO

O armazenamento ideal é aquele que minimiza, ao mesmo tempo, o total armazenado e o tempo computacional. Porém, em geral, estes dois requisitos são incompatíveis e seu intercâmbio deve ser explorado.

Em geral, as matrizes esparsas de grande porte são armazenadas nos computadores em "packed form" (forma compactada); isto é, somente os elementos não nulos destas matrizes são armazenados. Esta forma de armazenamento é vantajosa pois, possibilita armazenar na memória interna do computador (e realizar todas as operações e armazenamentos envolvidos no programa computacional) matrizes muito maiores que as possíveis na forma convencional. Além disso o tempo computacional total é poupado se as operações envolvendo os elementos nulos não são realizadas.

Por outro lado, o cálculo da matriz inversa na forma fatorada (através do produto de matrizes elementares) onde somente os elementos não triviais de tais matrizes são armazenados em "packed form", usualmente precisa de menos memória que a inversa explícita (por exemplo, cálculo da inversa usando escalonamento) em "packed form". As formas fatoradas de inversas são particularmente vantajosas, quando trabalhamos, por exemplo, com programação linear.

Existem vários esquemas de armazenamento na forma compactada.

Alguns deles são descritos a seguir; outros descreveremos posteriormente.

Se A é uma matriz quadrada de ordem n com τ elementos não nulos, $\tau << n^2$, então A é esparsa. Denotaremos o elemento da i-ésima linha e j-ésima coluna de A por a_{ij}. Para armazenarmos somente os elementos a_{ij} não nulos necessitamos armazenar i, j e a_{ij}. Se usarmos uma posição de memória para o armazenamento de cada um destes valores necessitaremos então de 3τ posições de memória para armazenarmos todos os elementos não nulos de A.

É fácil observarmos a vantagem do processo de "packed list" (listas compactadas), pois ele diminue muito o tempo computacional envolvido no acondicionamento da matriz e dispensa esforço extra. É evidente que 3τ deverá ser substancialmente menor que n para que o esforço computacional e o tempo extra gasto nesta forma de armazenamento sejam compensados.

Em alguns processos, a transformação da matriz A dada em alguma das formas desejáveis (matriz diagonal, tridiagonal, triangular superior ou inferior, de banda, etc...), faz com que sejam criados novos elementos não nulos. Devemos então tomar alguns cuidados e providências para que os novos elementos não nulos nas linhas (ou colunas) de A façam parte dos cálculos. Dentre os vários esquemas de armazenamento de matriz esparsa não existe um que seja o melhor. Tudo depende do contexto em que o esquema será empregado e do hardware que será usado.

A seguir, apresentaremos vários esquemas (em "packed form") para armazenamento de matrizes esparsas.

1.2. LISTAS ACOPLADAS (LINKED LISTS)

Uma das maneiras pelas quais os elementos não nulos de uma matriz esparsa A podem ser armazenados é através da utilização de listas acopladas.

Neste caso, cada elemento não nulo a_{ij} é armazenado como um ítem em sua coluna j. Um ítem é formado por uma terna ordenada (i, a, p) onde:

i é o índice da linha;

a é o valor do elemento a_{ij};

p é o endereço do próximo elemento não nulo da coluna j.

O endereço de **p** é igual a zero se o ítem correspondente ao último elemento não nulo da coluna. Assim, o ítem correspondente a a_{ij} tem a forma:

índice de linha **i** endereço do ítem	valor do elemento **a**	endereço do próximo ítem **p (ou 0)**

A memória total é formada por duas partes:

EC - endereço de início de coluna,

MI - memória para ítens,

A primeira parte, EC, tem **n** locações consecutivas, cada uma delas contendo o endereço de início do primeiro ítem da correspondente coluna.

Por exemplo, a j-ésima posição de EC contém o endereço de início MI(α) do item associado ao primeiro elemento não nulo da coluna j.

A segunda parte, MI, consiste de todos os ítens associados com os elementos não nulos de A. Assim se A possui τ elementos não nulos, como a cada um corresponde um ítem com três elementos, MI - exigirá 3τ posições de memória. Estas posições de memória não precisam ser necessáriamente consecutivas.

Logo, se usarmos listas acopladas é necessário, um total de $3\tau + \eta$ posições de memória para armazenar a matriz esparsa A na forma compactada.

Observamos que se durante os cálculos forem criados novos elementos não nulos em posições antes ocupadas por elementos nulos (o que pode ocorrer, por exemplo, quando se aplica a eliminação de Gauss), a maioria dos esquemas de armazenamento exige, para inserção do novo elemento, o deslocamento de todos os elementos posteriores ao recém criado.

A principal vantagem do esquema de listas acopladas é que os novos elementos não nulos criados podem ser facilmente armazenados em MI. Além disso, as celas em MI não precisam ser consecutivas, basta que elas estejam em grupos que sejam divisíveis por 3.

Observamos que a terceira cela de cada ítem contém o endereço do início do próximo ítem disponível na memória. Caso ele seja o último, então sua terceira cela será nula.

Observe que a criação de um novo elemento não nulo afeta EC e MI.

Suponha $a_{19} = a_{99} = 0$, $a_{29} = 5$, $a_{49} = 15$. Se o armazenamento de EC iniciar no endereço 101 e posteriormente a_{99} tornar-se não nulo, digamos, $a_{99} = 6$ teremos

endereço	101	102	103	104	105	106	107	108	109
conteúdo atual	102	2	5	105	4	15			
novo conteúdo	102	2	5	107	4	15	3	6	105

Fica evidente que somente o conteúdo de uma locação na lista acoplada original será modificado com a inserção de um novo elemento.

Algumas vezes, métodos de compactação que não usam listas acopladas são úteis. Eles usam menos memória, mas os elementos não nulos adicionais podem ser introduzidos somente fazendo a relocação de todos os elementos que o sucedem.

Vamos agora descrever alguns outros esquemas para armazenamento de matriz esparsa.

ESQUEMA 1

A cada elemento não nulo da matriz corresponde um ítem com duas posições de memória. A primeira posição de memória de cada ítem contém o índice da linha e a segunda o valor do elemento. Um índice de linha nulo num ítem significa o final da coluna. Neste caso, a segunda posição do ítem conterá o índice da próxima coluna. Zero em ambas as posições de um ítem significa o fim da matriz.

Assim, neste esquema de armazenamento teríamos um total de $\eta + \tau + 1$ ítens, sendo η para as colunas, τ para os elementos não nulos e 1 para indicar o fim do armazenamento da matriz.

Como cada ítem usa duas posições de memória, serão necessárias um total de $2(\eta + \tau + 1)$ posições para armazenar A.

Assim a matriz A_7 cujos elementos não nulos são a_{41}, a_{61}, a_{52}, a_{19}, a_{99}, a_{24} e a_{75}, para a qual $\tau = 7$ e $\eta = 5$ é armazenada como

(0,1; 4, a_{41}; 6, a_{61}; 0,2; 5, a_{52}; 0,3; 1, a_{13}; 3, a_{33}; 0,4; 2, a_{24}; 0,5; 7, a_{75}; 0,0).

ESQUEMA 2

As informações sobre a matriz dada são armazenadas em três arranjos: **LI** (índice de linha), **VE** (valor do elemento) e **CP** (ponteiro de colunas).

Aqui, LI(k) indica o k-ésimo elemento de LI e contém o índice da linha do correspondente elemento VE(k) de VE.

Se o primeiro elemento não nulo da j-ésima coluna de uma matriz está em VE(t_j) então t_j é armazenado no j-ésimo elemento de CP, isto é CP(j) = t_j.

É evidente que VE e LI contém, cada um, τ elementos e CP contém η elementos; então neste esquema, para armazenar A serão necessárias $2\tau + \eta$ posições de memória.

Por este esquema, o armazenamento de A_7 será feito da seguinte maneira:

VE = (a_{41}, a_{61}, a_{52}, a_{13}, a_{33}, a_{24}, a_{75}).

LI = (4, 6, 5, 1, 3, 2, 7).

CP = (1, 3, 4, 6, 7).

Este esquema é de fácil utilização. Por exemplo, vejamos como recuperar da memória o valor do elemento a_{33}.

Como CP(3) = 4 então LI(4) nos dá o índice da linha do primeiro elemento não nulo da coluna 3. Portanto, LI(4) ou um dos elementos de LI subsequentes deve conter o índice 3 se $a_{33} \neq 0$. Neste caso, LI(5) = 3 e, portanto, VE(5) contém o valor de a_{33}.

Suponhamos agora que desejamos recuperar o elemento a_{15}. Como

CP(5) = 7 e LI(7) = 7, concluímos que o primeiro elemento não nulo da coluna 5 é a_{75}. Portanto $a_{15} = 0$.

ESQUEMA 3

Agora, a cada elemento não nulo da matriz dada é associado um único inteiro f(i,j), dado por:

$$f(i,j) = i + (j - 1)\,\eta \qquad , \qquad a_{ij} \neq 0$$

O armazenamento é feito por dois arranjos: VE contendo os valores dos elementos não nulos e F contendo os valores de f(i,j).

Assim, F(k) contém o f(i,j) correspondente ao a_{ij} em VE(k) onde $k = 1, 2, \ldots, \tau$.

É claro que VE e F possuem cada um τ elementos. Portanto, aqui utiliza-se 2τ posições de memória para armazenar A.

Exemplo1: Seja a matriz

$$A = \begin{pmatrix} . & 4 & 3 & . & . & 1 & . \\ 2 & 4 & . & . & . & . & . \\ . & . & 2 & . & 3 & . & 4 \\ . & . & . & 1 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ 1 & . & . & . & . & . & 5 \end{pmatrix}$$

onde os pontos representam os elementos nulos.

Com este esquema, a matriz será armazenada como:

VE = (2,1,4,4,2,3,2,1,2,3,4,2,1,2,4,5)

F = (2,7,8,9,13,15,17,25,26,31,33,34,36,41,45,49)

A matriz original poderá ser facilmente recuperada a partir deste esquema de armazenamento, pois da definição de f(i,j), temos que:

j é o menor inteiro ≥ f(i,j)/n e i = f(i,j) - (j - 1) n.

Por exemplo se f(i,j) = F (12) = 34, então o menor inteiro maior ou igual a f(i,j)/n = 34/7 é 5. Portanto, j = 5 e, então i = f(i,j) - (j - 1) n = 34 - (5 - 1) 7 = 6.

Consequentemente, o elemento procurado é o a_{65} e seu valor, que está em VE(12), é 2.

Observemos que os esquemas 1, 2 e 3 foram descritos usando armazenamentos por colunas. Pode-se fazer, de modo similar, o armazenamento por linhas.

ESQUEMA 4

Se a matriz A é simétrica e tal que para todo i ≥ j,

$$a_{ij} = 0 \quad , \quad i - j > \theta_i$$

onde $\theta_i << n$ e é geralmente diferente para cada valor de i, então A é chamada matriz simétrica de banda, tendo uma largura de banda localmente variável.

Neste caso, como A é simétrica, somente a parte triangular inferior da matriz A sobre e abaixo da diagonal principal é armazenada.

O armazenamento é feito em dois arranjos: **VE** (valor do elemento) e **PD** (posições dos elementos diagonais em VE).

Para cada linha, os elementos não nulos mais à esquerda e todos os demais elementos à sua direita até e incluindo o elemento diagonal são armazenados em VE.

Assim, a i-ésima linha de A precisa de $\theta_i + 1$ posições de memória e VE terá $\sum_{i=1}^{n} (\theta_i + 1)$ elementos. Adicionando a estes os n elementos necessários para PD concluímos que são precisos um total de

$\sum_{i=1}^{n} \theta_i$ + 2n posições de memória para armazenar A.

Se as bandas são cheias, isto é, $a_{ij} \neq 0$ para todo $i - j \leq \theta_i$, com $i > j$, então $\sum_{i=1}^{n} \theta_i = (\tau - n)/2$ e a memória total necessária será $(\tau + 3n)/2$.

Para ilustrar o esquema, consideremos

$$A = \begin{pmatrix} a_{11} & a_{21} & . & . & . \\ a_{21} & a_{22} & a_{32} & a_{42} & a_{52} \\ . & a_{32} & a_{33} & . & a_{53} \\ . & a_{42} & . & a_{44} & . \\ . & a_{52} & a_{53} & . & a_{55} \end{pmatrix}$$

Devido à simetria armazenaremos (por linha) apenas os elementos sobre e abaixo da diagonal.

Primeiramente, é interessante observar que o primeiro elemento não nulo nas linhas três, quatro e cinco está na segunda coluna. Então, os elementos nulos a_{43} e a_{54} deverão ser armazenados.

Armazenaremos A fazendo:

VE = $(a_{11}, a_{21}, a_{22}, a_{32}, a_{33}, a_{42}, 0, a_{44}, a_{52}, a_{53}, 0, a_{55})$,

PD = (1, 3, 5, 8, 12).

Um elemento a_{ij} da matriz original pode ser recuperado deste esquema como segue:

PD(i)-(i-j) dá a posição de a_{ij} em VE, pois PD(i)-(i-j) > PD (i-1); isto é, a_{ij} não está à esquerda do primeiro elemento não nulo da linha i. Caso contrário $a_{ij} = 0$ e não está armazenado em VE.

Por exemplo, para recuperar a_{53} de VE, temos:

PD (5) - (5 - 3) = 12 - 2 = 10 > 8 = PD (4)

Portanto, a_{53} está guardado em VE (10).

A principal vantagem deste esquema de compactação é que se durante algum procedimento computacional (por exemplo a eliminação de Gauss) elementos adicionais não nulos são criados somente à direita do elemento não nulo mais à esquerda em cada linha, então eles podem ser armazenados em VE sem que seja necessário movimentar-se todos os elementos subsequentes.

ESQUEMA DE SHERMAN

As informações sobre a matriz A são armazenadas em cinco arranjos:

DIAG onde são armazenados os valores dos elementos da diagonal principal de A.

LNZ onde são armazenados os valores dos elementos não nulos de A que se encontram acima ou abaixo da diagonal principal, coluna por coluna, com a posição de início dos elementos não nulos da coluna i, guardada em XLNZ.

XLNZ guarda a posição de início dos elementos não nulos de cada coluna.

NZSUB Os índices de linhas e colunas, de A são guardados no arranjo NZSUB de maneira similar, com a posição inicial em NZSUB para o índice da coluna, sendo dado por XNZSUB(i).

NZSUB(i) Guarda a posição inicial em NZSUB para o índice da coluna i.

Este esquema é denominado Esquema de Sherman (1975).

Para ilustrar, consideremos a seguinte matriz:

$$A = \begin{pmatrix} a_{11} & . & . & . & . & . & . \\ a_{21} & a_{22} & . & . & . & . & . \\ . & . & a_{33} & . & . & . & . \\ a_{41} & a_{42} & . & a_{44} & . & . & . \\ . & . & a_{53} & a_{54} & a_{55} & . & . \\ . & . & a_{63} & . & a_{65} & a_{66} & . \\ . & . & . & . & a_{75} & a_{76} & a_{77} \end{pmatrix}$$

Por este esquema, a matriz é armazenada na memória da seguinte forma:

DIAG	a_{11}	a_{22}	a_{33}	a_{44}	a_{55}	a_{66}	a_{77}		
LNZ	a_{21}	a_{41}	a_{42}	a_{53}	a_{63}	a_{54}	a_{65}	a_{75}	a_{76}
XLNZ	1	3	4	6	7	9	10		
NZSUB	2	4	5	6	5	6	7		
XNZSUB	1	2	3	5	6	7	8		

De NZSUB e XNZSUB vemos que

a primeira coluna tem início na segunda linha,

a segunda coluna tem início na quarta linha,

a terceira coluna tem início na quinta linha,

a quarta coluna tem início na quinta linha,

a quinta coluna tem início na sexta linha,

a sexta coluna tem início na sétima linha.

1.3. OBSERVAÇÕES SOBRE OS ESQUEMAS DE COMPACTAÇÃO

Faremos agora alguns comentários sobre os meios pelos quais é

possível fazer economia adicional nos esquemas de compactação descritos.

Se A é simétrica, com o esquema 4, somente os elementos não nulos da parte triangular superior (ou inferior) serão armazenados.

É possível armazenar dois ou mais ítens das listas acopladas em blocos de locações consecutivas. Deste modo, todos os ítens de um bloco, exceto o penúltimo exigem duas posições de memória em vez de três. É claro que a inserção ou retirada de um ítem do bloco torna-se difícil nesta forma de armazenagem.

Em todos os esquemas de armazenamento que descrevemos, se o número de digitos binários de cada posição de memória do computador é razoavelmente grande, então duas ou mais linhas (ou colunas) de índices podem ser armazenadas em uma mesma posição para poupar a memória. No entanto, isto, exige, conhecimento de linguagem de programação de baixo nível e geralmente, não é possível ser feito se usarmos uma linguagem de programação de alto nível como PASCAL, FORTRAN, ALGOL.

Outros esquemas de armazenamento de matriz serão apresentados adiante quando o contexto assim exigir.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

CAPÍTULO 2

O MÉTODO DE ELIMINAÇÃO DE GAUSS PARA A RESOLUÇÃO DE SISTEMAS LINEARES ESPARSOS

2.1. INTRODUÇÃO

Neste capítulo descreveremos o método de eliminação de Gauss para resolução de um sistema de equações lineares. Mostraremos como as matrizes associadas aos vários passos do processo de eliminação podem ser utilizadas para expressar a inversa da matriz dos coeficientes do sistema linear na forma fatorada. Alguns teoremas serão mostrados e então usados para determinar as formas fatoradas esparsas de inversas de matrizes esparsas com minimização do fill-in (total de elementos que são nulos num passo e se tornam não nulos no passo seguinte da eliminação de Gauss). Mostraremos também o método de Markowitz, 1957 que nos fornece um modo simples mas menos preciso de tornarmos menor o fill-in; e o Teorema de Tewarson, 1967 que pode ser usado para determinar o pivô que torna menor o fill-in.

Consideremos o sistema de equações lineares

$$Ax = b, \qquad (2.1.1)$$

onde a matriz A dos coeficientes é quadrada de ordem n, esparsa, e onde x e b são vetores coluna de dimensão n.

O método de eliminação de Gauss é realizado em duas etapas. Na primeira, A é transformada numa matriz triangular superior U, com 1's na diagonal principal. Na segunda, a matriz U é invertida.

2.2. DESCRIÇÃO DO MÉTODO

A primeira etapa da eliminação de Gauss é realizada em n passos.

Denotaremos por $A^{(k)}$ a matriz no k-ésimo passo, sendo $A^{(1)} = A$ e $A^{(n+1)} = U$. Denominaremos $a_{ij}^{(k)}$ ao elemento da i-ésima linha e j-ésima coluna de $A^{(k)}$; isto é,

$$a_{ij}^{(k)} = e_i^t A^{(k)} e_j,$$

onde e_i é a i-ésima coluna da matriz identidade I_n.

A matriz $A^{(k)}$ é sempre da forma triangular superior nas $(k-1)$ primeiras colunas. Ou seja, ela tem a forma:

$$A^{(k)} = \begin{array}{|ccccccc|} \hline X & X & X & X & X & \ldots & X \\ & & & X & . & & . \\ & & & X & . & & . \\ & X & & . & . & & . \\ & & X & . & . & & . \\ & & & . & . & & . \\ & & & X & \ldots & & X \\ & & & X & \ldots & & X \\ & & & . & . & & . \\ & & & . & . & & . \\ & & & . & . & & . \\ & & & X & \ldots & & X \\ \hline \end{array} \longrightarrow \text{linha } k$$

$$\text{coluna } k \hookleftarrow$$

No k-ésimo passo, a linha k de $A^{(k)}$ é dividida por a_{kk} e seus múltiplos são subtraídos de todas as linhas seguintes de modo que todos os elementos não nulos, situados abaixo da diagonal principal, na k-ésima coluna, tornem-se nulos. A matriz resultante será denotada por $A^{(k+1)}$.

Assim, a primeira etapa da eliminação de Gauss, consiste em calcular

$$A^{(k+1)} = L_k A^{(k)}, \quad k = 1, 2, \ldots, n, \qquad (2.2.1)$$

onde a matriz triangular inferior L_k é dada por

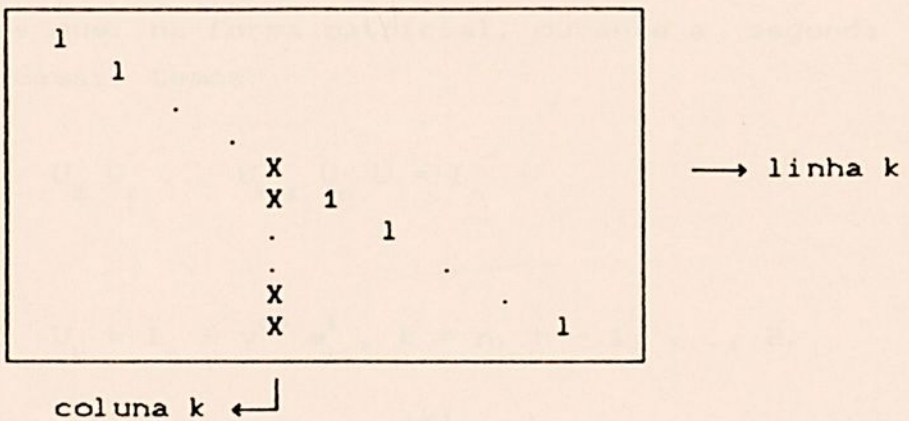
$$L_k = I_n + (c^{(k)} - e_k)\, e_k^t$$

sendo que os elementos $c_i^{(k)}$ do vetor coluna $c^{(k)}$, são definidos como:

$$c_i^{(k)} = \begin{cases} 0, & \text{se } i < k, \\ 1/a_{kk}^{(k)}, & \text{se } i = k \\ -a_{ik}^{(k)}/a_{kk}^{(k)}, & \text{se } i > k. \end{cases}$$

Assim L_k é uma matriz que possui 1's na diagonal principal exceto

na k-ésima coluna, que é formada pelos elementos $c_i^{(k)}$, sobre e abaixo da diagonal. Os demais elementos de L_k são nulos. Ou seja, no passo k a matriz L_k tem a forma:



De (2.2.1), podemos escrever

$$A^{(n+1)} = L_n L_{n-1} \ldots L_2 L_1 A^{(1)}$$

Fazendo

$$L = L_n L_{n-1} \ldots L_2 L_1 \qquad (2.2.2)$$

e como

$$A^{(1)} = A \qquad e \qquad A^{(n+1)} = U,$$

obtemos

$$U = LA. \qquad (2.2.3)$$

Então a primeira etapa da eliminação de Gauss consiste em encontrar uma matriz triangular inferior L que transforma A numa matriz triangular superior U.

De (2.2.3) e do fato de que as matrizes L_k são aplicadas em ambos os lados de (2.1.1) concluímos que

$$Ux = Lb. \qquad (2.2.4)$$

A segunda etapa da eliminação consiste na resolução da equação (2.2.4). Ela é feita como segue:

Seja x_i o i-ésimo elemento do vetor coluna x. Então seu último elemento x_n é igual ao último elemento de Lb, pois a última linha de U

é formada por zeros, exceto o elemento u_{nn} que é igual a 1. O valor de x_n é substituído na $(n-1)^{a}$ equação e x_{n-1} é facilmente calculado. Os valores de x_n e x_{n-1} são substituídos na $(n-2)^{a}$ equação de (2.2.4) e produzem x_{n-2}, e, assim em diante.

Observamos que, na forma matricial, durante a segunda etapa da eliminação de Gauss, temos:

$$U_2 U_3 \ldots U_{n-1} U_n U = I_n \qquad (2.2.5)$$

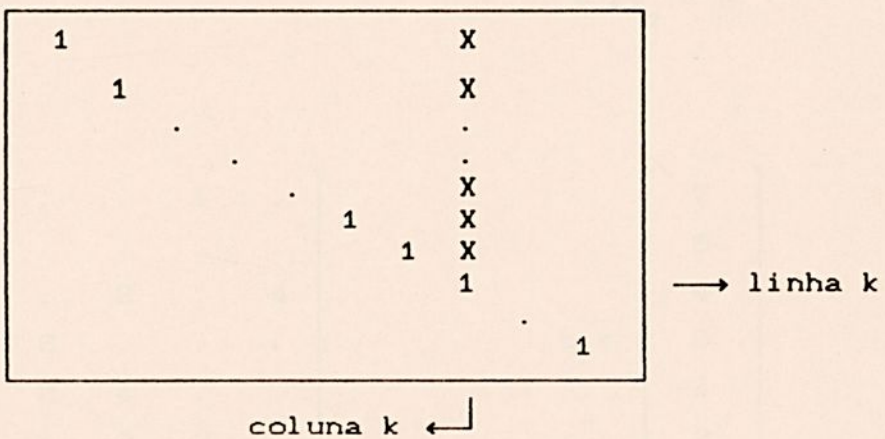
onde

$$U_k = I_n + v^{(k)} e_k^t, \quad k = n, n-1, \ldots, 2, \qquad (2.2.6)$$

com os elementos do vetor coluna $v^{(k)}$ dados por

$$v_i^{(k)} = \begin{cases} -a_{ik}^{(i+1)}, & \text{se } i < k, \\ 0, & \text{se } i \geq k. \end{cases} \qquad (2.2.7)$$

Então U_k possui 1's na diagonal principal e $v_i^{(k)}$ acima da diagonal principal, na k-ésima coluna. Todos os demais elementos de U_k são nulos. A matriz U_k tem, portanto, a forma:



Agora, podemos descrever o efeito total das duas etapas do método de eliminação sobre A.

De (2.2.5) obtemos

$$U^{-1} = U_2 U_3 \ldots U_{n-1} U_n, \qquad (2.2.8)$$

e de (2.2.4), (2.2.2) e (2.2.8) segue que

$$x = U^{-1} Lb$$
$$= U_2 U_3 \ldots U_{n-1} U_n L_n L_{n-1} \ldots L_2 L_1 b. \quad (2.2.9)$$

A seguir, apresentamos um exemplo que mostra, passo a passo, a resolução de um sistema linear por este método. Aqui e nos outros exemplos, para facilitar, trabalharemos com todos os elementos, inclusive os nulos. Mais à frente, mostraremos como deve ser feito o armazenamento e o uso computacional do método.

Exemplo: Na resolução do sistema linear

$$\begin{cases} 4x_1 + 2x_2 + x_6 = 7 \\ 2x_1 + 4x_2 = 6 \\ 2x_3 + 2x_5 + 4x_7 = 4 \\ 2x_4 = 0 \\ 2x_4 + 4x_5 = -4 \\ 2x_2 + 2x_5 + 2x_6 = 2 \\ 2x_1 + 4x_7 = 6 \end{cases}$$

Obtemos

$$A = \begin{pmatrix} 4 & 2 & . & . & . & 1 & . \\ 2 & 4 & . & . & . & . & . \\ . & . & 2 & . & 2 & . & 4 \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ 2 & . & . & . & . & . & 4 \end{pmatrix} \qquad b = \begin{pmatrix} 7 \\ 6 \\ 4 \\ 0 \\ -4 \\ 2 \\ 6 \end{pmatrix}$$

Primeira Etapa: $A^{(1)} = A$; $A^{(2)} = L_1 A^{(1)}$

e de (2.2.4), (2.2.2) e (2.2.8) segue que

$$x = U^{-1} Lb$$
$$= U_2 U_3 \ldots U_{n-1} U_n L_n L_{n-1} \ldots L_2 L_1 b. \quad (2.2.9)$$

A seguir, apresentamos um exemplo que mostra, passo a passo, a resolução de um sistema linear por este método. Aqui e nos outros exemplos, para facilitar, trabalharemos com todos os elementos, inclusive os nulos. Mais à frente, mostraremos como deve ser feito o armazenamento e o uso computacional do método.

Exemplo1: Na resolução do sistema linear

$$\begin{cases} 4x_1 + 2x_2 + x_6 = 7 \\ 2x_1 + 4x_2 = 6 \\ 2x_3 + 2x_5 + 4x_7 = 4 \\ 2x_4 = 0 \\ 2x_4 + 4x_5 = -4 \\ 2x_2 + 2x_5 + 2x_6 = 2 \\ 2x_1 + 4x_7 = 6 \end{cases}$$

Obtemos

$$A = \begin{pmatrix} 4 & 2 & . & . & . & 1 & . \\ 2 & 4 & . & . & . & . & . \\ . & . & 2 & . & 2 & . & 4 \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ 2 & . & . & . & . & . & 4 \end{pmatrix} \qquad b = \begin{pmatrix} 7 \\ 6 \\ 4 \\ 0 \\ -4 \\ 2 \\ 6 \end{pmatrix}$$

Primeira Etapa: $A^{(1)} = A$; $A^{(2)} = L_1 A^{(1)}$

$$C^{(1)} = \begin{pmatrix} 1/4 \\ -1/2 \\ . \\ . \\ . \\ . \\ -1/2 \end{pmatrix} \qquad L_1 = \begin{pmatrix} 1/4 & . & . & . & . & . & . \\ -1/2 & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ -1/2 & . & . & . & . & . & 1 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 1 & 1/2 & . & . & . & 1/4 & . \\ . & 3 & . & . & . & -1/2 & . \\ . & . & 2 & . & 2 & . & 4 \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ . & -1 & . & . & . & -1/2 & 4 \end{pmatrix}$$

$$C^{(2)} = \begin{pmatrix} . \\ 1/3 \\ . \\ . \\ . \\ -2/3 \\ 1/3 \end{pmatrix} \qquad L_2 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1/3 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & -2/3 & . & . & . & 1 & . \\ . & 1/3 & . & . & . & . & 1 \end{pmatrix}$$

$$A^{(3)} = \begin{pmatrix} 1 & 1/2 & . & . & . & 1/4 & . \\ . & 1 & . & . & . & -1/6 & . \\ . & . & 2 & . & 2 & . & 4 \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & . & . & . & 2 & 7/3 & . \\ . & . & . & . & . & -2/3 & 4 \end{pmatrix}$$

$$C^{(3)} = \begin{pmatrix} . \\ . \\ 1/2 \\ . \\ . \\ . \\ . \end{pmatrix} \qquad L_3 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1/2 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

$$A^{(4)} = \begin{pmatrix} 1 & 1/2 & . & . & . & 1/4 & . \\ . & 1 & . & . & . & -1/6 & . \\ . & . & 1 & . & 1 & . & 2 \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & . & . & . & 2 & 7/3 & . \\ . & . & . & . & . & -2/3 & 4 \end{pmatrix}$$

$$C^{(4)} = \begin{pmatrix} . \\ . \\ . \\ 1/2 \\ -1 \\ . \\ . \end{pmatrix} \qquad L_4 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1/2 & . & . & . \\ . & . & . & -1 & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

$$A^{(5)} = \begin{pmatrix} 1 & 1/2 & . & . & . & 1/4 & . \\ . & 1 & . & . & . & -1/6 & . \\ . & . & 1 & . & 1 & . & 2 \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 4 & . & . \\ . & . & . & . & 2 & 7/3 & . \\ . & . & . & . & . & -2/3 & 4 \end{pmatrix}$$

$$C^{(5)} = \begin{pmatrix} . \\ . \\ . \\ . \\ 1/4 \\ -1/2 \\ . \end{pmatrix} \qquad L_5 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1/4 & . & . \\ . & . & . & . & -1/2 & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

$$A^{(6)} = \begin{pmatrix} 1 & 1/2 & . & . & . & 1/4 & . \\ . & 1 & . & . & . & -1/6 & . \\ . & . & 1 & . & 1 & . & 2 \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 7/3 & . \\ . & . & . & . & . & -2/3 & 1 \end{pmatrix}$$

$$C^{(6)} = \begin{pmatrix} . \\ . \\ . \\ . \\ . \\ 3/7 \\ 2/7 \end{pmatrix} \qquad L_5 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 3/7 & . \\ . & . & . & . & . & 2/7 & 1 \end{pmatrix}$$

$$A^{(7)} = \begin{pmatrix} 1 & 1/2 & . & . & . & 1/4 & . \\ . & 1 & . & . & . & -1/6 & . \\ . & . & 1 & . & 1 & . & 2 \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 4 \end{pmatrix}$$

$$C^{(7)} = \begin{pmatrix} . \\ . \\ . \\ . \\ . \\ . \\ 1/4 \end{pmatrix} \qquad L_7 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1/4 \end{pmatrix}$$

$$A^{(8)} = \begin{pmatrix} 1 & 1/2 & . & . & . & 1/4 & . \\ . & 1 & . & . & . & -1/6 & . \\ . & . & 1 & . & 1 & . & 2 \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix} = U$$

Segunda Etapa: Resolução de $Ux = Lb$

$$x = U^{-1} Lb$$

$$= U_2 U_3 \ldots U_7 L_7 \ldots L_1 b$$

$$U_2 = \begin{pmatrix} 1 & -1/2 & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix} \qquad U_3 = U_4 = I_7$$

$$U_5 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & -1 & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

$$U_6 = \begin{pmatrix} 1 & . & . & . & . & -1/4 & . \\ . & 1 & . & . & . & 1/6 & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

$$U_7 = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & . & -2 \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

Como $U_2\ U_3 \ldots U_7 = U^{-1}$, temos

$$U^{-1} = \begin{pmatrix} 1 & -1/2 & . & . & . & -1/3 & . \\ . & 1 & . & . & . & 1/6 & . \\ . & . & 1 & . & -1 & . & -2 \\ . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . \\ . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

Como $U_2 \ldots U_7\ L_7 \ldots L_1 = A^{-1}$, temos

$$A^{-1} = \begin{pmatrix} 2/7 & -1/14 & . & -1/14 & 1/14 & -1/7 & . \\ -1/7 & 2/7 & . & 1/28 & -1/28 & 1/14 & . \\ 2/7 & -1/14 & 1/2 & 5/28 & -5/28 & -1/17 & -1/2 \\ . & . & . & 1/2 & . & . & . \\ . & . & . & -1/14 & 1/4 & . & . \\ 1/7 & -2/7 & . & 3/14 & -3/14 & 3/7 & . \\ -1/7 & 1/28 & . & 1/28 & -1/28 & 1/14 & 1/4 \end{pmatrix}$$

Como $x = A^{-1}b$, a solução do sistema é

$$x = (1 \quad 1 \quad 1 \quad 0 \quad -1 \quad 1 \quad 1)^t.$$

2.3. PIVOTAMENTO E ERROS DE ARREDONDAMENTO

O elemento $a_{kk}^{(k)}$ é denominado pivô do k-ésimo passo da eliminação de Gauss.

Como os cálculos são realizados com um número finito de dígitos significativos, erros de arredondamento são geralmente introduzidos. Em 1965, Wilkinson apresentou duas técnicas que minimizam os efeitos destes erros na eliminação de Gauss, denominadas pivotamento parcial e pivotamento total.

O pivotamento parcial é realizado da seguinte forma: a cada passo k é escolhido o elemento de maior valor absoluto dentre os elementos da k-ésima coluna de $A^{(k)}$ situados sobre ou abaixo da k-ésima linha, ou seja, o pivô do k-ésimo passo é:

$$\left|a_{lk}^{(k)}\right| = \max_{k \le i \le n} \left|a_{ik}^{(k)}\right|$$

e então, é feita a permutação das linhas l e k de $A^{(k)}$ antes do k-ésimo passo em (2.2.1).

O pivotamento total é assim realizado: a cada passo k, o elemento de maior valor absoluto é escolhido a partir das últimas (n-k+1) linhas e colunas de $A^{(k)}$; isto é:

$$\left|a_{lt}^{(k)}\right| = \max_{k\leq i,\, j\leq n} \left|a_{ij}^{(k)}\right|$$

e então, as linhas l e k e as colunas t e k de $A^{(k)}$ são permutadas, antes do k-ésimo passo de (2.2.1).

É claro que o uso de qualquer destas técnicas exige que as permutações realizadas sejam salvas para uso posterior no reordenamento da solução.

O pivotamento total não é recomendado, pois exige um elevado número de comparações entre os elementos para que a cada passo, se possa determinar o maior elemento em módulo que será usado como pivô.

Na maioria das aplicações práticas envolvendo grandes matrizes esparsas, em vez de pivotamento parcial ou total, em geral é suficiente assegurarmos que todos os pivôs sejam maiores que um certo número ε, pré-fixado, chamado Pivô de Tolerância. (Tewarson, 1969). Este procedimento é realizado, por exemplo, em programação linear (Clasen, 1966) onde as matrizes são geralmente grandes e esparsas. O pivô de tolerância evita a escolha de pivôs tão pequenos que conduzam à erros de arredondamento.

Na prática, para a maioria dos problemas que envolvam grandes matrizes esparsas, $\varepsilon = 10^{-9}$ tem sido satisfatório quando os cálculos são feitos utilizando-se nove ou dez dígitos decimais significativos.

2.4. A ELIMINAÇÃO NA FORMA DE INVERSA (EFI)

A solução de $Ax = b$, para um b arbitrário é dada por $x = A^{-1}b$. Comparando-a com a solução dada em (2.2.9), concluímos que

$$A^{-1} = U_2 \ldots U_{n-1} U_n L_n L_{n-1} \ldots L_2 L_1, \qquad (2.4.1)$$

que é denominada Eliminação na Forma de Inversa (EFI). Assim, A^{-1} é escrita numa forma fatorada como o produto de n matrizes triangulares inferiores e n-1 matrizes triangulares superiores.

Uma das principais vantagens da EFI é a facilidade com que a matriz pode ser pré e pós multiplicada por um vetor, como veremos abaixo.

Seja q_i o i-ésimo elemento de um vetor coluna q. Das definições de U_k e $v^{(k)}$, temos

$$\begin{aligned} e_i^t(U_k\, q) &= q_i + v_i^{(k)} q_k \quad , \; i < k; \\ e_i^t(U_k\, q) &= q_i \quad , \; i \geq k. \end{aligned} \qquad (2.4.2)$$

Assim, a pré multiplição de um vetor coluna por U_k é equivalente a adicionar ao vetor coluna dado seu k-ésimo elemento multiplicado por $v^{(k)}$. Além disso, temos que

$$\begin{aligned} (q^t\, U_k)\, e_i &= q_i \quad , \quad i \neq k; \\ (q^t\, U_k)\, e_k &= q^t\, v^{(k)} + q_k \, , \end{aligned} \qquad (2.4.3)$$

e portanto, a pós multiplicação do vetor linha q^t por U_k deixa todos os elementos de q^t iguais exceto o k-ésimo elemento ao qual é adicionado o produto interno de q^t por $v^{(k)}$.

Das definições de L_k e $c_i^{(k)}$, temos

$$\begin{aligned} e_i^t(L_k\, q) &= q_i && , \ i < k; \\ e_k^t(L_k\, q) &= c_k^{(k)}\, q_k && , \ i = k; \\ e_i^t(L_k\, q) &= c_i^{(k)}\, q_k + q_i && , \ i > k \end{aligned} \qquad (2.4.4)$$

Então $L_k q$ pode ser obtida de q trocando-se seu k-ésimo elemento por zero e então adicionando-se $q_k\, c^{(k)}$.

Temos também que

$$\begin{aligned} (q^t L_k)\, e_i &= q_i^t && , \ i \neq k; \\ (q^t L_k)\, e_k &= q^t\, c^{(k)} && , \end{aligned} \qquad (2.4.5)$$

e portanto, a pós multiplicação de um vetor linha por L_k conserva iguais todos os elementos do vetor, exceto seu k-ésimo elemento que é trocado pelo produto interno entre o vetor e $c^{(k)}$.

O cálculo de $A^{-1}y$ onde y é um vetor coluna e A^{-1} é uma EFI dada por (2.4.1) pode ser rearranjado de modo que nos n primeiros passos se use (2.4.4) e nos últimos n-1 passos se use (2.4.2). O vetor coluna formado à cada passo é usado no próximo passo da seguinte forma:

$$A^{-1}\, y = U_2\,(\ldots\,(U_n\,(L_n\,\ldots\,(L_2\,(L_1 y)\,\ldots).$$

Similarmente, yA^{-1} pode ser calculado usando (2.4.3) e (2.4.5); e, o vetor linha formado a cada passo é usado no próximo, isto é,

$$yA^{-1} = (\,\ldots(y\,U_2)\,U_3)\,\ldots\,U_n)\,L_n)\,\ldots\,L_1).$$

Agora, de L_k, $c_i^{(k)}$, U_k, $v_i^{(k)}$ e A^{-1}, é evidente que ao calcularmos

a EFI necessitaremos somente dos elementos não nulos dos $c^{(k)}$'s e $v^{(k)}$'s. Por isso, apenas estes elementos acompanhados das informações de posições relevantes serão armazenados.

Como a memória de um computador é escassa para grandes matrizes esparsas, é importante que sua EFI seja determinada de modo que o armazenamento seja minimizado. Em outras palavras, o número de elementos não nulos de todos os $c^{(k)}$'s e $v^{(k)}$'s devem ser minimizados. A seguir, mostraremos como isso pode ser feito.

2.5. MINIMIZAÇÃO DO NÚMERO TOTAL DE ELEMENTOS NÃO NULOS NA EFI

Em cada estágio do método de eliminação de Gauss, múltiplos de uma linha são subtraídos de várias outras linhas da matriz. Por isso este processo geralmente torna não nulo um elemento que no estágio anterior era nulo. Por exemplo, se os elementos $a_{kk}^{(k)}$, $a_{kj}^{(k)}$, $a_{ik}^{(k)}$ são não nulos e $a_{ij}^{(k)} = 0$ para $i, j > k$, então no final do k-ésimo estágio teremos:

$$a_{ij}^{(k+1)} = - \left(a_{ij}^{(k)} / a_{kk}^{(k)} \right) a_{kj}^{(k)} ,$$

que é claramente não nulo. Assim, neste exemplo, uma posição (i,j) que tinha um elemento nulo em $A^{(k)}$ passou a ter um elemento não nulo em $A^{(k+1)}$.

Ao número total de elementos nulos em $A^{(k)}$ que se tornam não nulos em $A^{(k+1)}$ denominamos "FILL-IN" local (preenchimento local).

Podemos escolher para pivô do k-ésimo estágio qualquer elemento não nulo de $A^{(k)}$. Como já vimos, se quisermos escolher

$a_{lt}^{(k)}$, $l \geq k$, $t \geq k$, para pivô, devemos fazer a permutação das linhas k e l e das colunas k e t de $A^{(k)}$, obtendo assim $\hat{A}^{(k)}$, antes de determinarmos $A^{(k+1)}$.

Dessa forma, no lugar de $A^{(k+1)} = L_k A^{(k)}$, teremos

$$A^{(k+1)} = L_k \hat{A}^{(k)} , \quad k = 1, 2, \ldots, n, \tag{2.5.1}$$

onde

$$\hat{A}^{(k)} = P_k A^{(k)} Q_k \tag{2.5.2}$$

sendo P_k e Q_k matrizes obtidas da I_n pela troca das linhas k e l e das colunas k e t, respectivamente.

As matrizes L_k são as mesmas vistas anteriormente, mas os $c_i^{(k)}$'s serão dados por:

$$C_i^{(k)} = \begin{cases} 0 & , i < k; \\ 1 / \hat{a}_{kk}^{(k)} & , i = k; \\ -\hat{a}_{ik}^{(k)} / \hat{a}_{kk}^{(k)} & , i > k, \end{cases} \tag{2.5.3}$$

onde $\hat{a}_{ij}^{(k)}$ é o elemento (i, j) de $\hat{A}^{(k)}$.

Devemos escolher para pivô o elemento que minimize o fill-in local, lembrando que o valor absoluto do elemento escolhido deve sempre ser maior que o pivô de tolerância ε descrito anteriormente. Isto pode ser feito como segue.

Definição (2.5.4): Seja B_k a matriz obtida das últimas n-k+1 linhas e colunas de $A^{(k)}$ pela troca dos elementos não nulos pela unidade.

Veremos agora um teorema (Tewarson, 1967b) que pode ser usado

para determinar o pivô que torna menor o fill-in.

Teorema (2.5.5): Se $a^{(k)}_{i+k-1,\,j+k-1}$ é escolhido como pivô do k-ésimo passo da primeira etapa da eliminação de Gauss, então o fill-in local é dado pelo elemento (i,j) da matriz F_k, com

$$F_k = B_k \bar{B}_k^t B_k ,$$

onde $\bar{B}_k^t$ é a transposta da matriz que resulta quando cada elemento nulo de B_k é trocado pela unidade e vice-versa.

Demonstração: Se em $A^{(k)}$ o elemento (p+k-1 , q+k-1) é nulo mas os elementos (i+k-1 , q+k-1) e (p+k-1 , j+k-1) não são nulos, então o elemento (p+k-1 , q+k-1) de $A^{(k+1)}$ será não nulo.

Isto significa que se $b^{(k)}_{pq} = 0$ e $b^{(k)}_{iq} = b^{(k)}_{pj} = 1$ onde $b^{(k)}_{pq}$ é o elemento (p,q) de B_k, então um novo elemento não nulo é criado. Denotando-se o fill-in local no k-ésimo passo por $f^{(k)}_{ij}$ então

$$f^{(k)}_{ij} = \sum_p \sum_q b^{(k)}_{iq} \,(1-b^{(k)}_{pq})\, b^{(k)}_{pj} \;,\; p \neq i,\; q \neq j.$$

Abandonando a restrição $p \neq i$, $q \neq j$, já que para $p = i$ temos $b^{(k)}_{iq} = 0$ ou $1-b^{(k)}_{pq} = 0$ e para $q = j$ teremos $1-b^{(k)}_{pq} = 0$ ou $b^{(k)}_{pj} = 0$, então

$$f^{(k)}_{ij} = \sum_p \sum_q e_i^t B_k e_q \,(1-e_p^t B_k e_q)\, e_p^t B_k e_j .$$

Agora, se M é uma matriz quadrada de ordem (n-k+1) com todos os elementos iguais a 1, então:

$$1 - e_p^t B_k e_q = e_p^t M e_q - e_p^t B_k e_q = e_p^t (M - B_k) e_q$$

$$= e_p^t \bar{B}_k e_q = e_q^t \bar{B}_k^t e_p$$

onde usamos o fato de que $e_p^t \bar{B}_k e_q$ é um escalar. Temos, então

$$f_{ij}^{(k)} = \sum_p \sum_q e_i^t B_k e_q e_q^t \bar{B}_k^t e_p e_p^t B_k e_j$$

$$= e_i^t B_k \sum_q e_q e_q^t \bar{B}_k^t \sum_p e_p e_p^t B_k e_j$$

$$= e_i^t B_k \bar{B}_k^t B_k e_j ,$$

pois $\sum_q e_q e_q^t = \sum_p e_p e_p^t = I_{n-k+1}$.

Isto completa a demonstração do teorema.

Agora, podemos escolher um pivô que minimiza o fill-in local. Ele é dado pelo corolário seguinte, demonstrado por Tewarson, 1973.

Corolário (2.5.6): Se no k-ésimo estágio da eliminação de Gauss, $a_{lt}^{(k)}$ é escolhido como pivô, onde $l = \alpha + k - 1$ e $t = \beta + k - 1$ com α, β dados por

$$f_{\alpha\beta}^{(k)} = \min_{i,j} e_i^t F_k e_j \quad \text{para todo} \quad |a_{i+k-1,j+k-1}^{(k)}| > \varepsilon,$$

sendo ε um pivô de tolerância convenientemente escolhido, então o fill-in local torna-se mínimo.

A segunda etapa da Eliminação de Gauss não cria novos elementos

não nulos. Estes são criados apenas, como já vimos, na primeira etapa.

No final do k-ésimo passo, em geral, $A^{(k+1)}$ possui nas últimas $(n - k)$ linhas e colunas, alguns elementos não nulos que eram nulos em $A^{(k)}$. Como, nos estágios seguintes estas linhas e colunas são utilizadas para calcular os elementos de $c^{(k)}$ e $v^{(k)}$ então, minimizando o fill-in local estaremos minimizando o número de elementos não nulos nos $c^{(k)}$'s e $v^{(k)}$'s, supondo que o mínimo local conduza ao mínimo global. Isto pode ser verdadeiro para algumas matrizes, mas não o é para matrizes esparsas arbitrárias. No entanto, a minimização do crescimento dos elementos não nulos locais, pelo uso do Corolário (2.5.6) acarreta uma razoável diminuição no número de elementos não nulos criados nos vetores $c^{(k)}$'s e $v^{(k)}$'s.

A escolha do pivô $a_{lt}^{(k)}$ que minimiza o fill-in local não acarreta maiores problemas.

As mudanças relevantes podem ser descritas como: De (2.5.1) e (2.5.2), temos

$$A^{(n+1)} = L_n P_n \ldots L_2 P_2 L_1 P_1 A \; Q_1 Q_2 \ldots Q_n . \qquad (2.5.7)$$

Fazendo

$$\hat{L} = L_n P_n \ldots L_2 P_2 L_1 P_1 , \qquad (2.5.8)$$

$$Q = Q_1 \ldots Q_n \quad e \quad A^{(n+1)} = \hat{U} ,$$

obtemos

$$A^{-1} = Q \hat{U}^{-1} \hat{L}. \qquad (2.5.9)$$

Observamos que o armazenamento das matrizes de permutação P_k e Q_k

exige n posições de memória e não n^2, pois de cada uma delas, somente as posições dos elementos não nulos precisam ser armazenadas.

Exemplo 2: Minimização do fill-in local para o cálculo da EFI usando o Teorema (2.5.5) e o Corolário (2.5.6).

Seja A a matriz dos coeficientes do sistema linear do exemplo 1.

$$A = \begin{pmatrix} 4 & 2 & . & . & . & 1 & . \\ 2 & 4 & . & . & . & . & . \\ . & . & 2 & . & 2 & . & 4 \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ 2 & . & . & . & . & . & 4 \end{pmatrix}$$

Então no primeiro estágio teremos:

Passo 1. Determinação de B_1

Dimensão de $B_1 = n - k + 1 = 7$

$$B_1 = \begin{pmatrix} 1 & 1 & . & . & . & 1 & . \\ 1 & 1 & . & . & . & . & . \\ . & . & 1 & . & 1 & . & 1 \\ . & . & . & 1 & . & . & . \\ . & . & . & 1 & 1 & . & . \\ . & 1 & . & . & 1 & 1 & . \\ 1 & . & . & . & . & . & 1 \end{pmatrix} \qquad B_1 \overrightarrow{B_1} B_1 = \begin{pmatrix} 3 & 2 & 3 & 6 & 7 & 1 & 5 \\ 1 & 1 & 2 & 4 & 5 & 1 & 3 \\ 8 & 8 & . & 5 & 4 & 5 & 2 \\ 3 & 3 & 1 & . & 2 & 2 & 2 \\ 6 & 5 & 1 & 1 & 2 & 3 & 3 \\ 6 & 3 & 2 & 5 & 4 & 1 & 5 \\ 2 & 4 & 1 & 4 & 5 & 3 & 1 \end{pmatrix}$$

Minimizaremos os cálculos se escolhermos para pivô $a_{33}^{(1)}$ ou $a_{44}^{(1)}$.

Se escolhermos $a_{33}^{(1)}$ para pivô, devemos permutar as linhas e colunas 1 e 3 respectivamente.

Então obtemos:

$$\hat{A}^{(1)} = \begin{pmatrix} 2 & . & . & . & 2 & . & 4 \\ . & 4 & 2 & . & . & . & . \\ . & 2 & 4 & . & . & 1 & . \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ . & . & 2 & . & . & . & 4 \end{pmatrix} \qquad C^{(1)} = \begin{pmatrix} 1/2 \\ . \\ . \\ . \\ . \\ . \\ . \end{pmatrix}$$

e portanto

$$A^{(2)} = L_1 \hat{A}^{(1)} = \begin{pmatrix} 1 & . & . & . & 1 & . & 2 \\ . & 4 & 2 & . & . & . & . \\ . & 2 & 4 & . & . & 1 & . \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ . & . & 2 & . & . & . & 4 \end{pmatrix}$$

Passo 2. Determinação de B_2

Dimensão de $B_2 = 7 - 2 + 1 = 6$

$$B_2 = \begin{pmatrix} 1 & 1 & . & . & . & . \\ 1 & 1 & . & . & 1 & . \\ . & . & 1 & . & . & . \\ . & . & 1 & 1 & . & . \\ 1 & . & . & 1 & 1 & . \\ . & 1 & . & . & . & 1 \end{pmatrix} \qquad B_2 \, \overline{B}_2^t \, B_2 = \begin{pmatrix} 1 & 1 & 4 & 3 & 1 & 1 \\ 2 & 3 & 6 & 4 & 1 & 2 \\ 3 & 3 & . & 1 & 2 & 1 \\ 5 & 6 & . & 1 & 3 & 2 \\ 3 & 6 & 5 & 2 & 1 & 3 \\ 4 & 2 & 4 & 4 & 3 & . \end{pmatrix}$$

Escolhemos $a_{44}^{(2)}$ para pivô. Então

$$\hat{A}_2 = \begin{pmatrix} 1 & . & . & . & 1 & . & 2 \\ . & 2 & . & . & . & . & . \\ . & . & 4 & 2 & . & 1 & . \\ . & . & 2 & 4 & . & . & . \\ . & 2 & . & . & 4 & . & . \\ . & . & . & 2 & 2 & 2 & . \\ . & . & 2 & . & . & . & 4 \end{pmatrix} \qquad C^{(2)} = \begin{pmatrix} 1 \\ 1/2 \\ . \\ . \\ -1 \\ . \\ . \end{pmatrix}$$

e portanto

$$A^{(3)} = \begin{pmatrix} 1 & . & . & . & 1 & . & 2 \\ . & 1 & . & . & . & . & . \\ . & . & 4 & 2 & . & 1 & . \\ . & . & 2 & 4 & . & . & . \\ . & . & . & . & 4 & . & . \\ . & . & . & 2 & 2 & 2 & . \\ . & . & 2 & . & . & . & 4 \end{pmatrix}$$

Passo 3. Determinação de B_3

Dimensão de $B_3 = 7 - 3 + 1 = 5$

$$B_3 = \begin{pmatrix} 1 & 1 & . & 1 & . \\ 1 & 1 & . & . & . \\ . & . & 1 & . & . \\ . & 1 & 1 & 1 & . \\ 1 & . & . & . & 1 \end{pmatrix} \qquad B_3 \, \overline{B}_3^t \, B_3 = \begin{pmatrix} 3 & 2 & 4 & 1 & 2 \\ 1 & 1 & 3 & 1 & 1 \\ 3 & 2 & . & . & 1 \\ 6 & 3 & 2 & 1 & 3 \\ 2 & 4 & 4 & 3 & . \end{pmatrix}$$

Escolhemos $a_{55}^{(3)}$ para pivô. Temos

$$\hat{A}_3 = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 4 & . & . & . & . \\ . & . & . & 4 & 2 & . & . \\ . & . & 2 & 2 & 4 & . & 4 \\ . & . & 2 & 2 & . & 2 & . \\ . & . & . & . & 2 & . & 4 \end{pmatrix} \qquad C^{(3)} = \begin{pmatrix} . \\ . \\ 1/4 \\ . \\ -1/2 \\ -1/2 \\ . \end{pmatrix}$$

e então

$$A^{(4)} = \begin{pmatrix} 1 & . & . & . & . & . & 2 \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & . & 2 & . & 4 \\ . & . & . & 2 & 4 & 1 & . \\ . & . & . & 2 & . & 2 & . \\ . & . & . & 4 & 2 & . & 4 \end{pmatrix}$$

Passo 4. Determinação de B_4

Dimensão de $B_4 = 7 - 4 + 1 = 4$

$$B_4 = \begin{pmatrix} 1 & 1 & . & . \\ 1 & 1 & 1 & . \\ 1 & . & 1 & . \\ . & 1 & . & 1 \end{pmatrix} \qquad B_4 \, \overline{B}_4^t \, B_4 = \begin{pmatrix} 1 & 1 & . & 1 \\ 2 & 3 & 1 & 2 \\ 1 & 2 & . & 1 \\ 4 & 2 & 3 & . \end{pmatrix}$$

Escolhemos $a_{77}^{(4)}$ para pivô. Temos então

$$\hat{A}_4 = \begin{pmatrix} 1 & . & 1 & 2 & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 4 & 2 & . & . \\ . & . & . & . & 4 & 1 & 2 \\ . & . & . & . & . & 2 & 2 \\ . & . & . & 4 & 2 & . & 4 \end{pmatrix} \qquad C^{(4)} = \begin{pmatrix} . \\ . \\ . \\ 1/4 \\ . \\ . \\ -1 \end{pmatrix}$$

e portanto

$$A^{(5)} = \begin{pmatrix} 1 & . & 1 & 2 & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 4 & 1 & 2 \\ . & . & . & . & . & 2 & 2 \\ . & . & . & . & . & . & 4 \end{pmatrix}$$

Passo 5. Determinação de B_5

Dimensão de $B_5 = 7 - 5 + 1 = 3$

$$B_5 = \begin{pmatrix} 1 & 1 & 1 \\ . & 1 & 1 \\ . & . & 1 \end{pmatrix} \qquad B_5 \bar{B}_5^t B_5 = \begin{pmatrix} . & 1 & 3 \\ . & . & 1 \\ . & . & . \end{pmatrix}$$

Não há necessidade de efetuarmos permutações pois escolhemos $a_{55}^{(5)}$ para pivô. Temos então

$$\hat{A}_5 = A^{(5)} \qquad C^{(5)} = \begin{pmatrix} . \\ . \\ . \\ 1/4 \\ . \\ . \\ . \end{pmatrix}$$

e portanto

$$A^{(6)} = \begin{pmatrix} 1 & . & 1 & 2 & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 1 & 1/4 & 1/2 \\ . & . & . & . & . & 2 & 2 \\ . & . & . & . & . & . & 4 \end{pmatrix}$$

Passo 6. Determinação de B_6

Dimensão de $B_6 = 7 - 6 + 1 = 2$

$$B_6 = \begin{pmatrix} 1 & 1 \\ . & 1 \end{pmatrix} \qquad B_6 \bar{B}_6^t B_6 = \begin{pmatrix} . & . \\ . & . \end{pmatrix}$$

$$\hat{A}_6 = A^{(6)} \qquad C^{(6)} = \begin{pmatrix} . \\ . \\ . \\ . \\ . \\ 1/2 \\ . \end{pmatrix}$$

e portanto

$$A^{(7)} = \begin{pmatrix} 1 & . & 1 & 2 & . & . & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 1 & 1/4 & 1/2 \\ . & . & . & . & . & 1 & 1 \\ . & . & . & . & . & . & 4 \end{pmatrix} \qquad C^{(7)} = \begin{pmatrix} . \\ . \\ . \\ . \\ . \\ . \\ 1/4 \end{pmatrix}$$

Passo 8. Determinação de B_7

Dimensão de $B_7 = 7 - 7 + 1 = 1$

$B_7 = (1)$ $\qquad B_7 \bar{B}_7^t B_7 = (0)$

Então

$$A^{(8)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 1 & 1/4 & 1/2 \\ . & . & . & . & . & 1 & 1 \\ . & . & . & . & . & . & 1 \end{pmatrix}$$

2° estágio: Determinação da forma fatorada de A^{-1}

Temos: $\qquad A^{-1} = Q\,\hat{U}^{-1}\,\hat{L}$

onde $\qquad Q = Q_1\,Q_2 \ldots Q_7$

$$\hat{L} = L_7\,P_7\,L_6\,P_6 \ldots L_1\,P_1,$$

$$A^{(8)} = \hat{U}$$

Então

$$A^{-1} = \begin{pmatrix} 2/7 & -1/14 & . & -1/14 & 1/14 & -1/7 & . \\ -1/7 & 2/7 & . & 1/28 & -1/28 & 1/14 & . \\ 2/7 & -1/14 & 1/2 & 5/28 & -5/28 & -1/7 & 1/2 \\ . & . & . & 1/2 & . & . & . \\ . & . & . & -1/4 & 1/4 & . & . \\ 1/7 & -2/7 & . & 3/14 & -3/14 & 3/7 & . \\ -1/7 & 1/28 & . & 1/28 & -1/28 & 1/14 & 1/4 \end{pmatrix}.$$

2.6. ESTRATÉGIA DE MARKOWITZ

Um modo mais simples, mas menos preciso, para determinarmos o pivô que minimiza o fill-in local foi dado por Markowitz.

Assumiremos que os (k-1) primeiros passos da eliminação de Gauss já tenham sido calculados e que agora procuramos encontrar o pivô do k-ésimo passo. Seja A_k a matriz composta pelas últimas (n-k+1) linhas e colunas de $A^{(k)}$. A matriz A_k é portanto uma submatriz de $A^{(k)}$ e é denominada parte ativa da matriz $A^{(k)}$. Seja r(i,k) o número de elementos não nulos da linha i de A_k e seja c(j,k) o número de elementos não nulos da coluna j de A_k.

O custo de Markowitz do elemento $a_{ij}^{(k)}$ é dado por:

$$M_{ijk} = (r(i,k) - 1)\,(c(j,k) - 1), \quad i, j, k = 1, 2, \ldots, n.$$

Então M_{ijk} é igual ao número máximo de elementos nulos da matriz $A^{(k)}$ que se tornam não nulos em $A^{(k+1)}$ caso $a_{ij}^{(k)}$ venha a ser escolhido para pivô. Ele é portanto, um limitante superior para o fill-in local no estágio k se $a_{ij}^{(k)}$ for escolhido para pivô. Assim, devemos escolher para pivô o elemento cujo custo seja

$$M_k = \min_{i,j,k = 1,\ldots,n} \left(M_{ijk} \right)$$

Observemos que isto não significa necessariamente que teremos minimizado o fill-in local no k-ésimo estágio.

Existem pelo menos duas desvantagens para a aplicação desta estratégia:

1) Se a ordem de A é muito grande, a pesquisa do custo de Markowitz envolverá muitos elementos;

2) Pode ocorrer instabilidade numérica .

Para preservar a estabilidade numérica não devemos aceitar como pivô elementos de $A^{(k)}$ com valores absolutos muito pequenos.

Um artifício usado para diminuir o número de elementos a serem pesquisados para a escolha do pivô que minimiza o fill-in é escolher apenas um certo número de linhas da submatriz A_k, selecionadas de modo que haja boa chance de diminuir o fill-in local.

Exemplo 4: Seja A a matriz dos coeficientes do sistema linear do exemplo 1.

O custo de Markowitz de cada elemento é dado por

$$M_{ijk} = (r(i, k) - 1) * (c(j, k) - 1).$$

No primeiro passo temos

$M_{111} = 4$ $M_{211} = 2$ $M_{931} = 0$ $M_{441} = 0$ $M_{621} = 4$ $M_{711} = 2$

$M_{121} = 4$ $M_{221} = 2$ $M_{951} = 4$ $M_{541} = 1$ $M_{651} = 4$ $M_{771} = 1$

$M_{161} = 2$ $M_{961} = 2$ $M_{551} = 1$ $M_{661} = 2$

Portanto $M_1 = 0$.

Então devemos escolher como pivô do 1º estágio $a_{33}^{(1)}$ ou $a_{44}^{(1)}$

Escolhendo $a_{33}^{(1)}$, temos

$$\hat{A}^{(1)} = \begin{pmatrix} 2 & . & . & . & 2 & . & 4 \\ . & 4 & 2 & . & . & . & . \\ . & 2 & 4 & . & . & 1 & . \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ . & . & 2 & . & . & . & 4 \end{pmatrix} \quad A^{(2)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 4 & 2 & . & . & . & . \\ . & 2 & 4 & . & 2 & . & 4 \\ . & . & . & 2 & . & . & . \\ . & . & . & 2 & 4 & . & . \\ . & 2 & . & . & 2 & 2 & . \\ . & . & 2 & . & . & . & 4 \end{pmatrix}$$

No segundo passo, temos

$M_{222} = 2$ $M_{322} = 6$ $M_{442} = 0$ $M_{542} = 1$ $M_{622} = 4$ $M_{732} = 2$

$M_{232} = 2$ $M_{332} = 0$ $M_{552} = 2$ $M_{652} = 4$ $M_{772} = 1$,

$M_{352} = 6$ $M_{662} = 2$

$M_{372} = 3$

Portanto $M_2 = 0$. Escolhendo $a_{44}^{(2)}$ para pivô, temos

$$\hat{A}^{(2)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 2 & . & . & . & . & . \\ . & . & 4 & 2 & 2 & . & 4 \\ . & . & 2 & 4 & . & . & . \\ . & 2 & . & . & 4 & . & . \\ . & . & . & 2 & 2 & 2 & . \\ . & . & 2 & . & . & . & 4 \end{pmatrix} \quad A^{(3)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 4 & 2 & 2 & . & 4 \\ . & . & 2 & 4 & . & . & . \\ . & . & . & . & 4 & . & . \\ . & . & . & 2 & 2 & 2 & . \\ & . & & . & . & . & 4 \end{pmatrix}$$

No terceiro passo, temos

$M_{333} = 6$ $M_{433} = 2$ $M_{553} = 0$ $M_{633} = 4$ $M_{733} = 2$

$M_{343} = 6$ $M_{443} = 2$ $M_{643} = 4$ $M_{733} = 1$

$M_{353} = 6$ $M_{653} = 2$

$M_{373} = 3$

Portanto $M_3 = 0$. Escolhemos $a_{55}^{(3)}$ para pivô. Então

$$\hat{A}^{(3)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 4 & . & . & . & . \\ . & . & . & 4 & 2 & . & . \\ . & . & . & 2 & 4 & . & 4 \\ . & . & . & 2 & . & 2 & . \\ . & . & . & . & 2 & . & 4 \end{pmatrix} \quad A^{(4)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 4 & 2 & . & . \\ . & . & . & 2 & 4 & . & 4 \\ . & . & . & 2 & . & 2 & . \\ . & . & . & . & 2 & . & 4 \end{pmatrix}$$

No quarto passo, temos

$M_{444} = 2$ $M_{544} = 4$ $M_{644} = 2$ $M_{754} = 2$

$M_{454} = 2$ $M_{554} = 4$ $M_{664} = 1$ $M_{774} = 1$

$M_{564} = 2$

Portanto $M_4 = 1$. Escolhemos para pivô $a_{77}^{(4)}$ e então

$$\hat{A}^{(4)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 4 & 2 & . & . \\ . & . & . & 4 & 4 & . & 2 \\ . & . & . & . & . & 2 & 2 \\ . & . & . & . & 2 & . & 4 \end{pmatrix} \quad A^{(5)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 2 & . & 2 \\ . & . & . & . & . & 2 & 2 \\ . & . & . & . & 2 & . & 4 \end{pmatrix}$$

No quinto passo, temos

$M_{555} = 1$ $M_{665} = 0$ $M_{755} = 2$

$M_{575} = 1$ $M_{755} = 2$

Portanto $M_5 = 0$. Escolhendo $a_{66}^{(5)}$ para pivô, temos

$$\hat{A}^{(5)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & . & 1/2 & . \\ . & . & . & . & 2 & . & 2 \\ . & . & . & . & . & 2 & 2 \\ . & . & . & . & . & 2 & 4 \end{pmatrix} \quad A^{(6)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 1 & . & 1 \\ . & . & . & . & . & 2 & 2 \\ . & . & . & . & . & . & 4 \end{pmatrix}$$

No sexto passo temos:

$M_{666} = 0 \quad M_{766} = 1$

$M_{776} = 0$

Portanto $M_6 = 0$. Escolhemos $a_{66}^{(6)}$ para pivô. Temos $\hat{A}^{(6)} = A^{(6)}$

Então $A^{(7)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 1 & . & 1 \\ . & . & . & . & . & 1 & 1 \\ . & . & . & . & . & . & 2 \end{pmatrix}$

No sétimo passo, temos: $M_{777} = 0$ Portanto $\hat{A}^{(7)} = A^{(7)}$ e então

$A^{(8)} = \begin{pmatrix} 1 & . & . & . & . & 1/2 & . \\ . & 1 & . & . & . & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1/2 & . & . \\ . & . & . & . & 1 & . & 1 \\ . & . & . & . & . & 1 & 1 \\ . & . & . & . & . & . & 1 \end{pmatrix}$

Observamos que uma das principais razões para escolhermos apenas os elementos da diagonal principal como pivôs é que se A é simétrica, então frequentemente somente a parte superior de A e seus elementos diagonais são armazenados e a escolha dos pivôs dentre os elementos diagonais preserva a simetria. Além disso, os $c^{(k)}$'s podem ser facilmente obtidos da matriz triangular superior obtida no fim da primeira etapa da eliminação de Gauss. Esta fato é estabelecido no teorema seguinte.

Teorema (2.6.1): Se A é simétrica e somente os elementos diagonais são escolhidos como pivôs, então:

(a) para k = 1, 2, ..., n-1, a matriz formada pelas últimas n-k linhas e colunas de $A^{(k+1)}$ em (2.5.1) também é simétrica; e,

(b) Em (2.5.3), os $c^{(k)}$'s são dados por

$$c_i^{(k)} = -a_{ki}^{(k+1)} = v_i^{(k)}, \quad i > k. \qquad (2.6.2)$$

Demonstração: Se podemos mostrar que $a_{ij}^{(k+1)} = a_{ji}^{(k+1)}$ para $i,j > k$ sempre que $a_{ij}^{(k)} = a_{ji}^{(k)}$, $i,j \geq k$, então por indução sobre k e pelo fato de que $a_{ij}^{(1)} = a_{ji}^{(1)}$ a parte (a) do teorema fica demonstrada. Como somente os elementos da diagonal são escolhidos como pivôs então em (2.5.2), $Q_k = P_k^t$ e $\hat{a}_{ij}^{(k)} = \hat{a}_{ji}^{(k)}$, $i,j \geq k$. Agora, de (2.5.1) e (2.5.3), para $i,j > k$, temos

$$a_{ij}^{(k+1)} = \hat{a}_{ij}^{(k)} - \hat{a}_{ik}^{(k)} \hat{a}_{kj}^{(k)} / \hat{a}_{kk}^{(k)},$$

$$a_{ji}^{(k+1)} = \hat{a}_{ji}^{(k)} - \hat{a}_{jk}^{(k)} \hat{a}_{ki}^{(k)} / \hat{a}_{kk}^{(k)},$$

e portanto

$$a_{ij}^{(k+1)} = a_{ji}^{(k+1)}, \quad i,j > k.$$

Isto completa a demonstração de (a).

Agora, de (2.5.1) e (2.5.3) temos para $i>k$, $a_{ki}^{(k+1)} = \hat{a}_{ki}^{(k)} / \hat{a}_{kk}^{(k)}$ e $\hat{c}_i^{(k)} = -\hat{a}_{ik}^{(k)} / \hat{a}_{kk}^{(k)}$ e de (2.2.7) e do fato de que $\hat{a}_{ik}^{(k)} = \hat{a}_{ki}^{(k)}$, temos (2.6.2). Isto completa a demonstração do teorema.

2.7. ARMAZENAMENTO E USO DA EFI

Os $c^{(k)}$'s necessários na eliminação na forma de inversa (EFI) são

armazenados como segue: No k-ésimo passo da eliminação de Gauss todos os $\hat{a}_{ik}^{(k)} \neq 0$, $i > k$ são transformados em zeros e $\hat{a}_{kk}^{(k)}$ é transformado na unidade; isto é, $a_{ik}^{(k+1)} = 0$, $i > k$ e $a_{kk}^{(k+1)} = 1$. Portanto, de (2.5.3), está claro que cada $c_i^{(k)} \neq 0$, $i > k$, pode ser armazenado no lugar do correspondente $\hat{a}_{ik}^{(k)} \neq 0$, $i > k$; e, $c_k^{(k)}$ pode ser armazenado no lugar de $a_{kk}^{(k+1)}$ pois não há necessidade de armazenar $a_{kk}^{(k+1)} = 1$ (isto é verdadeiro para todo k; ou seja, os elementos diagonais de U são todos iguais a 1).

As matrizes de permutação P_k e Q_k em (2.5.2) podem ser facilmente construídas se l e t são conhecidos. Portanto, para cada k, são necessárias somente duas posições de memória para guardar P_k e Q_k; e, então, para guardar todos os P_k's e Q_k's de (2.5.8) são necessárias 2n posições de memória.

De (2.2.6), (2.2.7) e (2.2.8) vemos que para calcular as U_k's precisamos apenas dos elementos não nulos dos $v^{(k)}$'s. Além disso, estes elementos podem ser obtidos trocando-se os sinais de todos os elementos não nulos de U que estão acima da diagonal. Portanto, os $v^{(k)}$'s não nulos podem ser armazenados no espaço ocupado pelos elementos não nulos de U.

Os elementos não nulos de cada $A^{(k)}$ e os correspondentes $c^{(k)}$ e $v^{(k)}$ devem, é claro, ser armazenados numa das formas compactadas descritas no Capítulo 1. Como, a cada passo, k elementos não nulos adicionais são criados em $A^{(k)}$ então a lista acoplada é especialmente indicada para o armazenamento destes elementos.

A seguir, apresentamos alguns exemplos nos quais se justifica o trabalho extra exigido para a obtenção de uma EFI esparsa.

Em muitas aplicações práticas, a equação (2.1.1) deve ser

unesp

resolvida repetidamente para diferentes vetores b's e/ou para matrizes A's dos coeficientes com a mesma estrutura de esparsidade. Isto ocorre, por exemplo, na resolução de sistema de equações não lineares pelo método de Newton que conduz à (2.1.1) onde A tem uma estrutura de esparsidade fixa e os b's mudam de caso para caso (Liniger e Willoughby, 1969; Churchill, 1971). Também, na análise estrutural é necessário resolver (2.1.1) para vários vetores b's (Allwood, 1971). Nestes casos, a EFI é bastante conveniente. O uso da EFI em problemas de fluxo de energia consegue vantagens em velocidade, armazenamento e precisão que são aproximadamente proporcionais ao grau de esparsidade de A (Tinney, 1969).

Além disso, as técnicas para minimização do fill-in não geram quaisquer problemas de arredondamento. Por exemplo, Churchill (1971) observa que na resolução de problemas de fluxo de energia nenhum problema de arredondamento foi encontrado quando foram usados métodos para a minimização do fill-in.

CAPÍTULO 3

MÉTODOS ADICIONAIS PARA A MINIMIZAÇÃO DO ARMAZENAMENTO DA EFI

3.1. INTRODUÇÃO

Descrevemos agora métodos que tornam o fill-in local pequeno, durante a primeira etapa da eliminação de Gauss e que ao mesmo tempo não requerem grande esforço computacional.

Eles são geralmente denominados métodos " à priori", pois a informação para a escolha do pivô em cada estágio é obtida da matriz original e não de suas formas transformadas em cada passo. Em geral este processo conduz a uma significativa diminuição de tempo e trabalho. Porém, estes métodos são geralmente menos eficientes para a minimização do fill-in local, que os métodos descritos anteriormente.

Os métodos que descreveremos procuram encontrar uma EFI esparsa. Eles podem ser classificados em duas categorias:

i) os que envolvem, à priori, permutações de colunas.

ii) os que envolvem, à priori, permutações de linhas e colunas.

A seguir descreveremos métodos do tipo (ii) pois eles transformam A numa das formas desejáveis para a eliminação de Gauss. Nestas formas desejáveis durante a primeira etapa da eliminação de Gauss ou não existe fill-in ou este é limitado somente a certas regiões conhecidas da matriz.

Nosso objetivo será determinar matrizes de permutação P e Q, antes do início da eliminação de Gauss, de modo que

$$PAQ = \hat{A} \qquad (3.1.1)$$

onde $\hat{A}$ é uma matriz cujos elementos diagonais sendo sequencialmente

tomados como pivôs, a partir de $a_{11}^{(1)}$, conduz-nos ao fill-in total mínimo.

É desejável minimizarmos o esforço exigido para determinar P e Q.

3.2. FORMAS DESEJÁVEIS PARA A ELIMINAÇÃO DE GAUSS

A forma de banda cheia que definiremos a seguir é uma das formas mais comuns que não conduzem a qualquer fill-in, quando escolhemos para pivôs os elementos da diagonal principal.

Definição Uma matriz A é denominada de banda com largura m se, $a_{ij} = 0$, para $|i - j| > \beta$, onde $0 < \beta \leq n - 1$, $\beta = \frac{m - 1}{2}$ é o número da banda acima ou abaixo da diagonal principal.

Observe que se A é simétrica, tendo uma largura de banda localmente variável como definimos no Esquema 4 do Capítulo 1, então $m = \max \theta_i$.

Se A é de banda cheia e os pivôs são escolhidos sequencialmente na diagonal principal a partir de a_{11}, então da demonstração do Teorema (2.5.5), concluímos que o fill-in será nulo, pois como B_k é de banda cheia, sempre que $b_{ik}^{(k)} = b_{kj}^{(k)} = 1$ então $b_{ij}^{(k)} = 1$. Por outro lado, se a banda da matriz A não é cheia, então alguns elementos do interior da banda são nulos e o fill-in é limitado a esses elementos do interior da banda.

Para analisar algumas formas desejáveis assumiremos que A possa ser permutada para $\hat{A}$ por (3.1.1), onde $\hat{A}$ tem a forma

$$\hat{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1,p-1} & A_{1p} \\ 0 & A_{22} & \cdots & A_{2,p-1} & A_{2p} \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & \cdots & A_{p-1,p-1} & A_{p-1,p} \\ A_{p1} & A_{p2} & \cdots & A_{p,p-1} & A_{pp} \end{pmatrix}, \qquad (3.2.1)$$

onde as submatrizes A_{ii}, $i = 1, 2, \ldots, p$, são não singulares.

Se os pivôs são escolhidos entre os elementos não nulos dos blocos A_{ii}, a partir de A_{11}, e prosseguindo sequencialmente, então o fill-in é limitado somente aos blocos de matrizes que em (3.2.1) não estão rotulados como nulos. Posteriormente, descreveremos uma outra ordem de escolha dos pivôs de modo que não ocorra fill-in em nenhum lugar nem nas matrizes A_{ji}, $j < i$, $i \neq p$.

Em conseqüência, pretendemos determinar as matrizes de permutação P e Q tais que $\hat{A}$ em (3.1.1) tenha a forma (3.2.1). Se A é simétrica então é útil expressar $\hat{A}$ também numa forma simétrica, pois neste caso somente os elementos não nulos de A acima e sobre a diagonal precisam ser armazenados. Além disso do Teorema (2.6.1), se os elementos da diagonal principal podem ser escolhidos como pivôs (por exemplo, quando A é definida positiva), então a simetria é conservada durante o processo de eliminação de Gauss e a parte triangular inferior não precisará ser armazenada. Neste caso, no lugar de (3.1.1) temos

$$P A P^t = \hat{A}, \qquad (3.2.2)$$

e em (3.2.1), $A_{ij} = 0$, para todo $i \neq j$, exceto para $i = p$ ou $j = p$.

A seguir descreveremos alguns métodos para a determinação de P e

Q em (3.1.1) e (3.2.2) que conduzem às várias formas desejáveis de $\hat{A}$. Para isso devemos lembrar algumas noções da Teoria dos Grafos.

Matrizes e Grafos.

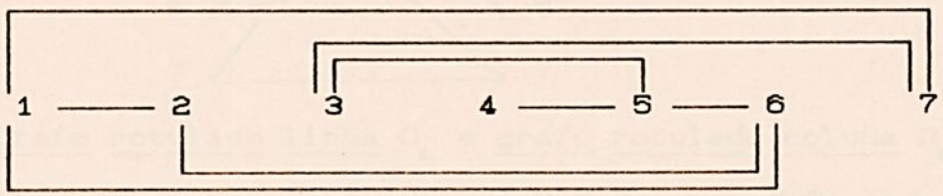
Denotaremos b_{ij} ao elemento (i, j) da matriz B que obtemos pela troca de cada elemento não nulo da matriz A pela unidade.

Um grafo rotulado Ω consiste num conjunto de n vértices rotulados como 1, 2, ..., n e τ_o arestas. É denominada aresta [p, q] ao segmento que une o vértice p ao vértice q se b_{pq} ou b_{qp} (ou ambos) são iguais a 1. Logo, τ_o é igual ao número de elementos não nulos de $B \oplus B^t$ que estão acima da diagonal.

Exemplo: Seja a matriz

$$B = \begin{pmatrix} 1 & 1 & . & . & . & 1 & . \\ 1 & 1 & . & . & . & . & . \\ . & . & 1 & . & 1 & . & 1 \\ . & . & . & 1 & . & . & . \\ . & . & . & 1 & 1 & . & . \\ . & 1 & . & . & 1 & 1 & . \\ 1 & . & . & . & . & . & 1 \end{pmatrix}$$

Temos $\tau_o = 8$. O grafo rotulado Ω associado a B é

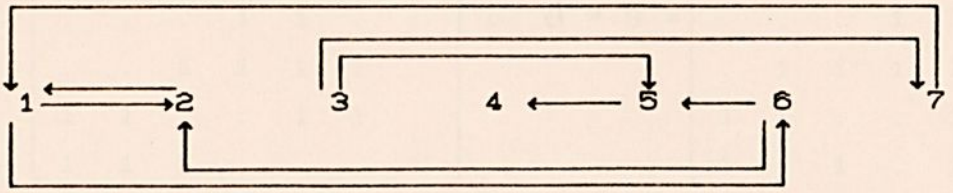


Observação: A mesma matriz B deste exemplo será usada nos exemplos a seguir.

Grafo orientado rotulado (digráfo rotulado) Ω_D, consiste de um conjunto de n vértices rotulados de 1 a n e τ_D arcos. É chamado arco [p, q] ao arco que liga o vértice p ao vértice q se, e somente se,

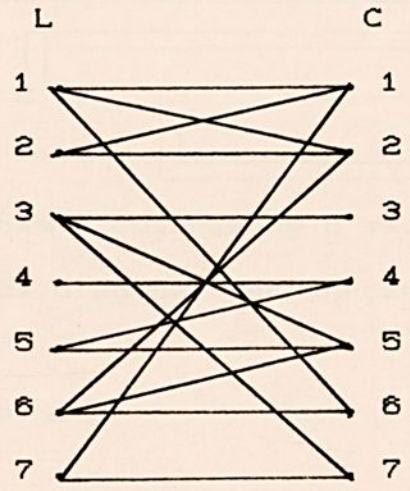
$b_{pq} = 1$. Então τ_D é igual ao número de elementos não diagonais de B.

Na matriz do mesmo exemplo anterior temos $\tau_D = 9$ e o grafo orientado rotulado é



Um grafo bipartido rotulado (bigrafo rotulado) Ω_B, consiste de dois conjuntos distintos de vértices L e C, cada um com n elementos rotulados 1, 2, ..., n e τ arestas que unem vértices de L aos de C. Existe uma aresta [p, q] com vértice p no conjunto L e vértice q no conjunto C se, e somente se, $b_{pq} = 1$. Logo τ é o número total de elementos não nulos da matriz B.

Em nosso exemplo temos $\tau_B = 16$ e o grafo bipartido rotulado é

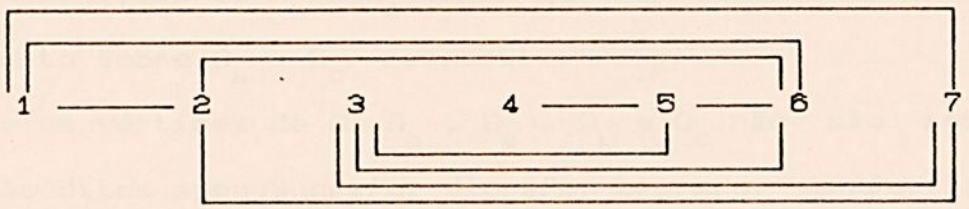


Grafo rotulado linha Ω_L e grafo rotulado coluna Ω_C são, respectivamente, os grafos rotulados de $B * B^t$ e $B^t * B$, onde * indica que no produto interno de vetores, necessários nos produtos das matrizes, usamos soma Booleana $\oplus$. Como $B * B^t$ e $B^t * B$ são simétricas, então τ_L e τ_C são iguais ao número de 1's que estão acima da diagonal de $B * B^t$ e de $B^t * B$, respectivamente.

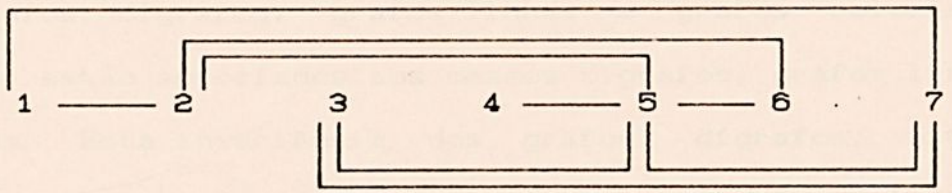
No exemplo temos

$$B * B^t = \begin{pmatrix} 1 & 1 & . & . & . & 1 & 1 \\ 1 & 1 & . & . & . & 1 & 1 \\ . & . & 1 & . & 1 & 1 & 1 \\ . & . & . & 1 & 1 & . & . \\ . & . & 1 & 1 & 1 & 1 & . \\ 1 & 1 & 1 & . & 1 & 1 & . \\ 1 & 1 & 1 & . & . & . & 1 \end{pmatrix} \quad e \quad B^t * B = \begin{pmatrix} 1 & 1 & . & . & . & . & 1 \\ 1 & 1 & . & . & 1 & 1 & . \\ . & . & 1 & . & 1 & . & 1 \\ . & . & . & 1 & 1 & . & . \\ . & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & . & 1 \end{pmatrix}$$

Então, o grafo rotulado linha Ω_L será



e o grafo rotulado coluna Ω_C será:



Se as linhas e colunas de B são permutadas de modo que todos os elementos de sua diagonal permaneçam na diagonal, ou seja,

$$P\ B\ P^t = \tilde{B} \qquad (3.2.3)$$

onde P é matriz de permutação, então serão trocados apenas os rótulos dos vértices de Ω e Ω_D, mas os grafos rotulados não se alteram.

A aplicação de diferentes permutacões de linhas e colunas a B, isto é,

$$P\ B\ Q = \hat{B} \qquad (3.2.4)$$

onde P e Q são matrizes de permutação, deixam Ω_B inalterado exceto por uma renumeração dos vértices em L e C.

Permutações P de linhas em B renumeram os vértices em Ω_L e permutações Q de colunas renumeram os vértices de Ω_C.

De (3.2.4) temos

$$P\,B * B^t\,P^t = \hat{B} * \hat{B}^t \qquad e \qquad Q^t\,B^t * B\,Q = \hat{B}^t * \hat{B},$$

e das definições de Ω_L e Ω_C temos que Q e P em (3.2.4) não produzem efeito sobre Ω_L e Ω_C, respectivemente.

Se os vértices de Ω, Ω_D, Ω_B, Ω_L e Ω_C não são rotulados então eles são ditos apenas grafo, dígrafo, bigrafo, grafo linha e grafo coluna. Assim, as matrizes B e $\tilde{B}$ em (3.2.3) estão associados aos mesmos grafos, dígrafos, grafos linhas e grafos colunas. B e $\hat{B}$ em (3.2.4) estão associados aos mesmos bigrafos, grafos linhas e grafos colunas. Esta invariância dos grafos, dígrafos, bigrafos, grafos linhas e grafos colunas com relação à permutação de linhas e colunas, torna-se especialmente útil no estudo do fill-in e na obtenção das formas desejáveis para a eliminação de Gauss.

Outras definições.

Os vértices p e q que são unidos por uma aresta em Ω, Ω_C ou Ω_L, ou por um arco em Ω_D, são chamados vértices adjacentes. Se existe um subconjunto $v_1, \ldots, v_\sigma, v_{\sigma+1}$ de vértices distintos tais que para $i = 1, 2, \ldots, \sigma$, os vértices v_i, v_{i+1} são adjacentes, então v_i e $v_{\sigma+1}$ são ditos conectados por um caminho $[v_1, v_2, \ldots, v_{\sigma+1}]$ em Ω, Ω_C ou Ω_L (por um caminho orientado em Ω_D) de comprimento σ.

Se em $[v_1, \ldots, v_{\sigma+1}]$ o vértice inicial v_1 e o vértice final $v_{\sigma+1}$

são os mesmos, então o caminho é denominado um ciclo orientado de Ω_D, ou um ciclo de Ω, Ω_L ou Ω_C de comprimento σ. Como em Ω, Ω_L, Ω_C e Ω_D o número total de vértices é n, então o valor máximo de σ será n.

Se o conjunto de vértices em Ω (Ω_D, Ω_C e Ω_L) pode ser dividido em dois ou mais subconjuntos de modo que somente os vértices de um subconjunto sejam conectados, então dizemos que Ω (Ω_L, Ω_C, Ω_D) possuem dois ou mais subgrafos rotulados disjuntos.

O número de arestas de Ω (Ω_L, Ω_C, Ω_B) que ocorrem num dado vértice é o grau do vértice. Assim, o grau de um vértice i em L de Ω_B é o número de elementos iguais a 1 da i-ésima linha de B; e, o grau de um vértice i de Ω é o número de elementos iguais a 1 fora da diagonal principal na i-ésima linha de $B \oplus B^t$. O número de arcos de Ω_D que partem para um dado vértice é o grau externo do vértice. Então, o grau externo de um vértice i é igual ao número de 1's na linha i de B. O grau interno do vértice é igual ao número de arcos de Ω_D que terminam neste vértice. Então, o grau interno de um vértice i é igual ao número total de 1's fora da diagonal na i-ésima coluna de B. Um vértice do qual somente partam arcos é chamado vértice emissor. Um vértice não conectado a qualquer outro é dito vértice isolado.

Teorema (3.2.5): Sejam

$$W = B \oplus B^t \oplus I \qquad (3.2.6)$$

$$W^{\sigma+1} = W^{\sigma} * W \qquad (3.2.7)$$

onde $\sigma = 1, 2, \ldots, n-1$. Então $e_i^t W^{\sigma} e_j = 1$ se, e somente se, os vértices i e j do grafo rotulado associado à B são conectados por um caminho de comprimento menor ou igual à σ.

Demonstração: Por indução. O teorema é certamente verdadeiro para $\sigma = 1$, pois $e_i^t W e_j = w_{ij} = 1$ se, e somente se, existe uma aresta de comprimento 1 entre i e j.

Supondo o teorema válido para σ mostraremos que é verdadeiro para $\sigma + 1$.

De (3.2.7) temos

$$w_{ij}^{\sigma+1} = \sum_{p=1}^{n} w_{ij}^{\sigma} * w_{pj} \qquad (3.2.8)$$

Agora

$$w_{ij}^{\sigma+1} = 1 \qquad (3.2.9)$$

se, e somente se, $w_{ip}^{\sigma} = 1$ e $w_{pj} = 1$, para pelo menos um p. Mas $w_{ij}^{\sigma} = 1$ se, e somente se, os vértices i e p são conectados por um caminho de comprimento menor ou igual a σ ou menor e $w_{pj} = 1$ se, e somente se, os vértices p e j são conectados por uma aresta quando $p \neq j$. Se $p = j$ temos $w_{jj} = 1$ em conseqüência de (3.2.6). Portanto, em qualquer caso, (3.2.9) se verifica se, e somente se, os vértices i e j são conectados por um caminho de comprimento menor ou igual a $\sigma + 1$. Isto completa a prova do teorema.

Desde que o comprimento máximo de um caminho é n (pois existem n vértices), a matriz W^n será usada mais à frente para determinar todos os caminhos e ciclos.

Um teorema similar ao Teorema (3.2.5) para grafos orientados rotulados é:

Teorema (3.2.10): Sejam

$$\hat{W} = B \oplus I,$$

$$\hat{W}^{\sigma+1} = \hat{W}^{\sigma} * \hat{W}$$

com $\sigma = 1, 2, \ldots, n$. Então $e_i \hat{W}^{\sigma} e_j = 1$ se, e somente se, no dígrafo rotulado associado a B existe um caminho orientado do vértice i ao j de comprimento menor ou igual a σ.

Supondo o teorema válido para σ mostraremos que é verdadeiro para $\sigma + 1$.

De (3.2.7) temos

$$w_{ij}^{\sigma+1} = \sum_{p=1}^{n} w_{ip}^{\sigma} * w_{pj} \qquad (3.2.8)$$

Agora

$$w_{ij}^{\sigma+1} = 1 \qquad (3.2.9)$$

se, e somente se, $w_{ip}^{\sigma} = 1$ e $w_{pj} = 1$, para pelo menos um p. Mas $w_{ij}^{\sigma} = 1$ se, e somente se, os vértices i e p são conectados por um caminho de comprimento menor ou igual a σ ou menor e $w_{pj} = 1$ se, e somente se, os vértices p e j são conectados por uma aresta quando $p \neq j$. Se $p = j$ temos $w_{jj} = 1$ em conseqüência de (3.2.6). Portanto, em qualquer caso, (3.2.9) se verifica se, e somente se, os vértices i e j são conectados por um caminho de comprimento menor ou igual a $\sigma + 1$. Isto completa a prova do teorema.

Desde que o comprimento máximo de um caminho é n (pois existem n vértices), a matriz W^n será usada mais à frente para determinar todos os caminhos e ciclos.

Um teorema similar ao Teorema (3.2.5) para grafos orientados rotulados, e demonstrado por Tewarson, 1973 é:

Teorema (3.2.10): Sejam

$$\hat{W} = B \oplus I,$$

$$\hat{W}^{\sigma+1} = \hat{W}^{\sigma} * \hat{W}$$

com $\sigma = 1, 2, \ldots, n$. Então $e_i^t \hat{W}^{\sigma} e_j = 1$ se, e somente se, no dígrafo rotulado associado a B existe um caminho orientado do vértice i ao j de comprimento menor ou igual a σ.

3.3. FORMA DIAGONAL DE BLOCOS (FDB)

A matriz $\hat{A}$ definida por (3.2.1) está na Forma Diagonal de Blocos (FDB) se para todo $i \neq j$, $A_{ij} = 0$ e os blocos diagonais A_{ii} são matrizes quadradas. Se $\hat{A}$ está na FDB então, de (3.1.1) e (3.2.4), $\hat{B}$ também está nesta forma. Isto implica que os grafos Ω_L e Ω_C, associados a $\hat{B}$ consistem de subgrafos rotulados disjuntos, e cada subgrafo rotulado está associado a um bloco diagonal. Desde que os grafos Ω_L e Ω_C, associados a $\hat{B}$ e a B são os mesmos exceto por uma reordenação dos vértices, Ω_L e Ω_C associados a B podem ser usados para a determinação de P e Q em (3.1.1) como mostra o Teorema (Harary, 1962; Tewarson, 1967) abaixo, cuja demonstração pode ser encontrada em Tewarson, 1973.

Teorema (3.3.1): Se

$$W = B * B^t \qquad (3.3.2)$$

e

$$W^{2^{h+1}} = W^{2^h} * W^{2^h}, \qquad h = 0, 1, \ldots \qquad (3.3.3)$$

então existe h_1 tal que

$$W^{2^{h_1+1}} = W^{2^{h_1}} = F \qquad (3.3.4)$$

e $e_i^t F e_j = 1$ se, e somente se, as linhas i e j de B pertencem ao mesmo bloco diagonal.

Quando se utiliza este teorema, a equação (3.3.3) é usada até $2^{h_1} \geq n$, pois geralmente não conhecemos o comprimento v da diagonal do maior bloco de $\hat{B}$, mas somente o fato de que $v \leq n$. Tomamos $F = W^{2^{h_1}}$. Em lugar de (3.3.3), existem vários métodos práticos para se determinar a matriz F (Baker, 1962; Warshall, 1962; Ingermam, 1962; Comstock, 1964). A seguir mostraremos o método de Comstock.

Primeiramente, fazemos uma pesquisa na primeira linha de W até encontrarmos um zero, digamos, na coluna j. Então este zero é trocado

por

$$\sum_{p=1}^{n} w_{ip}\, w_{pj} \qquad \text{(Booleana)}$$

A pesquisa continua com as colunas j + 1, j + 2, ... da linha 1 até encontrarmos um outro zero, digamos, na coluna q. Este zero é trocado por

$$\sum_{p=1}^{n} w_{ip}\, w_{pq} \,.$$

Este processo continua até que toda a linha tenha sido pesquisada. Então, as linhas restantes são também pesquisadas e modificadas do mesmo modo. Quando todas as linhas tiverem sido pesquisadas reinicia-se a pesquisa novamente na primeira linha. O processo continua até que se tenha feito uma varredura completa por todas as linhas da matriz sem nenhuma troca. A matriz resultante é a matriz F.

É evidente que $f_{ij} = e_i^t F e_j = 1$ se, e somente se, as linhas i e j pertencem ao mesmo bloco diagonal. Assim todas as linhas de B que correspondem a elementos não nulos da primeira coluna de F pertencem ao primeiro bloco diagonal. Se todas as linhas de F, para as quais $f_{i1} = 1$ são eliminadas, então a próxima coluna não nula de F pode ser usada da mesma maneira que a primeira coluna de F, para determinar as linhas de B que pertencem ao segundo bloco diagonal de $\hat{B}$, e assim em diante.

Do mesmo modo podemos encontrar os blocos diagonais de $\hat{B}$ aos quais pertencem cada linha de B.

O corolário a seguir, pode ser usado para determinar as colunas de B dos diferentes blocos diagonais.

Corolário (3.3.5): Se F é a matriz definida no Teorema (3.3.1) e

$$F * B = \overline{F} \qquad (3.3.6)$$

e $\bar{f}_{ij} = e_i^t \bar{F} e_j = 1$, então a linha i e a coluna j de B pertencem ao mesmo bloco diagonal de $\hat{B}$.

Demonstração: De (3.3.6) temos

$\bar{f}_{ij} = \sum_{p=1}^{n} f_{ip} * b_{pj}$, e $\bar{f}_{ij} = 1$ se, e somente se,

$f_{ip} = b_{pj} = 1$, para pelo menos um valor de p.

Do Teorema (3.3.1), sabemos que $f_{ip} = 1$ implica que as linhas i e p pertencem ao mesmo bloco diagonal de $\hat{B}$, e como a coluna j tem pelo menos um elemento não nulo na linha p, então a coluna j estaria também no mesmo bloco diagonal que as linhas p e i. Então, $\bar{f}_{ij} = 1$ implica que a linha i e a coluna j pertencem ao mesmo bloco diagonal.

Para usarmos este corolário procedemos como segue:

Para todas as colunas de B que pertencem ao primeiro bloco diagonal selecionado de acordo com o Teorema (3.3.1) devemos ter $\bar{f}_{ij} = 1$. Deletando ou marcando estas colunas em $\bar{F}$, a próxima linha não nula da matriz resultante fornecerá o conjunto de colunas de B que pertencem ao segundo bloco diagonal; e, assim em diante.

Se no Teorema (3.3.1), em vez de (3.3.2), definirmos $W = B^t * B$, teremos um teorema para a permutação das colunas de B para $\hat{B}$. Contudo, apresentamos a seguir uma versão ligeiramente modificada que usa o Teorema (3.3.1) e assim evita a duplicação do trabalho.

Teorema (3.3.7): Se F é definida como no Teorema (3.3.1), e

$$e_i^t (B^t * F * B) e_j = 1, \qquad (3.3.8)$$

então as colunas i e j de B pertencem ao mesmo bloco diagonal de $\hat{B}$.

Demonstração: De (3.3.2), (3.3.3) e (3.3.4) temos

$$B^t * F * B = B^t * [(B * B^t) * (B * B^t) * \ldots * (B * B^t)] * B$$

$$= (B^t * B) * (B^t * B) * \ldots * (B^t * B)$$

$$= (B^t * B)^{\sigma}, \qquad \sigma > v.$$

Portanto, da definição do grafo coluna Ω_c e com os mesmos argumentos usados na prova do Teorema (3.3.1) segue que $e_i^t (B^t * F * B) e_j = e_i^t (B^t * B)^{\sigma} e_j = 1$, implica que as colunas i e j são conectadas por um caminho e portanto pertencem ao mesmo bloco diagonal. Isto completa a demonstração.

Como (3.3.8) requer uma multiplicação de matriz a mais que (3.3.6), o Corolário (3.3.5) é geralmente usado no lugar do Teorema (3.3.7) para permutar as colunas na FDB.

Exemplo. Seja

$$B = \begin{pmatrix} . & 1 & 1 & . \\ 1 & . & . & 1 \\ 1 & . & . & . \\ . & 1 & 1 & . \end{pmatrix}. \qquad \text{Então } W = B * B^t = \begin{pmatrix} 1 & . & . & 1 \\ . & 1 & 1 & . \\ . & 1 & 1 & . \\ 1 & . & . & 1 \end{pmatrix}$$

e $$W^2 = \begin{pmatrix} 1 & . & . & 1 \\ . & 1 & 1 & . \\ . & 1 & 1 & . \\ 1 & . & . & 1 \end{pmatrix} = W.$$ Logo, $F = W^2 = W$.

Do Teorema (3.3.1) temos $e_1^t F e_4 = 1$, $e_1^t F e_2 = 0$, $e_1^t F e_3 = 0$ e $e_2^t F e_3 = 1$.

Assim, a primeira e a quarta linhas de B pertencem ao primeiro bloco e a segunda e a terceira linhas de B pertencem ao segundo bloco. Do Corolário (3.3.5), concluímos que a segunda e a terceira colunas de B pertencem ao primeiro bloco diagonal e as colunas restantes pertencem ao segundo bloco diagonal. Portanto,

$$P = \begin{pmatrix} 1 & . & . & . \\ . & . & . & 1 \\ . & 1 & . & . \\ . & . & 1 & . \end{pmatrix}, \qquad Q = \begin{pmatrix} . & . & 1 & . \\ 1 & . & . & . \\ . & 1 & . & . \\ . & . & . & 1 \end{pmatrix}$$

e então

$$\hat{B} = PBQ = \begin{pmatrix} \boxed{\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}} & . & . \\ . & . & \boxed{\begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}} \\ . & . & \end{pmatrix}$$

3.4. FORMA TRIANGULAR DE BLOCOS (FTB)

Se em $\hat{A}$ dada por (3.2.1), $A_{pj} = 0$, $j \neq p$ e A_{ii}, $i = 1, 2, \ldots, p$ são matrizes quadradas então $\hat{A}$ é dita estar numa Forma Triangular de Blocos (FTB).

Vamos agora, descrever alguns métodos para transformar uma dada matriz B (que é o mesmo que permutar A) numa FTB.

Primeiro método.

O primeiro método é realizado em duas etapas:

$$\hat{P}\, B\, \hat{Q} = \tilde{B} \qquad (3.4.1)$$

e

$$\tilde{P}\, \tilde{B}\, \tilde{P}^t = \hat{B} \qquad (3.4.2)$$

onde $\tilde{B}$ possui todos os elementos da diagonal principal iguais a 1 e $\hat{B}$ está na FTB.

De (3.4.1) e (3.4.2) temos

$$P\, B\, Q = \hat{B}\ , \quad P = \tilde{P}\, \hat{P} \quad e \quad Q = \hat{Q}\, \tilde{P}^t \qquad (3.4.3)$$

Assim, precisamos determinar P e Q em (3.4.3).

Se A é uma matriz não singular existe pelo menos um termo não

nulo em seu determinante. Este termo consiste de n elementos de A dos quais dois quaisquer nunca aparecem na mesma linha ou coluna de A. Desta maneira as linhas e colunas de A podem ser permutadas independentemente para trazer os elementos não nulos deste termo para a diagonal. A determinação de $\hat{P}$ e $\hat{Q}$ em (3.4.1) tal que $\tilde{b}_{ii} = 1$, para todo $i = 1, 2, \ldots, n$ foi realizada por Steward, 1962; 1965; Dulmage e Mendelsohn, 1963; Kettler e Weil, 1969; Harary, 1971; Duff, 1972; e é feita como segue:

Algoritmo (3.4.4): Toma-se

$$B_1 = B \quad , \quad V_1 = \sum_{i=1}^{n} e_i \quad , \quad U_1 = V_1^t \quad , \quad P_1 = Q_1 = I_n \quad e \quad k = 1$$

Passo 1:

Para todo i, j com

$$b_{ij} = 1 \quad , \quad e_i^t V_k = 1 \quad , \quad U_k e_j = 1 \quad , \qquad (3.4.5)$$

calcula-se

$$e_{\alpha_k}^t B V_k + U_k B e_{\beta_k} = \min_{i,j} \left(e_i^t B V_k + U_k B e_j \right) . \qquad (3.4.6)$$

Se não existir o par (i,j) para o qual (3.4.5) seja válido vai-se para o Passo 2; caso contrário, faz-se os elementos α_k de V_k e β_k de U_k iguais a zero. Troca-se a k-ésima e a α_k-ésima linhas de P_k e a k-ésima e a β_k-ésima colunas de Q_k.

Troca-se k por k + 1. Se k = n + 1 vai-se para o Passo 2; caso contrário, retorna-se ao início deste passo.

Observação: Para cada valor de k, selecionamos o elemento (α_k, β_k) de B para ser o elemento (k,k) de $\tilde{B}$ em $\hat{P} B \hat{Q} = \tilde{B}$. Como todos os elementos que pertencem à linha α_k ou à coluna β_k de B não podem ser escolhidos posteriormente para elementos diagonais de $\tilde{B}$, então (3.4.5) e (3.4.6) garantem que um número máximo de elementos não nulos de B serão úteis como possíveis escolhas para o resto dos elementos diagonais de $\tilde{B}$.

unesp

Assim, no final deste passo teremos geralmente algumas poucas linhas e colunas de B que não tenham sido usadas na escolha dos elementos diagonais de $\tilde{B}$. Se k = n + 1, então todos os elementos diagonais não nulos terão sido encontrados.

Passo 2:

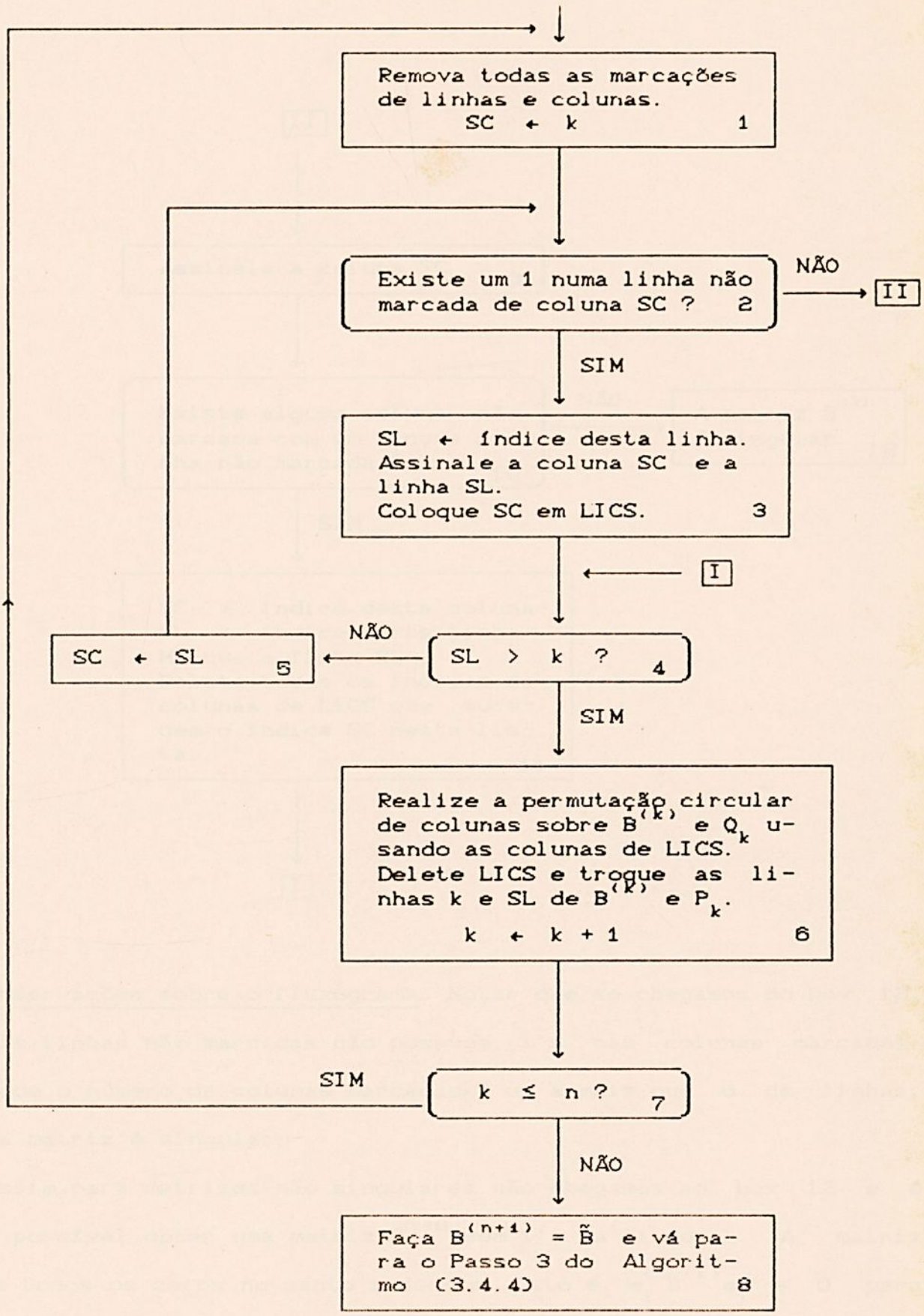
Calcula-se $B^{(k)} = P_k \, B \, Q_k$.

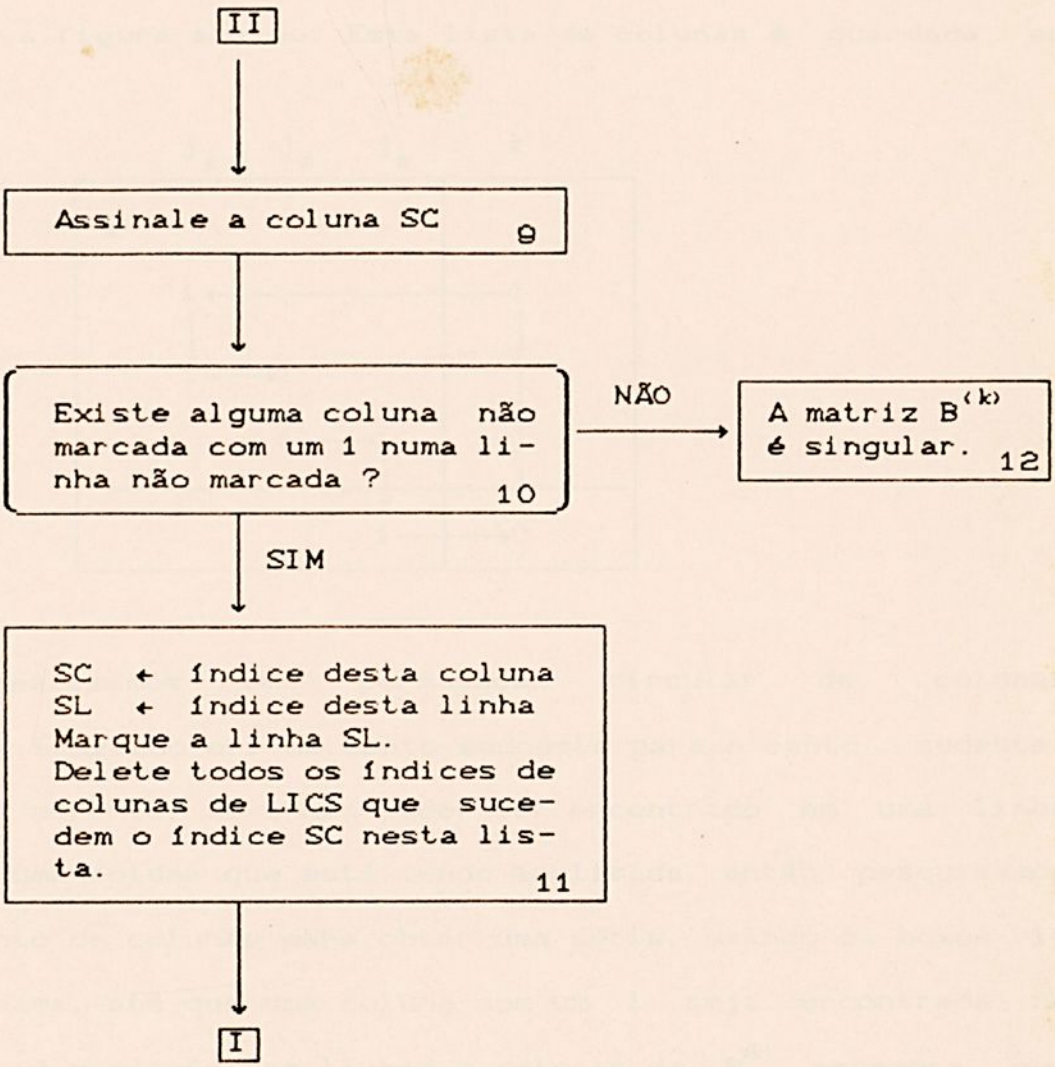
Se k = n + 1, então faz-se $\tilde{B} = B^{(k)}$ e vai-se para o Passo 3; caso contrário, usa-se os passos do fluxograma seguinte sobre a matriz $B^{(k)}$ para permutá-la na matriz $B^{(n+1)}$ que possui todos os elementos diagonais iguais a 1. No fluxograma, usado para criar uma diagonal de 1's em $B^{(k)}$, temos

SC = índice da coluna selecionada

SL = índice da linha selecionada

LICS = lista de índices de colunas numa série

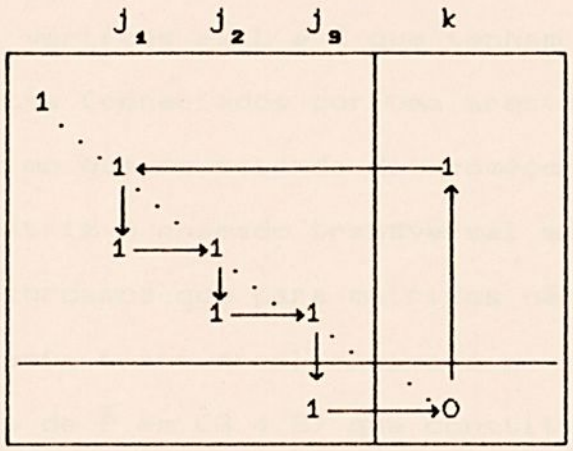

Remova todas as marcações de linhas e colunas.
SC ← k 1
Existe um 1 numa linha não marcada de coluna SC ? 2
NÃO
II
SIM
SL ← índice desta linha.
Assinale a coluna SC e a linha SL.
Coloque SC em LICS. 3
I
SL > k ? 4
NÃO
SC ← SL 5
SIM
Realize a permutação circular de colunas sobre $B^{(k)}$ e Q_k usando as colunas de LICS.
Delete LICS e troque as linhas k e SL de $B^{(k)}$ e P_k.
k ← k + 1 6
k ≤ n ? 7
SIM
NÃO
Faça $B^{(n+1)} = \tilde{B}$ e vá para o Passo 3 do Algoritmo (3.4.4) 8



Observações sobre o fluxograma: Notar que se chegamos do box 12, então as linhas não marcadas não possuem 1's nas colunas marcadas. Desde que o número de colunas marcadas é um a mais que o de linhas, então a matriz é singular.

Assim para matrizes não singulares não chegamos ao box 12 e é sempre possível obter uma matriz $B^{(n+1)}$ com 1's na diagonal. A matriz $B^{(k)}$ tem todos os zeros no canto sudoeste, isto é, $e_i^t B^{(k)} e_j = 0$ para todo $i,j \geq k$. Para a transformação de $B^{(k)}$ em $B^{(k+1)}$ é necessário

encontrar-se primeiramente uma série a partir de um 1 na k-ésima coluna no canto nordeste até um 1 na mesma linha no canto sudoeste de $B^{(k)}$ como mostra a figura abaixo. Esta lista de colunas é guardada em LICS.



Então realizamos uma permutação circular de colunas $k \rightarrow j_1 \rightarrow j_2 \rightarrow j_3 \rightarrow k$ trazendo o 1 do canto sudoeste para o canto sudeste. Se, em qualquer momento, um 1 não puder ser encontrado em uma linha não marcada de uma coluna que está sendo analisada, então pesquisamos em outro conjunto de colunas para obter uma série, usando os boxes 10 e 11 do fluxograma, até que uma coluna com um 1 seja encontrada no canto sudoeste. A marcação das linhas e colunas de $B^{(k)}$ assegura que um caminho que conduza a uma coluna sem elementos em suas linhas não assinaladas não é repetido novamente e a série é determinada num número finito de passos. Observar que LICS contém uma lista de índices de colunas de $B^{(k)}$ que formam uma série desde um 1 da k-ésima coluna até um 1 no canto sudoeste e que portanto será modificado no box 11 para livrar-se daquelas colunas que conduzem a um beco sem saída na busca de uma série.

Passo 3:

Toma-se $P_k = \hat{P}$ e $Q_k = \hat{Q}$. $\tilde{B}$ tem todos os elementos diagonais

iguais a 1; o que conclui o algoritmo.

Observação. A determinação de $\hat{P}$ e $\hat{Q}$ tais que $\tilde{B}$ tenha todos os elementos diagonais não nulos é equivalente à uma renumeração dos vértices do grafo bipartido rotulado Ω_B correspondente a B, de modo que quaisquer dois vértices em L e C que tenham o mesmo número como rótulo são adjacentes (conectados por uma aresta). Um string de bits 1 de comprimento máximo que se estenda do começo ao fim da diagonal principal de uma matriz é chamado transversal máxima (Dulmage e Mendelsohm, 1963). Recordamos que para matrizes não singulares de ordem n, o comprimento desta transversal máxima é n.

A determinação de $\tilde{P}$ em (3.4.2) que constitui a segunda parte do primeiro método de transformação de B numa forma triangular de blocos $\hat{B}$ é baseada no teorema seguinte.

Teorema (3.4.7): Se todos os elementos da diagonal principal de B são 1's e

$$B^{2^{h+1}} = B^{2^h} * B^{2^h} \qquad (3.4.8)$$

então existe um h_1 tal que

$$B^{2^{h_1+1}} = B^{2^{h_1}} = F \qquad (3.4.9)$$

e $e_i^t F e_j = 1$ se, e somente se, existe um caminho orientado do vértice i para o vértice j em Ω_D.

Demonstração:

Desde que $B \oplus I = B$, do Teorema (3.2.10) temos que $e_i^t B^v e_j = 1$ se, e somente se, existe um caminho orientado de comprimento menor ou igual a v entre os vértices i e j de Ω_D. Se v é o comprimento máximo do caminho orientado entre dois vértices quaisquer de Ω_D, então $v \leq n$ e, evidentemente, para todo $2^h \geq v$ a matriz B^{2^h} permanece inalterada. Portanto, existe em h_1 satisfazendo (3.4.9) e então

$e_i^t F e_j = 1$ se, e somente se, existe um caminho orientado do vértice i para o vértice j.

Se no Teorema (3.4.7) em vez de B iniciamos com $\tilde{B}$ que foi obtido de B no algoritmo (3.4.4) e então calculamos F de acordo com (3.4.8) e (3.4.9) ou pelo método de Comstock então $e_i^t F e_j = e_j^t F e_i = 1$ implica que os vértices i e j pertencem ao mesmo ciclo orientado. Como os vértices de Ω_D que estão no mesmo ciclo orientado pertencem ao mesmo bloco diagonal em $\hat{B}$ podemos usar F para determinar $\hat{B}$ a partir de $\tilde{B}$ como segue (Harary, 1962):

Determine todos os j tais que $e_1^t F e_j = e_j^t F e_1 = 1$. Então todas as linhas e colunas de $\tilde{B}$ com estes índices j pertencem ao mesmo bloco diagonal de $\hat{B}$. Marque (ou delete) todas essas linhas e colunas de F. A próxima linha não marcada (ou não deletada) e a correspondente coluna, pode agora ser usada exatamente do mesmo modo que a primeira linha e primeira coluna foram usadas, para determinar as linhas e colunas de um outro bloco diagonal; e, assim em diante. Deste modo todas as linhas e coluna de $\tilde{B}$ podem ser associadas a blocos diagonais apropriados. Notar que se $F = F^t$, então $\tilde{B}$ pode ser permutada para uma FDB (esta poderia ser uma alternativa à forma de se determinar F, mas é mais lenta que a já estudada).

No momento de organizar os blocos diagonais tal que $\hat{B}$ está na FTB procedemos como segue:

A cada bloco diagonal associamos uma linha e uma coluna que são formadas, respectivamente, pela soma Booleana de todas as linhas e colunas de F que estão associadas ao bloco diagonal. Deste modo obtemos a partir de F uma nova matriz quadrada $\hat{F}$ que tem a mesma ordem que o número de blocos diagonais. O dígrafo rotulado correspondente a F é

unesp

chamado digrafo rotulado de condensação Ω_s de $\tilde{B}$. Seus vértices são os blocos diagonais. Como uma reordenação nos vértices não altera o grafo de condensação, $\hat{B}$ tem o mesmo grafo de condensação. Assim, Ω_s deve ter um vértice emissor ($\hat{B}$ não tem blocos não nulos abaixo da diagonal principal) ou um vértice isolado. O bloco diagonal correspondente a este vértice Ω_s é o primeiro bloco diagonal. Abandonando a linha e a coluna de $\hat{F}$ que possibilitaram a escolha do primeiro bloco diagonal e cancelando-se os correspondentes vértices e arcos de Ω_s, descobrimos que o Ω_s resultante (e F) deve também ter um vértice emissor ou um vértice isolado e o correspondente bloco diagonal é tomado como o segundo bloco diagonal; e, assim em diante. Assim, a ordem dos blocos diagonais de $\tilde{B}$ é determinada. Um registro de todas estas permutações é feito e o resultado final é denotado por $\tilde{P}$. Então, usando (3.4.2), a matriz de blocos triangular superior $\hat{B}$ é determinada.

Segundo método.

O segundo método para transformar B em FTB é idêntico ao primeiro método até a determinação da transversal máxima, mas a segunda parte é diferente. A diferença é baseada nas seguintes observações:

É possível evitar o cálculo da matriz F dada por (3.4.9) quando transformamos $\tilde{B}$ em $\hat{B}$, como mostrou (Steward, 1965).

Primeiramente notamos que os blocos diagonais de $\hat{B}$, que têm ordem maior ou igual a dois, correspondem aos ciclos dos grafos orientados associados a eles. Por outro lado, todas as colunas de $\hat{B}$ que não possuem elementos não nulos abaixo da diagonal e que estão antes do primeiro bloco diagonal podem ser facilmente obtidas de $\tilde{B}$ da seguinte maneira:

Passo 1:

Determinamos uma coluna de $\tilde{B}$ que possua somente um único 1 e

movemos esta coluna e sua correspondente linha (que pode ter muitos 1's) para o canto noroeste e marcamos esta coluna e a correspondente linha. Este processo é repetido até que não haja mais colunas com apenas um único elemento 1 nas linhas não marcadas entre as colunas não marcadas de $\tilde{B}$.

Passo 2:

Determinamos as colunas (e linhas) que pertencem ao mesmo bloco diagonal que a primeira coluna não marcada de $\tilde{B}$: Do Passo 1 segue que a primeira coluna não marcada de $\tilde{B}$ tem pelo menos um 1 fora da diagonal numa linha não marcada. A coluna correspondente à linha do primeiro tal 1 deve também ter um 1 numa linha não marcada; e, assim por diante. Em algum momento, uma coluna previamente mencionada é encontrada. Isto então completa um ciclo orientado. Trocamos o conjunto de colunas num ciclo orientado por uma coluna que é a soma Booleana de todas estas colunas sem seus elementos diagonais.

Isto é chamado colapso de colunas num grafo orientado. A soma Booleana de vetores é a adição usual de vetores onde $1 \oplus 1 = 1$. Similarmente, as linhas do grafo orientado são trocadas por uma linha que é a soma Booleana delas. Quando uma coluna com somente um único 1 é marcada, então as colunas nas quais houve colapso para formá-la, pertencem ao mesmo bloco.

A ordem na qual as colunas com um único 1 foram marcadas fornecenos a ordem dos blocos diagonais. Continuamos com os Passos 1 e 2 até que todas as colunas e linhas da matriz tenham sido marcadas.

Exemplificaremos estas regras começando com a figura (3.4.10) a seguir (que representa uma matriz) e prosseguiremos passo a passo até terminarmos na figura (3.4.11).

$$\begin{bmatrix} x & x & . & . & . & x & . \\ x & x & . & . & . & . & . \\ . & . & x & . & x & . & x \\ . & . & . & x & . & . & . \\ . & . & . & x & x & . & . \\ . & x & . & . & x & x & . \\ x & . & . & . & . & . & x \end{bmatrix} \qquad (3.4.10)$$

As posições com 1's são indicadas por x; as, com zeros por .; e qualquer outra criada pelo colapso de linhas ou colunas será indicada por +.

1. A coluna 3 tem um único 1. Marcamos a coluna 3 e a linha 3. Colocamos um 1 na linha 3 da coluna "ordem" para registrar a seqüência na qual a coluna foi eliminada.

2. A coluna 7 tem um único 1 nas suas linhas não marcadas. Marcamos a coluna 7 e a linha 7 e colocamos um 2 na coluna "ordem".

ordem	em		1	2	3	4	5	6	7
3		1	x	x	.	.	.	x	.
	1	2	x	x	.	.	.	.	.
1		3	.	.	x	.	x	.	x
5		4	.	.	.	x	.	.	.
4		5	.	.	.	x	x	.	.
	1	6	+	x	.	.	x	x	.
2		7	x	.	.	.	.	.	x

(3.4.11)

3. Não existem mais colunas não marcadas com um único 1 nas li-

nhas não marcadas. Começamos traçando um caminho orientado com coluna 1, escolhendo sempre o primeiro 1 encontrado na coluna. Isto nos dá o caminho orientado [1,2,1] que implica que 1 e 2 estão num ciclo orientado. Tombamos (fazemos o "colapso") a coluna 2 sobre a coluna 1 acrescentando +'s à coluna 1 ao fazer a soma Booleana da coluna 1 com os 1's não diagonais da coluna 2. Analogamente, fazemos o "colapso" da linha 2 sobre a linha 1, e marcamos a coluna 2 e a linha 2. Colocamos 1 na segunda linha da coluna "em" para indicar que a coluna 2 foi tombada sobre a coluna 1.

4. Começando com a coluna 1 traçamos o caminho orientado [1,6,1] e tombamos a coluna 6 sobre a coluna 1 e a linha 6 sobre a linha 1. Marcamos a coluna 6 e a linha 6 e colocamos um 1 na linha 6 da coluna "em".

5. A coluna 1 tem agora somente um único 1 em linhas não marcadas. Portanto, colocamos um 3 na primeira linha da coluna "ordem" e marcamos a linha 1 e a coluna 1.

6. A coluna 5 tem um único 1 nas suas linha não marcadas e então colocamos um **4** na coluna "ordem" e marcamos a linha 5 e a coluna 5.

7. A coluna 4 tem um único 1 nas linhas não marcadas; portanto, colocamos um 5 na coluna "ordem" e marcamos a linha 4 e a coluna 4.

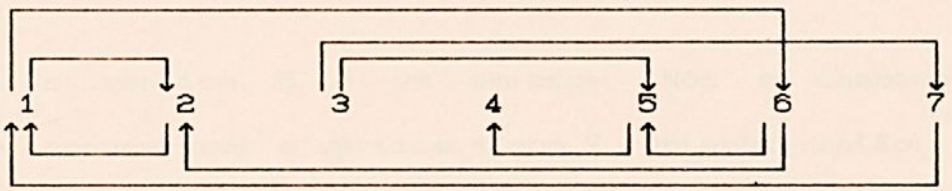
Agora, não existem mais colunas e linha não marcadas. Então podemos ler, a partir das colunas "ordem" e "em", o arranjo dos blocos diagonais. O primeiro bloco contém o elemento (3,3) de $\tilde{B}$; e, o segundo, o elemento (7,7). O terceiro bloco contém os elementos (1,1), (2,2) e (6,6) de $\tilde{B}$ como elementos diagonais. O quarto e o quinto blocos consistem, respectivamente, dos elementos (5,5) e (4,4). Assim, a matriz de permutação $\tilde{P}$ da equação (3.4.2) é dada por

$\tilde{P} = (e_9, e_7, e_1, e_2, e_6, e_5, e_4)$ e $\hat{B}$ por

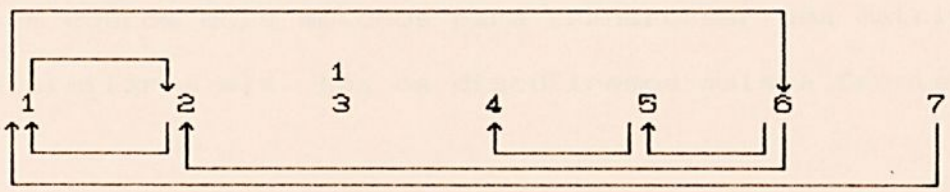
$$\hat{B} = \begin{bmatrix} \boxed{x} & x & . & . & . & x & . \\ . & \boxed{x} & x & . & . & . & . \\ . & . & \boxed{x} & x & x & . & . \\ . & . & \boxed{x} & x & . & . & . \\ . & . & \boxed{.} & x & x & x & . \\ . & . & . & . & . & \boxed{x} & x \\ . & . & . & . & . & . & \boxed{x} \end{bmatrix} \qquad (3.4.12)$$

Algoritmo (3.4.13):

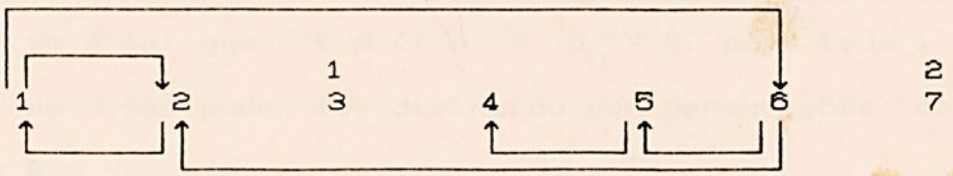
Observamos que se a matriz $\tilde{B}$ é pequena seu dígrafo pode ser usado diretamente para obtermos $\hat{B}$. Assim, por exemplo, o dígrafo correspondente a matriz $\tilde{B}$ de (3.4.10) é dado por



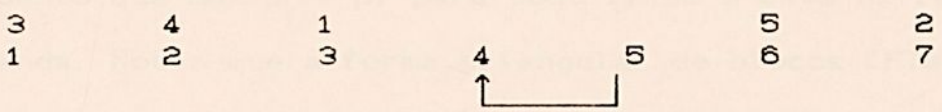
Procedemos então da seguinte forma: Olhamos para um vértice emissor (no qual nenhum arco termina); o vértice 3 é um emissor. Então, nós o renumeramos com 1 e eliminamos todos os arcos que emanam dele. Obtemos:



Agora, 7 transformou-se num emissor. Renumeramos-o com 2 e eliminamos todos os arcos que dele partem. Temos:



Agora não existe nenhum outro emissor. Então os vértices 1, 2 e 6 pertencem a um ciclo. Nós os renumeramos com 3, 4 e 5, respectivamente e eliminamos todos os arcos que partem de qualquer um deles. Obtemos:



Agora, o vértice 5 é um emissor. Nós o chamamos de 6 e finalmente, renumeramos o vértice 4 com 7. Obtemos, então:

3	4	1	7	6	5	2
1	2	3	4	5	6	7

e portanto

$$\tilde{P} = (e_9, e_7, e_1, e_2, e_6, e_5, e_4) \ ,$$

como tinhamos obtido anteriormente.

Existem outros dois métodos para transformar uma matriz em FTB ou numa forma similar a ela. Nós os discutiremos mais a frente.

3.5. FORMA TRIANGULAR DE BANDA (FTBa)

Dizemos que uma matriz B está na Forma Triangular de Banda (FTBa) se existir um β tal que $0 < \beta << n$ e $b_{ij} = 0$, para todo $i - j > \beta$, e este valor de β não puder ser diminuído por permutações de linhas e colunas de B.

Supondo que o primeiro elemento não nulo na i-ésima linha de B está na q_i-ésima coluna, definimos

$$\left.\begin{array}{l} \beta_i \\ \end{array}\right\} \begin{array}{lll} = i - q_i & , & q_i \leq i \\ = 0 & , & q_i > i \ . \end{array} \qquad (3.5.1)$$

É evidente que $\max \beta_i = \beta$, para todo i, se B está na forma triangular de banda. Notar que a forma triangular de blocos (FTB) descrita anteriormente é um caso especial de FTBa (neste caso, $\beta + 1$ é a medida do maior bloco diagonal de B).

Aqui, vamos assumir que, se possível, a matriz dada foi transformada numa FDB por algum dos métodos descritos anteriormente. Se isso foi feito, então consideramos os blocos diagonais de B, um de cada vez. Portanto não haverá perda de generalidade se assumirmos que o grafo de B é conectado.

A seguir, apresentamos um método que por permutações das linhas e colunas transformam B numa matriz triangular de banda.

Seja P e Q matrizes de permutação tais que

$$\hat{B} = P B Q$$

onde a i-ésima linha e a j-ésima coluna de B tornar-se-ão a ρ_i-ésima e a μ_j-ésima coluna de $\hat{B}$, respectivamente. Nosso problema é encontrar

as matrizes de permutação P e Q que minimizam

$$\max_i \phi_i ,$$

onde ϕ_i é a distância entre a diagonal principal e o elemento não nulo mais à esquerda da linha ρ_i de $\hat{B}$. Observamos que $\phi_i = 0$, se não existirem elementos não nulos à esquerda da diagonal na linha ρ_i de $\hat{B}$.

Este problema pode ser resolvido como segue (Chen, 1972):

Tomemos os elementos não nulos da linha i de B que pertencem às colunas $j_{i\alpha}$, $\alpha = 1, 2, \ldots, \gamma_i$; onde γ_i é o número total de elementos não nulos na linha i de B. Se o elemento mais à esquerda da diagonal na linha ρ_i de $\hat{B}$ está na coluna $\mu_{j_{is}}$, então

$$\rho_i - \mu_{j_{is}} = \phi_i , \qquad \rho_i \geq \mu_{j_{is}} ,$$

e para todo $\mu_{j_{i\alpha}}$ não à direita da coluna ρ_i, $\rho_i - \mu_{j_{i\alpha}} \leq \phi_i$. Se fixarmos $\phi_i \geq 0$, para todo i, então para algum elemento não nulo à direita da diagonal, digamos, na coluna $\mu_{j_{it}}$, temos $\rho_i - \mu_{j_{it}} < 0$, pois $\rho_i < \mu_{j_{it}}$.

Portanto $\rho_i - \mu_{j_{i\alpha}} \leq \phi_i$, para todo α.

Do acima exposto temos o seguinte problema:

Em correspondência a todo $b_{ij} = 1$, devemos encontrar uma seqüência de inteiros ρ_i, μ_j e ϕ_i que minimizem $\max_i \phi_i$ sujeitos às condições:

$$1 \leq \rho_i , \qquad \mu_j \leq n ; \qquad 0 \leq \phi_i \leq n - 1 ;$$

$$\rho_i - \mu_{j_{i\alpha}} - \phi_i \leq 0 ; \qquad \alpha = 1, 2, \ldots, \gamma_i$$

e

$$\rho_i \neq \rho_j , \qquad \mu_i \neq \mu_j , \qquad \text{para } i \neq j.$$

O problema (de Chebyshev) sujeito a estas condições pode ser expressado como um problema de programação inteira (Rabinowitz, 1968)

e resolvido de maneira usual (Dantzig, 1963).

3.6. FORMA DE BANDA (FBa)

Em muitas aplicações a matriz A é simétrica e positiva definida, sendo vantajoso escolher para pivôs os elementos da diagonal principal de A para preservar a simetria. Neste caso, realizamos as mesmas permutações de linhas e colunas de A para transformá-la numa forma desejável para a eliminação de Gauss.

São equações importantes as (3.2.2) e

$$P\ B\ P^t = \hat{B}.$$

Se temos

$$\beta_i = i - q_i \ , \qquad q_i \leq i \ ,$$

onde $\hat{b}_{i q_i}$ é o elemento igual a 1 mais à esquerda na linha i de $\hat{B}$, então desejamos determinar o mínimo β e um valor de P tal que

$$\beta = \min_{P} \max_{i} \beta_i \ .$$

Este problema pode ser expressado como um problema de Chebyshev de modo similar ao do método que transforma B numa FTBa visto anteriormente. Não o descreveremos aqui por não ser, em geral, um método prático. A seguir, descreveremos quatro métodos práticos para permutar linhas e colunas de uma dada matriz para transformá-la numa forma de banda (FBa). Para matrizes que podem ser permutadas para matrizes sem zeros dentro da banda, estes métodos produzem tais matrizes. Nos casos em que isto não é possível, os métodos levam a uma matriz com seu $\max_i \beta_i$, razoavelmente próximo do β mínimo.

Primeiro método.

Este é um método especialmente útil nos casos em que a numeração original do sistema espacial de vértices do grafo rotulado determina a largura da banda da matriz correspondente (por exemplo, sistemas estruturais de elementos finitos, sistemas de condutores de escoamento de fluídos, networks elétricos, sistemas de diferenças finitas).

O método de renumeração de vértices de um grafo para a redução da largura da banda da matriz correspondente, cujo algoritmo descreveremos abaixo, foi mostrado por Rosen, 1968.

Passo 1. Num conjunto unidimensional VL (lista de vértices) de n posições coloque os inteiros 1, 2, ... , n.

Determine β da matriz B e dois vértices nos quais ele ocorra. (Se $\max_i \beta_i = \beta_p$, então $(p, p - \beta_p)$ constitui o par de vértices). Se houver empate, escolha o primeiro par de vértices. Se o vértice de maior índice deste par pode ser trocado com outro de índice menor para reduzir a largura da banda ($\beta_i + (p - i) < \beta_p$, para algum $i < p$), então vá para o Passo 7.

Passo 2. Se o vértice de menor índice pode ser trocado por um de maior índice para diminuir a largura da banda, vá para o Passo 7.

Passo 3. Se um vértice de maior índice pode ser trocado por um de menor índice e ainda conservar a largura da banda, vá para o Passo 6.

Passo 4. Se um vértice de menor índice pode ser trocado por um de maior índice e ainda conservar a largura da banda, vá para o Passo 6.

Obs.: O propósito dos passos 3 e 4 é reorganizar o padrão de vértices sem aumentar a largura da banda de modo que mudanças adicionais sucessivas possam ser realizadas nos Passos 1 e 2.

unesp

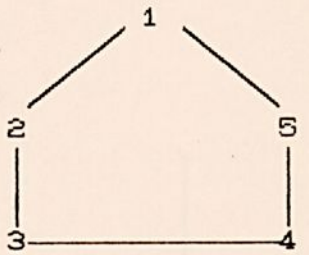
Passo 5. Agora, nenhuma outra troca é possível, e a matriz B está na forma de banda. Para j = 1, 2, ... , n, $e_{VL(j)}$ é a j-ésima coluna de P, onde VL(j) é o inteiro na j-ésima posição de VL. Então $\hat{A} = P\ A\ P^t$. Pare.

Passo 6. Se um número máximo de trocas contínuas foi realizado nos Passos 3 e 4 ou se dois vértices que já tinham sido trocados são novamente selecionados para a troca, vá para o Passo 5.

Passo 7. Realize as mudanças de vértices indicadas, isto é, troque os correspondentes elementos de VL e as linhas e colunas de B e vá para o Passo 1.

Exemplo:

Seja $B = \begin{pmatrix} x & x & . & . & x \\ x & x & x & . & . \\ . & x & x & x & . \\ . & . & x & x & x \\ x & . & . & x & x \end{pmatrix}$ com banda 9, cujo grafo Ω é

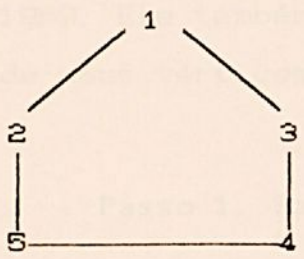


Passo 1. Temos VL = (1 2 3 4 5) ;

$\beta_1 = 0$, $\beta_2 = 1$, $\beta_3 = 1$, $\beta_4 = 1$, $\beta_5 = 4$. Então, $\max_i \beta_i = 4$ e o par de vértices escolhido é (5,1).

Verificando $\beta_i + (p - 1) < \beta_p$, são verdadeiras as sentenças $\beta_3 + (p - 3) < \beta_5$ e $\beta_4 + (p - 4) < \beta_5$. Como houve empate escolhemos o vértice 3 e vamos para o Passo 7.

Passo 7. Agora temos VL = (1 2 5 4 3) e o grafo Ω é



e então $\hat{B}_1 =$

$$\begin{pmatrix} x & x & x & . & . \\ x & x & . & . & x \\ x & . & x & x & . \\ . & . & x & x & x \\ . & x & . & x & x \end{pmatrix}$$

que possui banda 7.

Voltamos agora ao Passo 1. Temos

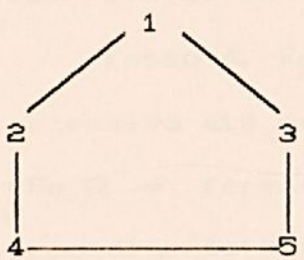
$\beta_1 = 0$, $\beta_2 = 1$, $\beta_3 = 2$, $\beta_4 = 1$, $\beta_5 = 3$. Então $\max_i \beta_i = 3$ e o par de vértices escolhido será (5,2).

A desigualdade $\beta_4 + (p - 4) < \beta_5$ é verdadeira.

Passo 2.

Trocamos os vértices 4 e 5 do grafo Ω.

Passo 7. VL = (1 2 4 5 3) e o grafo Ω é



Então $\hat{B}_2 =$

$$\begin{pmatrix} x & x & x & . & . \\ x & x & . & x & . \\ x & . & x & . & x \\ . & x & . & x & x \\ . & . & x & x & x \end{pmatrix}.$$

Passo 1. Temos $\beta_1 = 0$, $\beta_2 = 1$, $\beta_3 = 2$, $\beta_4 = 2$, $\beta_5 = 2$ e portanto $\max_i \beta_i = 2$.

Não é mais possível efetuar outras trocas. Então $\hat{A} = \hat{A}_2$ que tem banda 5.

Segundo método.

Vamos agora descrever o algoritmo mostrado por Cuthill e Mckee, 1969. Ele também reduz a largura da banda da matriz pela renumeração de seus vértices.

Passo 1. Para cada vértice i do grafo Ω associado à matriz B, calcule seus graus ρ_i (ρ_i é o número de 1's da i-ésima linha de B fora da diagonal). Escolha um vértice i_1 tal que $\rho_{i_1} = \min_i \rho_i$ e o renumere com 1.

Passo 2. Renumere os vértices adjacentes ao vértice 1, em seqüência, começando com 2, em ordem crescente de seus graus (se há empate, escolha o primeiro). Estes vértices são chamados de primeiro nível.

Passo 3. Repita este procedimento para cada vértice do primeiro nível em seqüência; isto é, primeiro para o vértice 2, depois para o 3, etc. Os novos vértices renumerados possuem caminho de comprimento 2 com relação ao vértice 1 e portanto são chamados de segundo nível.

Passo 4. Repita este procedimento para os vértices de cada nível sucessivo até que todos os n vértices de Ω tenham sido renumerados. (Se Ω é formado por dois ou mais subgrafos disjuntos então o procedimento termina quando todos os vértices de um subgrafo tenham sido renumerados. Neste caso, escolha um vértice inicial em cada um dos subgrafos e repita os passos 2, 3 e 4 para cada um deles).

Passo 5. Finalmente, permute as linhas e colunas de B (ou A) de acordo com a renumeração dos vértices para obter $\hat{B}$ (ou $\hat{A}$).

Este esquema de renumeração em geral não minimiza a largura da banda. Mas é possível gerar outros conjuntos para a renumeração de

vértices começando com um diferente vértices i_1 no Passo 1 tal que:

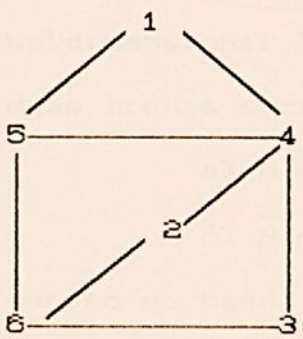
$$\rho_{min} \leq \rho_{i_1} \leq \rho_{min} + \frac{1}{2}\rho_{max}$$

onde ρ_{min} e ρ_{max} são, respectivamente, o menor e o maior valor de ρ_i; e, quando houver empate em algum dos passos sucessivos escolhidos, deve-se escolher uma ordem diferente de renumeração dos vértices daquele nível. O conjunto que deixa menor a largura da banda é então escolhido para permutar B para $\hat{B}$. Certamente é possível gerar somente uns poucos destes conjuntos; por outro lado este é um processo demorado.

Exemplo:

Seja $B = \begin{pmatrix} x & . & . & x & x & . \\ . & x & . & x & . & x \\ . & . & x & x & . & x \\ x & x & x & x & x & . \\ x & . & . & x & x & x \\ . & x & x & . & x & x \end{pmatrix}$ que possui banda 9.

Seu grafo e os graus correspondentes a cada um de seus vértices são:

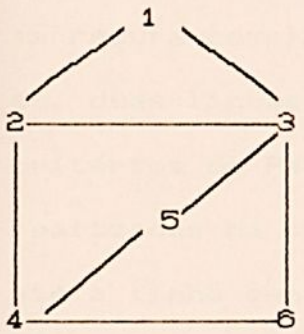


vértice	1	2	3	4	5	6
grau	2	2	2	4	3	3

Portanto $\rho_{i_1} = \min_i \rho_i = 2$

Então os vértices 1, 5 e 4 pertencem ao primeiro nível. Renume-

rando-os obtemos o grafo



Então $\hat{B} = \begin{pmatrix} \times & \times & \times & . & . & . \\ \times & \times & \times & \times & . & . \\ \times & \times & \times & . & \times & \times \\ . & \times & . & \times & \times & \times \\ . & . & \times & \times & \times & . \\ . & . & \times & \times & . & \times \end{pmatrix}$

que possui banda 7.

Terceiro método.

Este é um método iterativo que minimiza a largura da banda média

$$\bar{\beta} = \frac{1}{n} \sum_{i=1}^{n} \beta_i ,$$

onde β é dado por (3.5.1) e também é baseado na troca de linhas ou colunas como no primeiro método. A seguir descreveremos seu algoritmo que foi descrito por Akyuz e Utku, 1968.

Passo 1. Coloque os inteiros 1, 2, ... , n num conjunto unidimensional IL (lista de índices de linha) de n posições. Troque duas linhas sucessivas (e as correspondentes colunas) de B se

a) $\bar{\beta}$ diminui,

b) $\bar{\beta}$ continua o mesmo mas a linha que possui mais zeros dentro da banda saiu da linha central de B.

Registre estas trocas de IL. Ou seja, toda vez que duas linhas são trocadas, seus correspondentes elementos em IL são também

trocados.

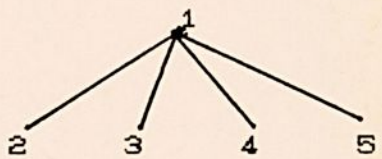
Passo 2. Efetue as trocas por comparação de linhas fazendo uma varredura completa, que consiste de (n - 1) comparações. Em cada passo, duas linhas sucessivas são comparadas e é feita uma troca se os critérios do Passo 1 são satisfeitos. As comparações de linhas são realizadas na seguinte ordem: (1,2), (n,n-1), (2,3), (n-1,n-2), ... até a linha central.

Passo 3. Pare quando nenhuma troca for efetuada ou quando não ocorrer diminuição de $\bar{\beta}$ em (3 + n/100) varreduras (este limite de varreduras foi empiricamente determinado pelos autores). Para j = 1, 2, ... , n , $e_{IL(j)}$ é a j-ésima coluna de P, onde IL(j) é o inteiro na j-ésima posição da lista IL.

Exemplo: A seguir, apresentamos um exemplo da aplicação deste método de redução da banda de uma matriz. Observe que a banda da matriz é 9.

$$B = \begin{pmatrix} x & x & x & x & x \\ x & x & . & . & . \\ x & . & x & . & . \\ x & . & . & x & . \\ x & . & . & . & x \end{pmatrix}$$

cujo Ω é



temos $\beta_1 = 0$

$\beta_2 = 1$

$\beta_3 = 2$

$\beta_4 = 3$

$\beta_5 = 4$

$$\bar{\beta} = \frac{1}{5}\left(0 + 1 + 2 + 3 + 4\right) = \frac{10}{5} = 2$$

Iniciando-se a varredura:

Passo 1. Comparando as linhas 1 e 2 obtemos

$$B_{(1,2)} = \begin{pmatrix} x & x & . & . & . \\ x & x & x & x & x \\ . & x & x & . & . \\ . & x & . & x & . \\ . & x & . & . & x \end{pmatrix} \qquad \begin{aligned} \beta_1 &= 0 \\ \beta_2 &= 1 \\ \beta_3 &= 1 \\ \beta_4 &= 2 \\ \beta_5 &= 3 \end{aligned}$$

$$\bar{\beta} = \frac{1}{5}\left(0 + 1 + 1 + 2 + 3\right) = \frac{7}{5}$$

Troca-se as linhas e colunas 1 e 2, pois $\bar{\beta}$ diminue, e portanto

$$\hat{B} = \begin{pmatrix} x & x & . & . & . \\ x & x & x & x & x \\ . & x & x & . & . \\ . & x & . & x & . \\ . & x & . & . & x \end{pmatrix}$$

cuja largura da banda média $\left(\bar{\beta} = \frac{7}{5}\right)$ é menor que o $\bar{\beta}$ da matriz inicial.

Aplicando-se novamente o método

Comparando-se as linhas 5 e 4 temos

$$B_{(5,4)} = \begin{pmatrix} x & x & . & . & . \\ x & x & x & x & x \\ . & x & x & . & . \\ . & x & . & x & . \\ . & x & . & . & x \end{pmatrix} \qquad \begin{aligned} \beta_1 &= 0 \\ \beta_2 &= 1 \\ \beta_3 &= 1 \\ \beta_4 &= 2 \\ \beta_5 &= 3 \end{aligned}$$

$$\bar{\beta} = \frac{1}{5}\left(0 + 1 + 1 + 2 + 3\right) = \frac{7}{5}$$

Não há trocas pois não satisfaz os critérios do Passo 1.

unesp

Comparando-se as linhas 2 e 3, obtemos

$$B_{(2,9)} = \begin{pmatrix} x & . & x & . & . \\ . & x & x & . & . \\ x & x & x & x & x \\ . & . & x & x & . \\ . & . & x & . & x \end{pmatrix} \qquad \begin{aligned} \beta_1 &= 0 \\ \beta_2 &= 0 \\ \beta_3 &= 2 \\ \beta_4 &= 1 \\ \beta_5 &= 2 \end{aligned}$$

$$\bar{\beta} = \frac{1}{5} \cdot \left(0 + 0 + 2 + 1 + 2 \right) = \frac{5}{5} = 1$$

Troca-se as linhas 2 e 3, pois $\bar{\beta}$ diminue.

Agora temos

$$\hat{B} = \begin{pmatrix} x & . & x & . & . \\ . & x & x & . & . \\ x & x & x & x & x \\ . & . & x & x & . \\ . & . & x & . & x \end{pmatrix}$$

cuja largura da banda média $\left(\bar{\beta} = 1 \right)$ é menor que o da matriz $\hat{B}$ anterior.

Observe que além disso $\hat{B}$ possui banda 5.

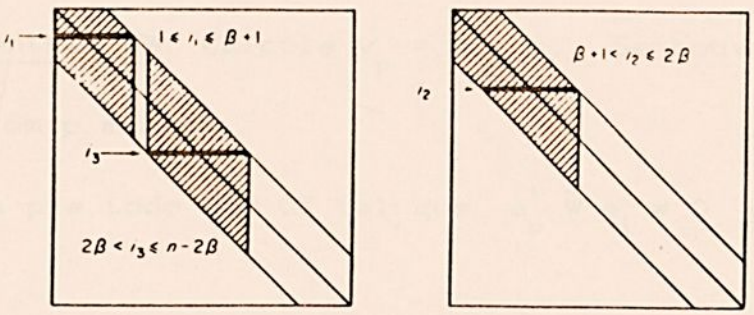
Quarto método.

Ao produto usual de duas linhas de uma matriz B, chamaremos de comprimento de sua intersecção. Denotaremos v_i à soma dos comprimentos de intersecção da linha i com todas as demais linhas de B.

Se assumirmos que B é uma matriz de banda com largura da banda β, então o número de 1's nas áreas sombreadas das figuras a seguir representa os valores de v_i para três conjuntos de i's.

A primeira área sombreada (associada a i_1) cresce com i_1 (onde

$1 \leq i_1 \leq \beta + 1$). O mesmo ocorre com a área sombreada associada a i_2, mas a área sombreada associada a i_3 permanece a mesma quando i_3 aumenta de $2\beta + 1$ até $n - 2\beta$. Além disso, todas as áreas associadas a i_1 são menores que as associadas a i_2 que, por sua vez, são menores que as correspondentes a i_3. Por isso, se assumirmos que os 1's não diagonais de B dentro da banda são randonicamente distribuídos, então os valores dos v_i's geralmente aumentam quando i aumenta de 1 até $2\beta + 1$ e depois permanecem os mesmos até i igual a $n - 2\beta$. Da simetria de B, podemos concluir que os v_i's geralmente diminuem quando i aumenta de $n - 2\beta + 1$ até n.



A seguir, apresentamos o método segundo o algoritmo de Tewarson, 1971.

Dada B simétrica com 1's na diagonal.

1. Calcule $\tilde{W} = B^2$ (produto usual de matrizes).

Então, o elemento (i,j) de $\tilde{W}$ é o comprimento da intersecção da linha i com a coluna j e a soma v_i dos comprimentos das intersecções da linha i com as linhas de B é dado por:

$$v_i = e_i^t \tilde{W} V ,$$

onde V é o vetor coluna de ordem n formado por 1's.

2. Calcule $\beta = \max \left[\frac{\tau - n}{2n} , \frac{\rho_{max} - 1}{2} \right]$, onde

τ é o número de 1's em B e ρ_{max} é o número máximo de 1's em qualquer linha de B.

3. Coloque os v_i's em ordem crescente de grandeza. Denote o conjunto dos índices correspondentes aos primeiros 2β valores dos v_i's por CI (índices de canto da matriz de banda) e os índices restantes por MI (índices de centro).

Divida o conjunto CI em dois subconjuntos NW e SE da seguinte forma:

Para encontrar NW: Calcule $v_p = \min_{i \in CI} v_i$. Se houver coincidência, escolha o v_p com p mínimo.

Transfira p e todo $i \in CI$ tal que $e_p^t \tilde{W} e_i \neq 0$ para o conjunto NW.

Transfira todos os índices $j \in CI$ para os quais $e_i^t \tilde{W} e_j \neq 0$, com $i \in NW$, para NW.

Repita este processo até que nenhum outro índice de CI possa ser transferido para NW.

Para encontrar SE:

Se $CI \neq \emptyset$ calcule $v_p = \min_{i \in CI} v_i$

Se houver coincidência, escolha v_p com máximo p.

Transfira p e todo $i \in CI$ tal que $e_p^t \tilde{W} e_i \neq 0$ para SE.

Transfira todos os índices $j \in CI$ para os quais $e_i^t \tilde{W} e_j \neq 0$, com $i \in SE$, para SE.

Repita este processo até que nenhum outro índice de CI possa ser

transferido para SE.

Transfira para MI os índices de CI que não puderam ser transferidos nem para NW nem para SE, de modo que os v_i's em MI estejam em ordem crescente de grandeza.

Reorganize os índices em NW e em SE de modo que eles estejam respectivamente em ordem crescente e decrescente de valores de v_i's.

4. Transfira os índices de MI, um de cada vez, ou para NW ou para SE da seguinte forma:

a) Transferência para NW: Considere V um vetor coluna de n componentes tal que

$$e_i^t V = \begin{cases} 1, & \text{se } i \in NW \\ 0, & \text{nos outros casos.} \end{cases}$$

Calcule $\hat{v}_p = \max_j \left(e_j^t \tilde{W} V \right)$, para todo $j \in MI$.

1) Se $e_i^t \tilde{W} V$ for nulo para todo $j \in MI$, vá para b,

2) Se houver coincidência, escolha $\hat{v}_p$ com p máximo. Transfira p para NW.

Repita este processo até que aproximadamente a metade dos índices de MI tenha sido transferida para NW. A transferência dos índices de MI para NW deve obedecer a ordem de obtenção dos $\hat{v}_p$'s.

b) Transferência para SE: Considere V um vetor coluna de n componentes tal que

$$e_i^t V = \begin{cases} 1, & \text{se } i \in SE \\ 0, & \text{nos outros casos.} \end{cases}$$

Calcule $\hat{v}_p = \max_j \left\{ e_j^t \tilde{W} V \right\}$, para todo $j \in MI$.

1) Se $e_i^t \tilde{W} V$ for nulo para todo $j \in MI$, vá para 5.

2) Se houver coincidência, escolha $\hat{v}_p$ com p máximo.

Transfira p para o conjunto SE, seguindo a ordem de obtenção de $\hat{v}_p$.

Repita este processo até que todos os índices de MI tenham sido tranferido para SE.

5) Se ainda existirem índices em MI eles devem permanecer em MI em ordem crescente dos valores dos v_i's.

6) Escreva a matriz P de permutação e obtenha $\hat{B} = P\ B\ P^t$.

Exemplo: Seja

$$B = \begin{pmatrix} 1 & 1 & . & . & . & 1 & . \\ 1 & 1 & . & . & 1 & . & . \\ . & . & 1 & . & . & . & . \\ . & . & . & 1 & 1 & . & . \\ . & 1 & . & 1 & 1 & 1 & 1 \\ 1 & . & . & . & 1 & 1 & 1 \\ . & . & . & . & 1 & 1 & 1 \end{pmatrix}$$

com banda 11, $n = 7$, $\sigma = 21$. Então $\beta = \max \left\{ \frac{\sigma - n}{2n}, \frac{\rho_{max} - 1}{2} \right\} = 2$.

Temos

$$B^2 = \begin{pmatrix} 3 & 2 & . & . & 2 & 2 & 1 \\ 2 & 3 & . & 1 & 2 & 2 & 1 \\ . & . & 1 & . & . & . & . \\ . & 1 & . & 2 & 2 & 1 & 1 \\ 2 & 2 & . & 2 & 5 & 3 & 3 \\ 2 & 2 & . & 1 & 3 & 4 & 3 \\ 1 & 1 & . & 1 & 3 & 3 & 3 \end{pmatrix}$$

Então $v_1 = 10$, $v_2 = 11$, $v_3 = 1$, $v_4 = 7$, $v_5 = 17$, $v_6 = 15$, $v_7 = 12$, e portanto V = (3, 4, 1, 2, 7, 6, 5).

Logo: CI = (3, 4, 1, 2) e MI = (7, 6, 5)

Divisão de CI em NW e SE.

- Transferência para NW.

$$v_p = \min_{i \in CI} v_i \longrightarrow v_p = v_3 \longrightarrow p = 3.$$

Transfira p e todo i ∈ CI tal que $e_p^t \tilde{W} e_i \neq 0$ para NW. Como não há i ∈ CI a transferir, teremos:

NW = (3) e CI = (4, 1, 2).

Transfira para NW todo j ∈ CI tal que $e_i^t \tilde{W} e_j \neq 0$ para i ∈ NW. Como não existe j ∈ CI a transferir, repetimos o processo até que nenhum índice de CI possa ser trasferido para NW.

Finalmente obtemos:

NW = (3) e CI = (4, 1, 2).

- Transferência para SE.

$$v_p = \min_{i \in CI} v_i \longrightarrow v_p = v_4 \longrightarrow p = 4$$

Transfira p e todo i ∈ CI tal que $e_p^t \tilde{W} e_i \neq 0$ para SE. Temos $e_4^t \tilde{W} e_2 = 1$. Portanto

SE = (4, 2) e CI = (1).

Transfira para SE todos $j \in CI$ tal que $e_i^t \tilde{W} e_j \neq 0$ para $i \in SE$.

Temos $e_4^t \tilde{W} e_1 = 2$. Portanto

SE = (2, 1, 4) e CI = ∅.

- Transferência de índices de MI para NW.

Seja V tal que $e_i^t V = \begin{cases} 1, & \text{se } i \in NW. \\ 0, & \text{nos outros casos.} \end{cases}$

Então $V = (0\ 0\ 1\ 0\ 0\ 0\ 0)^t$

Mas $\hat{v}_p = \max_{j \in MI} \left(e_j^t \tilde{W} V \right) = 0$, e então nenhum índice será transferido para NW.

- Da mesma maneira, transfira índices de MI para SE.

Seja V tal que $e_i^t V = \begin{cases} 1, & \text{se } i \in NW. \\ 0, & \text{nos outros casos.} \end{cases}$

Então $V = (1\ 1\ 0\ 1\ 0\ 0\ 0)^t$

$\hat{v}_p = \max_{j \in MI} \left(e_j^t \tilde{W} V \right) \longrightarrow v_p = v_5 \longrightarrow p = 5.$

Agora temos

SE = (2, 1, 4, 5) e MI = (7, 6).

Do mesmo modo, agora

$V = (1\ 1\ 0\ 1\ 1\ 0\ 0)^t$, $v_p = v_6 \longrightarrow p = 6$,

Então SE = (2, 1, 4, 5, 6) e MI = (7, 6).

Finalmente, teremos SE = (2, 1, 4, 5, 6) e nada mais há para ser transferido para SE. Então

$$P = \downarrow \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \underbrace{3}_{NW} & \underbrace{2 & 1 & 4 & 5 & 6 & 7}_{SE} \end{pmatrix} \quad e$$

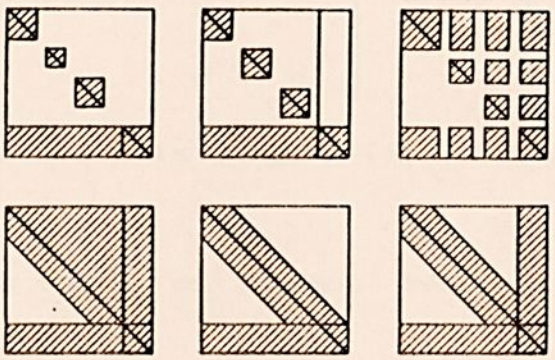
$$B = P\,A\,P^{t} = \begin{pmatrix} 1 & . & . & . & . & . & . \\ . & 1 & 1 & . & 1 & . & . \\ . & 1 & 1 & . & . & 1 & . \\ . & . & . & 1 & 1 & . & . \\ . & 1 & . & 1 & 1 & 1 & 1 \\ . & . & 1 & . & 1 & 1 & 1 \\ . & . & . & . & 1 & 1 & 1 \end{pmatrix}$$

que é de banda 7.

3.7. OUTRAS FORMAS DESEJÁVEIS

A seguir, mostraremos outras formas desejáveis para as quais podemos permutar a matriz dos coeficientes de um sistema de equações lineares.

A forma diagonal de blocos é um caso particular da forma diagonal de blocos de borda única (FDBUBo) e da forma diagonal de blocos de borda dupla (FDBDBo). A forma triangular de blocos (FTB) e a forma triangular de banda (FTBa) são casos particulares da forma bloco triangular de borda (FBTBo) e da forma bloco triangular de banda com borda dupla (FBTBaDBo) respectivamente. A matriz de banda está associada à forma de banda de borda única (FBaUBo) e à forma de banda de borda dupla (FBaDBo).



Descreveremos alguns métodos práticos para tratar formas de borda. O trabalho é razoavelmente bom com matrizes que podem ser permutadas para estas formas de borda, com os elementos não nulos regularmente distribuídos nas áreas sombreadas.

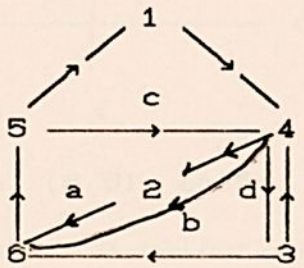
Não existe um algoritmo ideal para se determinar as linhas e/ou colunas que pertencem à borda ou às bordas, onde somente uns poucos elementos não nulos constituem a borda ou as bordas (Harary, 1971).

3.8. MÉTODO DE REMOÇÃO DE ARCOS DO DIGRAFO PARA QUEBRAR GRANDES CICLOS

Este método descrito por Steward, 1969 é especialmente útil quando a remoção de alguns arcos quebra grandes ciclos orientados e os correspondentes blocos diagonais da FTB tornam-se menores. Este método é prático somente para a formação de blocos pequenos. Vamos descrevê-lo através de um exemplo.

Seja a matriz e seu digrafo dados abaixo:

	1	2	3	4	5	6
1	x	.	.	x	.	.
2	.	x	.	.	.	x
3	.	.	x	x	.	x
4	.	x	x	x	.	x
5	x	.	.	x	x	.
6	.	.	.	.	x	x



Um de seus ciclos mais longo é [1,4,3,6,5,1].

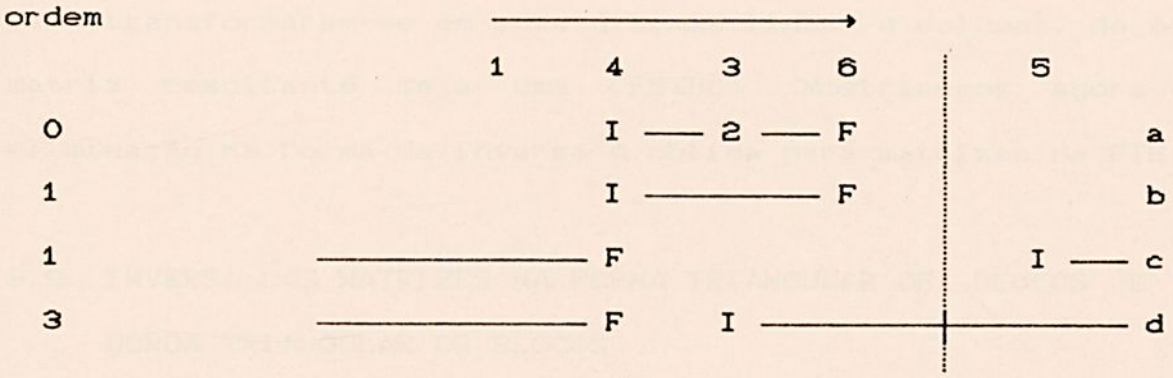
Dois caminhos entre os mesmos vértices, orientados na mesma

direção, são ditos paralelos se não contém ciclos e não possuem arcos em comum. Um caminho paralelo para um arco no ciclo longo é denominado desvio. No digrafo de nosso exemplo, quatro desvios são possíveis: a, b, c e d.

A ordem de um desvio é o comprimento do caminho desviado no ciclo longo menos o comprimento do desvio. Em nosso exemplo, c é [5,4] e [5,1,4] é o caminho no ciclo longo; então c tem ordem 2 - 1 = 1.

Se um arco no ciclo longo que tem um desvio é rompido, então resta ainda um caminho via desvio que é igual ao comprimento do ciclo longo menos a ordem do desvio. Então, rompemos o ciclo longo onde ele não possui desvios suficientes desde que estes desvios devam também serem rompidos.

O diagrama dos desvios associados ao ciclo longo em nosso exemplo é:



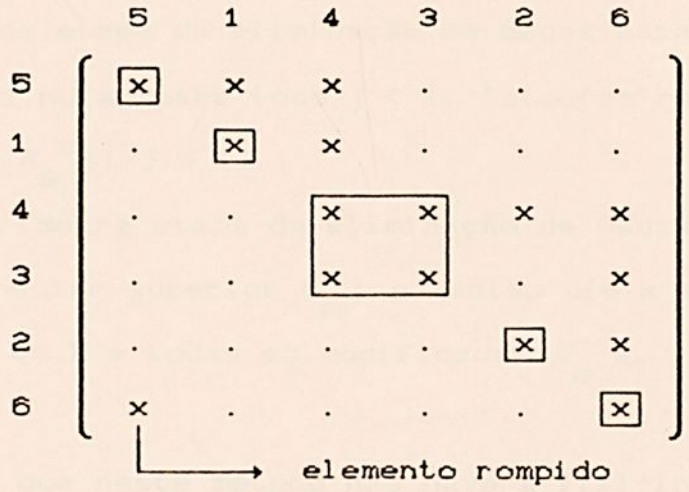
onde I denota início e F o final do desvio.

Fica evidente que se rompermos o arco [6,5] então somente o desvio d precisará ser rompido e por isso esta é a melhor escolha.

No digrafo, se o arco [6,5] e o desvio [3,4] são removidos então o vértice 5 transforma-se num emissor e é renumerado como vértice 1.

Agora, removendo-se os arcos [5,1] e [5,4], o vértice 1 torna-se um emissor e é renumerado como vértice 2. A matriz associada do digra-

fo renumerado é dada por



Esta matriz é uma FTB exceto pelo elemento rompido. Então é um caso especial de FBTBo com a borda formada por apenas um único elemento fora da diagonal.

Quando existem vários elementos a serem rompidos, todas as linhas e colunas da matriz que contem estes elementos podem ser permutadas para transformarem-se em suas últimas linhas e colunas, de modo que a matriz resultante seja uma (FBTBo). Mostraremos agora como a eliminação na forma da inversa é obtida para matrizes na FTB e TBTBo.

3.9. INVERSA DAS MATRIZES NA FORMA TRIANGULAR DE BLOCOS E FORMA DE BORDA TRIANGULAR DE BLOCOS

Considere a FBTBo dada em (3.2.1). Sejam todos A_{ii}, $i = 1, 2, \ldots, p$, matrizes não singulares. O método que descrevemos na forma algoritmica é devido a (Tewarson, 1972; Duff, 1972).

1) Para $i = p-1, p-2, \ldots, 2, 1$ realize os passos:

i) Transforme A_{ii} em uma matriz triangular superior U_{ii} só de 1's e A_{pi} na matriz nula usando a primeira etapa da eliminação de Gauss.

Isto faz com que o fill-in ocorra em A_{ip} e A_{pp}.

ii) Use a segunda etapa da eliminação de Gauss para transformar U_{ii} em I e A_{ji} na matriz nula, para todo $j < i$. Isto faz com que o fill-in ocorra em todas A_{jp}'s, $j \leq i$.

2) Use a primeira etapa da eliminação de Gauss para transformar A_{pp} em uma triangular superior U_{pp}; e, então use a segunda etapa para transformar U_{pp} em I e todas as modificadas A_{jp}'s, $j \neq p$ na matriz nula.

É evidente que neste método não haverá fill-in nas matrizes A_{ji}, $j < i$ e $i \neq p$.

Se a matriz dada é uma FTB então para cada i, $i = p, p-1, \ldots, 2, 1$, necessitaremos:

i) Transformar A_{ii} em U_{ii}.

ii) Reduzir U_{ii} a I e A_{ji} à matriz nula para $j < i$.

Claramente neste caso verificamos que não haverá fill-in em A_{ji}, $j \neq i$. As modificações acima, realizadas no método da eliminação de Gauss, são facilmente incorporadas e as correspondentes EFI são esparsas. Nos dois métodos descritos, quando transformamos A_{ii} em U_{ii} podemos usar os vários métodos já estudados para reduzir o fill-in.

Por fim, observamos que Tewarson, 1972 apresenta uma extensa bibliografia sobre as técnicas aqui descritas, bem como inúmeras referências a programas computacionais desenvolvidos aplicando estas técnicas.

Observação: Se na resolução do sistema $Ax = b$ fizermos transformações sobre A (ou B) para obtermos uma das formas desejáveis estas transformações deverão ser consideradas para a obtenção da solução x

do sistema.

Assim, por exemplo, se A (ou B) for transformada numa FDB, teremos:

$PAx = Pb \rightarrow PAQy = Pb$ onde $x = Qy$ e então $\hat{A}y = Pb$.

Agora, aplicando a eliminação de Gauss, sobre $\hat{A}y = Pb$ obteremos

$\hat{A}^{(n+1)} = L_n L_{n-1} \ldots L_1 \hat{A} = L\hat{A} = U$.

De $Uy = LPb$ temos que $y = U^{-1}LPb$.

Então a solução do sistema $Ax = b$ é $x = Qy$.

Para ilustrar, apresentamos a seguir um exemplo de resolução de sistema onde a matriz A é transformada numa FTB pelo Algoritmos 3.4.4 e 3.4.13. Como o sistema é pequeno, nós o resolvemos sem usar a eliminação de Gauss.

Exemplo: Seja $Ax = b$ onde

$$A = \begin{bmatrix} 2 & . & 4 & . & . \\ 1 & 1 & . & . & 3 \\ . & . & 3 & 4 & . \\ 2 & 2 & . & 1 & 5 \\ . & . & 1 & 1 & . \end{bmatrix} \quad e \quad A = \begin{bmatrix} 10 \\ -6 \\ 6 \\ -10 \\ 2 \end{bmatrix}$$

a) Determinemos $\tilde{B} = \hat{P}\, B\, \hat{Q}$ usando o Algoritmo 3.4.4. Temos

$$B_1 = B = \begin{bmatrix} 1 & . & 1 & . & . \\ 1 & 1 & . & . & 1 \\ . & . & 1 & 1 & . \\ 1 & 1 & . & 1 & 1 \\ . & . & 1 & 1 & . \end{bmatrix}$$

$V_1 = (1 \quad 1 \quad 1 \quad 1 \quad 1)^t$ e $U_1 = V_1^t$.

$P_1 = Q_1 = I$, $k = 1$

Como $\min_{i,j}(e_i^t BV_1 + U_1 Be_j)$ ocorre em b_{11}, temos $(\alpha_1, \beta_1) = (1,1)$.

Então $V_2 = (0\ 1\ 1\ 1\ 1)^t$, $U_2 = V_2^t$, $P_2 = P_1$ e $Q_2 = Q_1$.

Fazendo $k = 2$, temos que o $\min_{i,j\neq 1}(e_i^t BV_1 + U_1 Be_j)$ ocorre em b_{22}.

Portanto $(\alpha_2, \beta_2) = (2,2)$ e então $P_3 = P_2$, $Q_3 = Q_2$, $V_3 = (0\ 0\ 1\ 1\ 1)^t$ e $U_3 = V_3^t$.

Para $k = 3$, temos $\min(e_i^t BV_3 + U_3 Be_j)$; para $i \neq 1,2$ e $j \neq 1,2$; ocorre em b_{45}. Então $(\alpha_3, \beta_3) = (4,5)$.

Agora $P_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 3 & 5 \end{pmatrix}$, $Q_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 5 & 4 & 3 \end{pmatrix}$

$V_4 = (0\ 0\ 1\ 0\ 1)^t$ e $U_4 = (0\ 0\ 1\ 1\ 0)$.

Fazendo $k = 4$, temos que o $\min(e_i^t BV_3 + U_3 Be_j)$; para $i \neq 1,2,4$ e $j \neq 1,2,5$; ocorre em b_{33}. Então $(\alpha_4, \beta_4) = (3,3)$.

Portanto $P_5 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$, $Q_5 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 5 & 3 \end{pmatrix}$

$V_5 = (0\ 0\ 0\ 0\ 1)^t$ e $U_5 = (0\ 0\ 0\ 1\ 0)$.

Para $k = 5$ temos $(\alpha_5, \beta_5) = (4,4)$.

Portanto $P_6 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 5 & 4 \end{pmatrix}$, $Q_6 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 3 & 5 \end{pmatrix}$

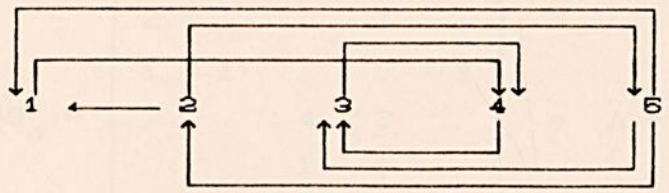
Como agora $k = 6 = n+1$ então

$$\tilde{B} = P_6 B Q_6 = \begin{pmatrix} 1 & . & . & 1 & . \\ 1 & 1 & . & . & 1 \\ . & . & 1 & 1 & . \\ . & . & 1 & 1 & . \\ 1 & 1 & 1 & . & 1 \end{pmatrix}.$$

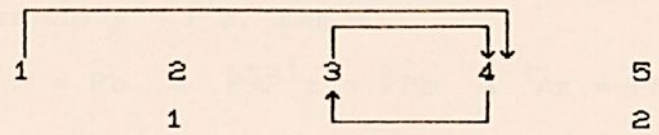
Então $\tilde{P} = P_6$ e $\hat{Q} = Q_6$.

b) Determinemos $\hat{B} = \tilde{P}\tilde{B}\tilde{P}$ pelo Algoritmo 3.4.13.

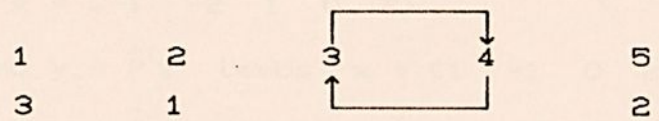
O digrafo correspondente a $\tilde{B}$ é



Não há emissor. Os vértices 2 e 5 pertencem a um ciclo. Nós os renumeramos com 1 e 2, respectivamente; e, eliminamos todos os arcos que deles partem. Obtemos



Agora, o vértice 1 é emissor. Nós chamamos de 3 e obtemos



Os vértices 3 e 4 formam um ciclo. Nós os renumeramos com 4 e 5, respectivamente. Obtemos

1	2	3	4	5
3	1	4	5	2

Assim, as componentes fortes de $\tilde{B}$ são

$$\left\{ \{e_2, e_5\} , \{e_1\} , \{e_3, e_4\} \right\} \text{ e então } \tilde{P} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 1 & 3 & 4 \end{pmatrix}$$

Portanto

$$\hat{B} = \tilde{P}\tilde{B}\tilde{P}^t = \begin{pmatrix} \boxed{\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}} & 1 & . & . \\ . & \boxed{1} & . & 1 \\ . & . & \boxed{\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}} & \\ . & . & . & \end{pmatrix}$$

que está na forma triangular de blocos (FTB).

c) Temos então

$$\tilde{A} = \hat{P}_\sigma A \hat{Q}_\sigma = \begin{pmatrix} 2 & . & . & 4 & . \\ 1 & 1 & . & . & 3 \\ . & . & 4 & 3 & . \\ . & . & 1 & 1 & . \\ 2 & 2 & 1 & . & 5 \end{pmatrix} \quad e \quad \hat{A} = \tilde{P}\tilde{A}\tilde{P}^t = \begin{pmatrix} 1 & 3 & 1 & . & . \\ 2 & 5 & 2 & 1 & . \\ . & . & 2 & . & 4 \\ . & . & . & 4 & 3 \\ . & . & . & 1 & 1 \end{pmatrix}$$

Em $Ax = b$, fazendo $x = \hat{Q}y$ temos

$A\hat{Q}y = b \rightarrow \hat{P}A\hat{Q}y = \hat{P}b \rightarrow \tilde{A}y = \hat{P}b.$

Fazendo $y = \tilde{P}^t z$, temos

$\tilde{A}\tilde{P}^t z = \hat{P}b \rightarrow \tilde{P}\tilde{A}\tilde{P}^t z = \tilde{P}\hat{P}b \rightarrow \tilde{A}z = \hat{P}b.$

Resolvendo o sistema $\hat{A}z = Pb$ onde $Pb = \tilde{P}(\hat{P}b) = (-6\ -10\ 10\ 6\ 2)^t$ obtemos $z = (-1\ \ -2\ \ 1\ \ 0\ \ 2)^t$.

Como $y = \tilde{P}^t z$ temos $x = (1\ \ -1\ \ 0\ \ 2\ \ -2)^t$.

Como $x = \hat{Q}y$ temos $x = (1\ \ -1\ \ 2\ \ 0\ \ -2)^t$ que é a solução do sistema dado.

Inst. de Biociências, Letras e Ciências Exatas - UNESP
BIBLIOTECA
S. J. Rio Preto

CAPÍTULO 4

OUTROS MÉTODOS EXATOS PARA RESOLUÇÃO DE SISTEMAS LINEARES ESPARSOS ESPECIAIS

4.1. INTRODUÇÃO

Este capítulo é dedicado a métodos exatos para a resolução de sistemas lineares esparsos especiais.

Assim, apresentamos métodos usados quando a matriz dos coeficientes é simétrica e positiva definida; de banda simétrica e positiva definida; tridiagonal; e, tridiagonal por blocos.

Também apresentamos um método denominado Partição PAQ que pode ser visto como um caso especial dos métodos de permutação à priori de linhas e colunas apresentado no capítulo 3. Este método determina as permutações P e Q que particionam a matriz dos coeficientes em suas componentes fortes.

4.2. SISTEMAS SIMÉTRICOS E POSITIVOS DEFINIDOS

Definição (4.2.1): A matriz simétrica A, quadrada de ordem n, é dita positiva definida se sua correspondente forma quadrática for positiva definida; isto é, se

$$(Ax,\ x) = \sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik}\, x_i\, x_k > 0, \quad \forall\ x \in \mathbb{R}^n,\ x \neq 0.$$

Uma matriz simétrica positiva definida pode ser decomposta em $A = LDL^t$. Entretanto, neste caso, é melhor fazermos a decomposição

abaixo, cuja demonstração pode ser encontrada em [Golub, 1985].

Teorema (4.2.2): Se a matriz A é simétrica positiva definida então existe uma única matriz triangular inferior G tal que $A = GG^t$. Os elementos diagonais de G são dados por

$$g_{11} = \sqrt{a_{11}}$$

$$g_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} g_{ik}^2 \right)^{1/2} , \quad i = 2, 3, \ldots, n$$

e os não diagonais por

$$g_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} g_{ik}\, g_{jk} \right) / g_{jj} , \quad i = 2, 3, \ldots, n$$
$$j = 1, 2, \ldots, n-1, \; i \neq j$$

Utilizando-se estas fórmulas numa ordem conveniente (por exemplo, por colunas de G) determinamos todos os elementos g_{ij}. Este método é conhecido como **Método de Cholesky**.

Observamos que se a matriz A é simétrica, positiva definida então o método de Cholesky é o melhor para a decomposição triangular (Wilkinson - 1965). As trocas de linhas ou colunas que em outros métodos são realizadas para evitar ou mesmo eliminar os erros de arredondamento não são necessárias em Cholesky. Alias, elas não podem ser realizadas em Cholesky pois destruiriam a simetria da matriz.

Por outro lado , se a matriz A é simetrica mas não positiva definida, o pivotamento parcial poderia ser usado para diminuir razoavelmente os erros de arredondamento mas isto destruiria a simetria. O pivotamento total que utiliza como pivô o maior elemento da diagonal em cada estágio preserva a simetria mas não garante a estabilidade

devido aos erros de arredondamento.

A solução de um sistema linear $Ax = b$, com A simétrica e positiva definida, é obtida pelo método de Cholesky em três passos:

i) $A = GG^t$ (Decomposição de Cholesky),

ii) $Gy = b$ (Resolução de um sistema linear triangular inferior)

iii) $G^t x = y$ (Resolução de um sistema linear triangular superior).

A seguir, apresentamos o algoritmo do método de Cholesky para resolução do sistema $Ax = b$.

Algoritmo (4.2.3): Dados os elementos da matriz A, sobre e abaixo da diagonal, e os elementos do vetor b, fazemos:

Para $k = 1, 2, \ldots, n$:
- Se $a_{kk} \leq 0$: pare
- Caso contrário, $g_{kk} = \sqrt{a_{kk}}$
- Para $i = k + 1, k + 2, \ldots, n$: (I)
  - $g_{ik} = a_{ik} / g_{kk}$
  - Para $j = k + 1, k + 2, \ldots, i$:
    - $a_{ij} = a_{ij} - g_{ik}\, g_{jk}$

$y_1 = b_1 / g_{11}$

Para $i = 2, 3, \ldots, n$:
- $s = b_i$
- Para $j = 1, 2, \ldots, i-1$: (II)
  - $s = s - g_{ij}\, y_j$
- $y_i = s / g_{ii}$,

$x_n = y_n / g_{nn}$

Para $i = n-1, n-2, \ldots, 1$:

    $s = y_i$

    Para $j = i+1, i+2, \ldots, n$:

        $s = s - g_{ji} x_j$

    $x_i = s / g_{ii}$

(III)

Os elementos a_{ik} da matriz são trocados durante o procedimento (I). No entanto, o valor de a_{ik} é necessário pela última vez quando g_{ik} é calculado. Assim, a matriz G pode ser guardada no lugar de A. Para isto, basta identificarmos a variável g com a variável a em (I). Similarmente podemos identificar os vetores b com y em (II) e x com y em (III), de modo que o vetor b é trocado pela solução x.

Como apenas os elementos g_{ij} da diagonal e abaixo dela devem ser calculados, então o k-ésimo passo da decomposição exige a extração de uma raiz quadrada, (n-k) divisões e $(1+2+\ldots+n-k) = (n-k+1)(n-k)/2$ multiplicações.

Então a decomposição completa de A em GG^t requer um total de $S = (n^3 + 3n^2 - 4n)/6$, operações essenciais, se ignorarmos as n extrações de raízes quadradas.

A resolução de um sistema linear por Cholesky, exige um total de $S = \frac{1}{6} n^3 + \frac{3}{2} n^2 + \frac{1}{3} n$ operações essenciais.

Como, ao aplicarmos Cholesky, trabalhamos apenas com a parte inferior das matrizes G e A é conveniente armazenarmos a matriz, linha por linha, na forma de um arranjo unidimensional (vetor); ou seja, na forma

a_{11}	a_{21}	a_{22}	a_{31}	a_{32}	a_{33}	a_{41}	a_{42}	....	a_{nn}

Assim, o armazenamento de A exigirá agora somente $S = (n + 1)n/2$ posições de memória, ou seja, cerca de metade da memória necessária no armazenamento usual. O elemento a_{ik} da matriz A é armazenado na r-ésima posição deste arranjo, sendo $r = i(i-1)/2 + k$.

4.3. SISTEMAS DE BANDA, SIMÉTRICOS E POSITIVOS DEFINIDOS

Definição (4.3.1): A largura da banda m de uma matriz simétrica A de ordem n x n é definida como sendo o menor inteiro $m < n$ tal que:

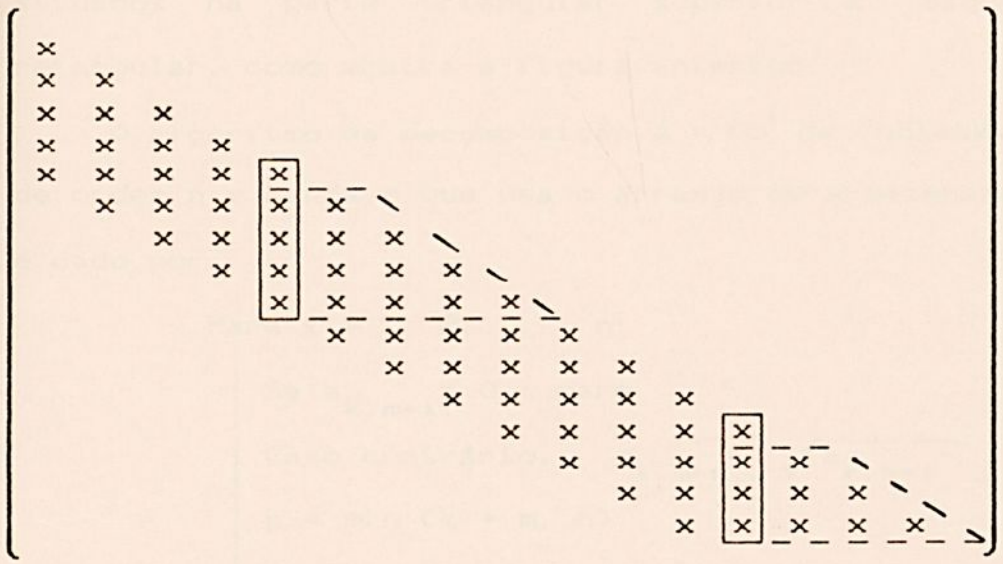
$a_{ik} = 0$ para todo i e k com $|i - k| > m$.

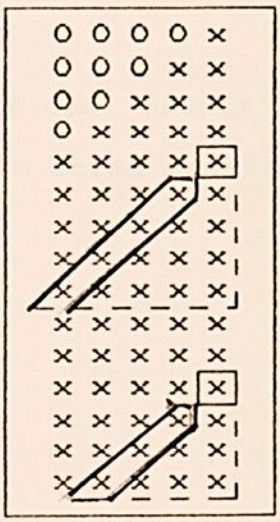
Assim, a largura da banda m indica o número de diagonais secundárias de um lado ou do outro da diagonal principal que contém todos os elementos não nulos.

Teorema (4.3.2): A matriz triangular inferior G da decomposição de Cholesky $A = GG^t$ de uma matriz de banda, simétrica e positiva definida A com largura de banda m tem a mesma estrutura de banda de modo que $g_{ik} = 0$ para todo i e k com $i - k > m$.

Assim, por este teorema, a decomposição de Cholesky de uma matriz de banda, simétrica e positiva definida A pode ser realizada com a matriz G podendo ser armazenada no lugar de A. Além disso, cada passo da redução de Cholesky muda somente aqueles elementos da banda que pertencem a um arranjo triangular que inclue, no máximo, as m linhas seguintes. A figura a seguir ilustra o k-ésimo e o (n-3)-ésimo passos da redução no caso de uma matriz de banda com m = 4. Ilustra também a melhor maneira de se armazenar esta matriz.

unesp





A decomposição completa de Cholesky de uma matriz de banda m e ordem n requer n raízes quadradas e no máximo $nm(m+3)/2$ operações essenciais. A resolução de cada sistema triangular na solução de $Ax=b$ exige no máximo $n(m+1)$ operações. Assim, a resolução de um sistema de n equações lineares cuja matriz dos coeficientes é de banda m, simétrica e positiva definida requer $S \leq nm(m+3)/2 + 2n(m+1)$ operações essenciais.

Ao se armazenar uma matriz de banda simétrica, armazena-se apenas os elementos da banda abaixo e sobre a diagonal principal num arranjo de n linhas e $(m+1)$ colunas como mostra a figura anterior. As diagonais secundárias de A são armazenadas por colunas de modo que a i-ésima linha de A aparece na i-ésima linha do arranjo. Os elementos da diagonal principal são armazenados na coluna $(m+1)$. Assim, um elemento a_{ik} de A com $\max(i-m,1) \leq k \leq i$ pode ser encontrado na $(k - i + m + 1)$-ésima coluna.

É conveniente zerar as posições dos elementos indefinidos

situados na parte triangular superior à esquerda do arranjo retangular, como mostra a figura anterior.

O algoritmo da decomposição $A = GG^t$ de Cholesky de uma matriz A de ordem n e banda m que usa o arranjo de armazenamento acima descrito é dado por:

Para k = 1, 2, ..., n:

- Se $a_{k,m+1} \leq 0$: pare
- Caso contrário, $g_{k,m+1} = \sqrt{a_{k,m+1}}$
- p = min (k + m, n)
- Para i = k + 1, k + 2, ..., p:
  - $g_{i,k-i+m+1} = a_{i,k-i+m+1} / g_{k,m+1}$
  - Para j = k + 1, k + 2, ..., i:
    - $a_{i,j-i+m+1} = a_{i,j-i+m+1} - g_{i,k-i+m+1} g_{j,k-j+m+1}$

SISTEMAS TRIDIAGONAIS

Sistemas de equações com a matriz dos coeficientes A tridiagonal ocorrem em muitas aplicações.

Estes sistemas podem ser tratados de modo simples uma vez que a i-ésima equação somente contém as três incógnitas x_{i-1}, x_i, x_{i+1}. A seguir apresentamos um modo de utilizar a estrutura especial dessa matriz e simplificar a implementação do método de eliminação de Gauss.

Seja o sistema

$$\begin{pmatrix} a_1 & b_1 & & & & & \\ c_1 & a_2 & b_2 & & & & \\ & c_2 & a_3 & b_3 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & c_{n-2} & a_{n-1} & b_{n-1} & \\ & & & & c_{n-1} & a_n & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d_n \end{pmatrix} \qquad (4.3.1)$$

Se o método de Gauss for aplicado sem troca de linhas (por exemplo, quando A é simétrica e positiva definida ou diagonalmente dominante), temos pela decomposição A = LU:

$$\begin{pmatrix} a_1 & b_1 & & & & & \\ c_1 & a_2 & b_2 & & & & \\ & c_2 & a_3 & b_3 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & c_{n-2} & a_{n-1} & b_{n-1} & \\ & & & & c_{n-1} & a_n & \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & & & & & \\ l_1 & 1 & & & & \\ & l_2 & 1 & & & \\ & & \cdot & \cdot & & \\ & & & l_{n-2} & 1 & \\ & & & & l_{n-1} & 1 \end{pmatrix} \begin{pmatrix} m_1 & r_1 & & & & \\ & m_2 & r_2 & & & \\ & & \cdot & \cdot & & \\ & & & m_{n-1} & r_{n-1} & \\ & & & & m_n & \end{pmatrix}$$

Assim, obtemos o

Algoritmo: Dados a matriz dos coeficientes e o vetor do termo independente do sistema linear, fazemos:

$m_1 = a_1$

Para $i = 1, 2, \ldots, n-1$:

$l_i = c_i / m_i$

$m_{i+1} = a_{i+1} - l_i b_i$

Para a resolução de $Ly = d$ fazemos:

$y_1 = d_1$

Para $i = 2, 3, \ldots, n$:

$y_i = d_i - l_{i-1}\, y_{i-1}$

Para a resolução de $Ux = y$ fazemos:

$x_n = y_n / m_n$

Para $i = n-1, n-2, \ldots, 1$

$x_i = (y_i - b_i x_{i+1}) / m_i.$

Por este método, o número de operações essenciais para a resolução do sistema linear é

$S = 2(n - 1) + (n - 1) + 1 + 2(n - 1) = 5n + 4.$

Ou seja, o custo computacional é proporcional somente ao número de incógnitas. Assim, podemos resolver grandes sistemas tridiagonais com um esforço computacional relativamente pequeno.

Apresentaremos a seguir um algoritmo para a resolução do sistema linear tridiagonal (4.3.1), usando o método de eliminação de Gauss com pivotamento parcial. No primeiro passo da eliminação, apenas a_1 e c_1 são candidatos a pivô. Vamos definir dois valores auxiliares

$$\alpha = |a_1| + |b_1|, \quad \beta = |c_1| + |a_2| + |b_2|. \qquad (4.3.2)$$

Se $|a_1| / \alpha \geq |c_1| / \beta$ então a_1 é escolhido para pivô; caso contrário uma troca de linhas é necessária. Neste caso, um elemento não nulo é criado na posição (1,3) que está fora da banda original. Para melhor descrevermos este passo da eliminação, definimos as seguintes quantidades:

$$\text{Se } a_1 \text{ é pivô} \begin{cases} r_1 = a_1 \quad s_1 = b_1 \quad t_1 = 0 \quad f_1 = d_1 \\ u = c_1 \quad v = a_2 \quad w = b_2 \quad z = d_2 \end{cases}$$

$$\text{Se } c_1 \text{ é pivô} \begin{cases} r_1 = c_1 \quad s_1 = a_2 \quad t_1 = b_2 \quad f_1 = d_2 \\ u = a_1 \quad v = b_1 \quad w = 0 \quad z = d_1. \end{cases} \qquad (4.3.3)$$

Com estas variáveis o sistema (4.3.1) escrito na forma

x_1	x_2	x_3			x_n	
a_1	b_1					d_1
c_1	a_2	b_2				d_2
	c_2	a_3	b_3			d_3
		.	.	.		.
		.	.	.		.
		.	.	.		.
		c_{n-2}	a_{n-1}	b_{n-1}		d_{n-1}
			c_{n-1}	a_n		d_n

é transformado em:

x_1	x_2	x_3		x_n	
r_1	s_1	t_1			f_1
u	v	w			z
	c_2	a_3	b_3		d_3
		. . .			
		. . .			
		. . .			
		c_{n-2}	a_{n-1}	b_{n-1}	d_{n-1}
			c_{n-1}	a_n	d_n

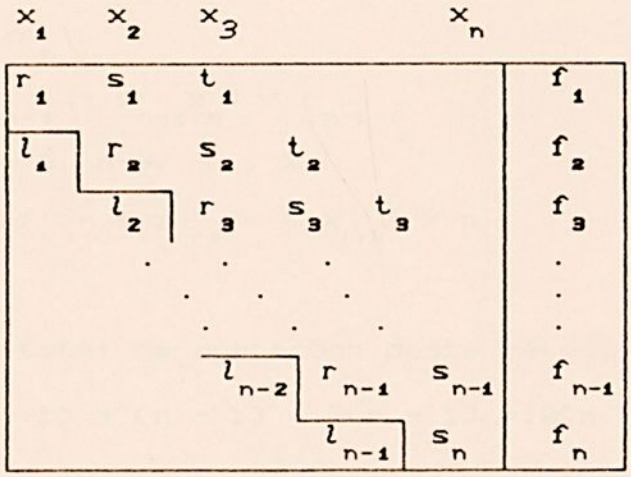
No primeiro passo da eliminação temos

x_1	x_2	x_3		x_n	
r_1	s_1	t_1			f_1
l_1	a'_2	b'_2			d'_2
	c_2	a_3	b_3		d_3
		. . .			
		. . .			
		. . .			
		c_{n-2}	a_{n-1}	b_{n-1}	d_{n-1}
			c_{n-1}	a_n	d_n

onde

$$l_1 = u/r_1 \qquad a'_2 = v - l_1 s_1 \qquad b'_2 = w - l_1 t_1 \qquad d'_2 = z - l_1 f_1$$

A mesma situação se apresenta para o sistema reduzido, pois sua matriz é também tridiagonal. Então, podemos aplicar o mesmo raciocínio para os passos seguintes. Para assegurarmos que as fórmulas (4.3.2) e (4.3.3) também se verifiquem no passo final da eliminação de Gauss, devemos definir $b_n = 0$. Consequentemente a eliminação de Gauss termina com



Assim, o método de eliminação de Gauss com pivotamento parcial para resolução de um sistema tridiagonal é dado pelo algoritmo:

Para $i = 1, 2, \ldots, n-1$:

$\alpha = |a_i| + |b_i|$; $\beta = |c_i| + |a_{i+1}| + |b_{i+1}|$

Se $|a_i| / \alpha \geq |c_i| / \beta$

Então $r_i = a_i$; $s_i = b_i$; $t_i = 0$; $f_i = d_i$;

$u = c_i$; $v = a_{i+1}$; $w = b_{i+1}$; $z = d_{i+1}$

caso contrário, $r_i = c_i$; $s_i = a_{i+1}$; $t_i = b_{i+1}$; $f_i = d_{i+1}$

$u = a_i$; $v = b_i$; $w = 0$; $z = d_i$

$l_i = u/r_i$; $a_{i+1} = v - l_i s_i$

$b_{i+1} = w - l_i t_i$; $d_{i+1} = z - l_i f_i$

$r_n = a_n$; $f_n = d_n$.

As incógnitas x_i são determinadas fazendo-se:

$$x_n = f_n / r_n$$

$$x_{n-1} = (f_{n-1} - s_{n-1} x_n) / r_{n-1}$$

Para $i = n-2, n-3, \ldots, 1$:

$$x_i = (f_i - s_i x_{i+1} - t_i x_{i+2}) / r_i$$

O número total de operações deste método é

$S = 5(n - 1) + (n - 1) + 3(n - 1) = 9(n - 1)$.

Observamos que aqui o número de operações é aproximadamente o dobro do exigido no método anterior que não usava pivotamento.

4.4. SISTEMAS TRIDIAGONAIS DE BLOCOS

Um sistema importante que aparece na discretização de sistemas de equações diferenciais unidimensionais é o tridiagonal por blocos que representaremos na forma:

$$\begin{bmatrix} D_1 & F_1 & & & \\ E_1 & D_2 & F_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & F_{n-1} & \\ & & E_{n-1} & D_n & \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} \qquad (4.4.1)$$

Assumamos que todos os blocos D_i, F_i e E_i têm ordem $q \times q$ e que X_i e B_i são vetores colunas de dimensão q.

O sistema (4.4.1) pode ser resolvido usando a decomposição LU.

Comecemos considerando a decomposição bloco LU para a matriz do sistema (4.4.1). Definimos as matrizes blocos tridiagonais A_k por:

$$A_k = \begin{bmatrix} D_1 & F_1 & & & \\ E_1 & D_2 & F_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & & F_{k-1} \\ & & & E_{k-1} & D_k \end{bmatrix}, \quad k = 1, 2, \ldots, n \qquad (4.4.2)$$

Comparando os blocos em:

$$A_n = \begin{bmatrix} I & & & \\ L_1 & I & & \\ & \ddots & \ddots & \\ & & L_{n-1} & I \end{bmatrix} \begin{bmatrix} U_1 & F_1 & & \\ & U_2 & F_2 & \\ & & \ddots & \ddots \\ & & & U_n \end{bmatrix}$$

obtemos o seguinte algoritmo para a determinação dos blocos L_i e U_i:

$U_1 = D_1$

Para i = 2, 3, ... ,n

  Resolva $L_{i-1}U_{i-1} = E_{i-1}$ para obter L_{i-1} (I)

  $U_i = D_i - L_{i-1}F_{i-1}$

O procedimento acima é definido desde que os blocos U_i sejam não singulares. Isto é assegurado, por exemplo, se as matrizes $D_1, \ldots, D_n$ são não singulares.

Tendo completado a decomposição (I), os blocos X_i do vetor solução de (4.4.1) são obtidos pelo algoritmo:

$Y_1 = B_1$

Para i = 2, 3, ... ,n

  $Y_i = B_i - L_{i-1}Y_{i-1}$ (II)

Resolva $U_n X_n = Y_n$

Para i = n-1, n-2, ... ,1

  Resolva $U_i X_i = Y_i - F_i X_{i+1}$ para obter X_i

Observamos que os sistemas existentes nos algoritmos (I) e (II)

podem ser resolvidos por qualquer método adequado. Por exemplo, podemos usar a eliminação de Gauss com pivotamento para decompor cada bloco U_i. No entanto, isto não garante a estabilidade de todo o processo. Mas, se os blocos D_i são diagonais e os blocos não forem grandes, o método é viável.

Exemplo: Para resolvermos o sistema Ax = b

$$\left(\begin{array}{cc|cc|cc} 1 & 0 & 1 & 1 & 0 & 0 \\ & D_1 & & F_1 & & \\ 0 & 2 & 2 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 3 & 0 \\ & E_1 & & D_2 & & F_2 \\ 3 & 2 & 0 & -1 & 1 & 2 \\ \hline 0 & 0 & -1 & 2 & 2 & 0 \\ & & & E_2 & & D_3 \\ 0 & 0 & 2 & -1 & 0 & 3 \end{array}\right) \left(\begin{array}{c} x_1 \\ x_2 \\ \hline x_3 \\ x_4 \\ \hline x_5 \\ x_6 \end{array}\right) = \left(\begin{array}{c} 3 \\ 2 \\ \hline 5 \\ 0 \\ \hline 0 \\ 1 \end{array}\right)$$

temos:

$$U_1 = D_1$$

$$L_1 U_1 = E_1 \Rightarrow L_1 = \begin{pmatrix} 1 & 1/2 \\ 3 & 1 \end{pmatrix}$$

$$L_1 F_1 + U_2 = D_2 \Rightarrow U_2 = D_2 - L_1 F_1 = \begin{pmatrix} -1 & -3/2 \\ -5 & -5 \end{pmatrix}$$

$$L_2 U_2 = E_2 \Rightarrow L_2 = \begin{pmatrix} -6 & 7/5 \\ 6 & -8/5 \end{pmatrix}$$

$$L_2 F_2 + U_3 = D_3 \Rightarrow U_3 = D_3 - L_2 F_2 = \begin{pmatrix} 93/5 & -14/5 \\ -82/5 & 31/5 \end{pmatrix}$$

e então

$$\left(\begin{array}{cc|cc|cc} 1 & 0 & 1 & 1 & & \\ & D_1 & & F_1 & & 0 \\ 0 & 2 & 2 & 1 & & \\ \hline 1 & 1 & 1 & 0 & 3 & 0 \\ & E_1 & & D_2 & & F_2 \\ 3 & 2 & 0 & -1 & 1 & 2 \\ \hline & & -1 & 2 & 2 & 0 \\ & 0 & & E_2 & & D_3 \\ & & 2 & -1 & 0 & 3 \end{array}\right) =$$

$$\left(\begin{array}{cc|cc|cc} 1 & 0 & & & & \\ & I & & 0 & & 0 \\ 0 & 2 & & & & \\ \hline 1 & 1/2 & 1 & 0 & & \\ & L_1 & & I & & 0 \\ 3 & 2 & 0 & 1 & & \\ \hline & & -6 & 7/5 & 1 & 0 \\ & 0 & & L_2 & & I \\ & & 6 & -8/5 & 0 & 1 \end{array}\right) \left(\begin{array}{cc|cc|cc} 1 & 0 & 1 & 1 & & \\ & U_1 & & F_1 & & 0 \\ 0 & 2 & 2 & 1 & & \\ \hline & & -1 & -9/2 & 3 & 0 \\ & 0 & & U_2 & & F_2 \\ & & -5 & -5 & 1 & 2 \\ \hline & & & & 99/5 & -14/5 \\ & 0 & & 0 & & U_3 \\ & & & & -82/5 & 31/5 \end{array}\right)$$

Portanto

$$Y_1 = B_1 \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$Y_2 = B_2 - L_1 Y_1 = \begin{pmatrix} 1 \\ -11 \end{pmatrix}$$

$$Y_3 = B_3 - L_2 Y_2 = \begin{pmatrix} 107/5 \\ -113/5 \end{pmatrix}$$

Observação: A seguir, num caso geral, se deveria resolver por Gauss, isto é, decompor U_i em LU. No nosso caso não há necessidade.

Resolvendo: $U_3 X_3 = Y_3$ obtemos $x_{31} = 1$, $x_{32} = -1$

$U_2 X_2 = Y_2 - F_2 X_3$ obtemos $x_{21} = 2$, $x_{22} = 0$

$U_1 X_1 = Y_1 - F_1 X_2$ obtemos $x_{11} = 1$, $x_{12} = -1$

e então da solução do sistema é

$$X = \left(\begin{array}{c} x_1 \\ \hline x_2 \\ \hline x_3 \end{array}\right) = \left(\begin{array}{c} 1 \\ -1 \\ \hline 2 \\ 0 \\ \hline 1 \\ -1 \end{array}\right)$$

REDUÇÃO CÍCLICA DE BLOCO

Vamos agora descrever o método de redução cíclica de blocos que pode ser usado para resolver alguns casos especiais de importantes sistemas tridiagonais de blocos.

Para simplificar, vamos assumir que no sistema linear $Ax = b$, a matriz A tenha ordem nq x nq e seja da forma:

$$A = \begin{pmatrix} D & F & & & & \\ F & D & F & & & \\ & F & D & F & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & F \\ & & & & F & D \end{pmatrix}$$

onde os blocos D e F são matrizes de ordem q x q que satisfazem

DF = FD. Vamos também assumir que $n = 2^k - 1$. Esta condição é verificada em algumas aplicações importantes, tais como a discretização da equação de Poisson sobre um retângulo. Neste caso, temos:

$$D = \begin{pmatrix} 4 & -1 & & & & \\ -1 & 4 & -1 & & & \\ & -1 & 4 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & -1 & \\ & & & -1 & 4 & \end{pmatrix} \qquad e \qquad F = -I_q ,$$

com o inteiro n determinado pelo tamanho da malha. Assim, é sempre possível escolhê-lo de modo que $n = 2^k - 1$. Em Sweet (1977), é mostrado como proceder quando n não pode ser escrito nesta forma.

A idéia básica da redução cíclica é repetidamente dividir à metade a dimensão do problema inicial, até que fiquemos com um só sistema q x q cuja incógnita é $x_{2^{k-1}}$. Este é então resolvido e os x_i anteriormente eliminados são encontrados por substituição para trás.

Para facilitar, consideremos o caso em que n = 7. Teremos:

$$\begin{aligned}
b_1 &= Dx_1 + Fx_2 \\
b_2 &= Fx_1 + Dx_2 + Fx_3 \\
b_3 &= \quad Fx_2 + Dx_3 + Fx_4 \\
b_4 &= \quad\quad Fx_3 + Dx_4 + Fx_5 \\
b_5 &= \quad\quad\quad Fx_4 + Dx_5 + Fx_6 \\
b_6 &= \quad\quad\quad\quad Fx_5 + Dx_6 + Fx_7 \\
b_7 &= \quad\quad\quad\quad\quad Fx_6 + Dx_7
\end{aligned} \qquad (4.4.3)$$

Para i = 2, 4 e 6 multiplicamos as equações i - 1, i e i + 1 por F, -D e F, respectivamente e, adicionamos as equações resultantes.

Assim, obtemos o sistema:

$$b_2^{(1)} = D^{(1)}x_2 + F^{(1)}x_4$$
$$b_4^{(1)} = F^{(1)}x_2 + D^{(1)}x_4 + F^{(1)}x_6 \qquad (4.4.4)$$
$$b_6^{(1)} = F^{(1)}x_4 + D^{(1)}x_6$$

onde: $D^{(1)} = 2F^2 - D^2$ e $F^{(1)} = F^2$ comutam;

e, $b_2^{(1)} = F(b_1 + b_3) - D\,b_2$, $b_4^{(1)} = F(b_3 + b_5) - Db_4$

$b_6^{(1)} = F(b_5 + b_7) - Db_6$.

Assim, com esta tática removemos as componentes x_i com i ímpares.

Repetindo a mesma técnica de redução, multiplicamos as três equações de (4.4.4) respectivamente por $F^{(1)}$, $-D^{(1)}$ e $F^{(1)}$ e adicionamos as equações resultantes.

Agora, chegamos à equação:

$$(2[F^{(1)}]^2 - [D^{(1)}]^2)\,x_4 = F^{(1)}(b_2^{(1)} + b_6^{(1)}) - D^{(1)}b_4^{(1)}$$

que escrevemos na forma:

$$D^{(2)}x_4 = b^{(2)} \qquad (4.4.5)$$

Assim, foi completada a redução cíclica. Agora, resolvemos o sistema (4.4.5) de ordem q x q , determinando o vetor x_4. Os vetores x_2 e x_6 são encontrados resolvendo-se os sistemas:

$$D^{(1)}x_2 = b_2^{(1)} - F^{(1)}x_4$$

e

$$D^{(1)}x_6 = b_6^{(1)} - F^{(1)}x_4$$

Finalmente, usamos a $1^{\underline{a}}$, a $3^{\underline{a}}$, a $5^{\underline{a}}$ e a $7^{\underline{a}}$ equações de (4.4.3) para calcularmos x_1 , x_3 , x_5 e x_7 , respectivamente.

Observamos que a quantidade de operações exigidas pelo método depende grandemente da esparcidade de D e F. No pior caso, quando estas matrizes forem densas, o número total de operações será da ordem de log (n) q^3. Buneman (1969) detalha os cuidados que se deve tomar

para se assegurar a estabilidade durante a redução.

O algoritmo de redução cíclica de blocos é o seguinte:

Dado $n = 2^{k-1}$; dados os blocos D e F e o vetor b de dimensão n fazemos:

$D^{(o)} = D$

$F^{(o)} = F$

$b^{(o)} = b$

Para $p = 1, 2, \ldots, k-1$

$D^{(p)} = 2\,[F^{(p-1)}]^2 - [D^{(p-1)}]^2$

$F^{(p)} = [F^{(p-1)}]^2$

$r = 2^p$

Para $j = 1, 2, \ldots, 2^{k-p} - 1$

$$b_{jr}^{(p)} = F^{(p-1)} \left(b_{jr-r/2}^{(p-1)} + b_{jr+r/2}^{(p-1)} \right) - D^{(p-1)}\, b_{jr}^{(p-1)}$$

Resolva $D^{(k-1)}\, x_{2^{k-1}} = b_1^{(k-1)}$ para obter $x_{2^{k-1}}$.

Para $p = k-2,\ k-3,\ \ldots, o$

$r = 2^p$

Para $j = 1, 2, \ldots, 2^{k-p-1}$

Se $j = 1$

$C = b_{(2j-1)r}^{(p)} - F^{(p)}\, x_{2jr}$

Caso contrário, se $j = 2^{k-p+1}$

$C = b_{(2j-1)r}^{(p)} - F^{(p)}\, x_{(2j-2)r}$

Caso contrário,

$C = b_{(2j-1)r}^{(p)} - F^{(p)} \left(x_{2jr} + x_{(2j-2)r} \right)$

Resolva $D^{(p)} x_{(2j-1)r} = C$ para obter $x_{(2j-1)r}$.

Exemplo: Suponhamos que temos $q = 1$, $D = [4]$ e $F = [-1]$ e desejamos resolver

$$\begin{bmatrix} 4 & -1 & & & & & & \\ -1 & 4 & -1 & & & & & \\ & -1 & 4 & -1 & & & & \\ & & -1 & 4 & -1 & & & \\ & & & -1 & 4 & -1 & & \\ & & & & -1 & 4 & -1 & \\ & & & & & -1 & 4 & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \\ 10 \\ 12 \\ 22 \end{bmatrix}$$

Temos

$n = 2^k - 1 = 7 \rightarrow k = 3$

$p = 1, 2.$

Para $p = 1$ temos

$D^{(1)} = 2F^2 - D^2 = [-14]$

$F^{(1)} = F^2 = [1]$

$r = 2^p = 2$

Para $j = 1, 2, 3$ temos

$b_2^{(1)} = F(b_1 + b_3) - Db_2 = [-24]$

$b_4^{(1)} = F(b_3 + b_5) - Db_4 = [-48]$

$b_6^{(1)} = F(b_5 + b_7) - Db_6 = [-80]$

Então $b^{(2)} = F^{(1)}(b_2^{(1)} + b_3^{(1)}) - D^{(1)}b_4^{(1)} = [-776]$

Assim, chegamos a

$$\begin{bmatrix} -14 & 1 & \\ 1 & -14 & 1 \\ & 1 & -14 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ x_6 \end{bmatrix} = \begin{bmatrix} -24 \\ -48 \\ -80 \end{bmatrix}$$

e $[-194]x_4 = [-776] \Rightarrow x_4 = 4.$

De $\begin{cases} [-14]\,[x_2] = [-24] - [-1]\,[x_4] \\ [-14]\,[x_2 = [-24] - [-1]\,[x_4] \end{cases}$

obtemos $x_2 = -2$ e $x_6 = 6$.

E então

$x_1 = 1, \quad x_3 = 3, \quad x_5 = 5, \quad x_7 = 7.$

4.5. PARTIÇÃO PAQ

Seja A uma matriz não singular, não simétrica, grande e esparsa.

Na resolução do sistema linear Ax =b usando a eliminação de Gauss é vantajoso permutarmos equações e variáveis de modo que se resolva:

$$(P\ A\ Q)y = Pb$$

$$x = Qy,$$

onde P e Q são matrizes de permutação.

Estas permutações podem ser escolhidas de modo que minimizem o custo da realização da eliminação de Gauss. Este custo pode ser medido pelo número de posições de memória necessárias e/ou pelo número de operações aritméticas usadas.

A escolha de P e Q pode ser vista como um processo de dois passos.

O primeiro passo é a partição da matriz A. Ele é realizado pela obtenção de P_o e Q_o tais que $P_o A\ Q_o$ seja uma matriz bloco triangular superior. Isto é:

$$P_o A Q_o = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ 0 & A_{22} & \cdots & A_{2k} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & A_{kk} \end{pmatrix} \qquad (4.5.1)$$

onde cada A_{ii} é uma matriz não singular que não pode mais ser particionada.

O segundo passo consiste em encontrar $\bar{P}_i$ e $\bar{Q}_i$, $1 \leq i \leq k$ que permutem as equações e variáveis do i-ésimo bloco de modo à minimizar o custo da aplicação da eliminação de Gauss sobre $\bar{P}_i A_{ii} \bar{Q}_i$.

Então:

$$P A Q = P_K \ldots P_1 P_o A Q_o Q_1 \ldots Q_K = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} & \cdots & \bar{A}_{1k} \\ 0 & \bar{A}_{22} & \cdots & \bar{A}_{2k} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \bar{A}_{kk} \end{pmatrix}$$

onde P_i e Q_i são $\bar{P}_i$ e $\bar{Q}_i$ extendidas ao tamanho de A.

A partição da matriz pode ser realizada da seguinte forma:

1- Determinar uma matriz de permutação R de modo que a diagonal principal de RA seja não nula.

2- Formar o grafo orientado G(RA) cuja matriz de adjacência é RA.

3- Determinar as componentes fortes $C_1, C_2, \ldots, C_k$ de G(RA) e ordená-las de modo que para $i < j$ não exista caminho de C_j para C_i. Tomar a permutação Q_o^t que corresponda a esta ordenação de vértices.

4- Agora, $Q_o^t RAQ_o = P_o A Q_o$ é uma matriz de blocos triangular superior com blocos diagonais $A_{11}, A_{22}, \ldots, A_{kk}$ correspondentes às componentes $C_1, \ldots, C_k$ de G(RA).

Observamos que o que se pretende ao se usar partições é que a resolução de um grande sistema linear possa ser trocada pela resolução

de vários sistemas menores. Assim, suponhamos que tenhamos encontrado P_o e Q_o de modo que P_oAQ_o esteja na forma (4.5.1). Tomemos y e P_ob particionadas respectivamente em $y_1, \ldots, y_k$ e $(P_ob)_1, \ldots, (P_ob)_k$ segundo a partição de P_oAQ_o na equação (4.5.1). Então o sistema:

$$\begin{cases} (P_oAQ_o)\, y = P_ob \\ x = Qy_o. \end{cases}$$

pode ser resolvido por substituição de blocos de baixo para cima:

$$A_{kk}\, y_k = (P_ob)_k$$
$$\vdots$$
$$A_{ii}\, y_i = (P_ob)_i - \sum_{j=i+1}^{k} A_{ij}\, y_j$$
$$\vdots$$
$$A_{11} y_1 = (P_ob)_1 - \sum_{j=2}^{k} A_{1j}\, y_j$$
$$x = Qy_o .$$

Agora, precisamos apenas aplicar a eliminação de Gauss aos blocos diagonais A_{ii}, $1 \leq i \leq k$.

O custo deste processo, certamente não é maior que o que teríamos se aplicássemos a eliminação de Gauss sobre toda a matriz e pode ser menor se a partição for conveniente.

Agora, tomemos $R = QP$. Então $PAQ = Q^tRAQ$.

As matrizes de permutação R e Q podem ser interpretadas da seguinte forma: R permuta as linhas de A de modo que os elementos que serão usados como pivôs na eliminação de Gauss pertencem à diagonal principal de RA. Então a permutação simétrica $Q^t(RA)Q$ determina a ordem em que estes pivôs serão usados.

Assim que R e Q tenham sido escolhidas, a eliminação de Gauss é aplicada aos blocos diagonais da matriz Q^tRAQ sem nenhum pivotamento. Por isso, quase sempre consideramos somente aquelas permutações R para

as quais a diagonal principal de RA (e também de Q^tRAQ) seja inteiramente não nula.

O segundo passo na escolha de P e Q é em geral, mais difícil. Ele exige que encontremos $\bar{P}_i$ e $\bar{Q}_i$ que minimizem o custo da eliminação de Gauss sobre os blocos diagonais A_{ii}. Até agora não se conhece um algoritmo que determine as permutações ótimas sem que se tenha que testar todas as permutações possíveis.

A seguir, apresentamos algumas definições e considerações necessárias ao desenvolvimento do método. Observamos que todos os grafos aqui utilizados são orientados.

Definição (4.5.2): Uma diagonal da matriz quadrada A de ordem n, é qualquer conjunto de n elementos contendo um elemento de cada linha e de cada coluna de A.

Como A é não singular, seu determinante é não nulo e, portanto, pelo menos uma diagonal de A tem todos os seus elementos não nulos.

Definição (4.5.3): Seja Π uma diagonal não nula de A. Ao grafo orientado associado à A e à Π denotaremos $G(A,\Pi)$. Ele é definido como segue:

Tomemos R e S matrizes de permutações tais que os elementos de Π formem a diagonal principal de RAS.

Então $G(A,\Pi) = (V,E)$ onde

$V = \{ v_i \mid 1 \leq i \leq n \}$ e $E = \{ (v_i, v_j) \mid i \neq j \text{ e } (RAS)_{ij} \neq 0 \}$

Quando Π_o é a diagonal principal de A, $G(A,\Pi_o)$ pode também ser escrito simplesmente como $G(A)$.

Mostraremos a seguir que $G(A,\Pi)$ é bem definido, isto é, não se altera pela escolha de R e S.

Teorema (4.5.4): Sejam $G_1(A,\Pi)$ e $G_2(A,\Pi)$ grafos associados aos pares de permutações (R_1,S_1) e (R_2,S_2), respectivamente. Então,

$G_1(A,\Pi)$ e $G_2(A,\Pi)$ são isomorfos.

Demonstração: Seja Π uma matriz de permutação de dimensão n x n cujos 1's estão nas posições correspondentes às locações dos elementos de Π em A.

Como (R_1,S_1) e (R_2,S_2) deslocam Π para a diagonal principal, então $R_1\Pi S_1 = R_2\Pi S_2 = I$. Portanto, $S_1 = \Pi^t R_1^t$ e $S_2 = \Pi^t R_2^t$.

Seja $C = R_2 R_1^t$. Então: $C(R_1 A S_1) C^t = R_2 R_1^t R_1 A S_1 R_1 R_2^t$

$$= R_2 A \Pi^t R_2^t$$

$$= R_2 A S_2$$

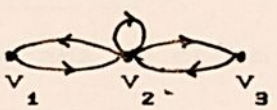
As matrizes R_1AS_1 e R_2AS_2 são as mesmas até uma renumeração das linhas e das colunas pela permutação de C. Portanto: $G_1(A,\Pi) \cong G_2(A,\Pi)$ com o isomorfismo dado por C.

Uma conseqüência deste teorema é que, sem perda de generalidade, podemos sempre mover os elementos pivôs para a diagonal principal usando somente permutações de linhas, isto é, usando $R = \Pi^t$ e $S = I$.

Definição (4.5.6): Um grafo orientado é <u>fortemente conexo</u> se a partir de cada vértice existir um caminho para cada um dos outros vértices.

Por exemplo, a matriz:

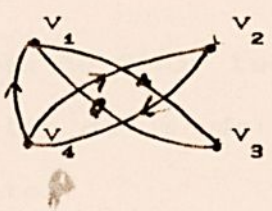
$$A_1 = \begin{pmatrix} . & 2 & . \\ -1 & 3 & 1 \\ . & 1 & . \end{pmatrix}$$ tem o grafo



e $G(A_1)$ é fortemente conexo.

Já a matriz:

$$A_2 = \begin{pmatrix} . & . & 1 & . \\ . & . & 1 & 1 \\ 1 & . & . & . \\ 1 & 1 & 1 & . \end{pmatrix}$$ tem o grafo



e $G(A_2)$ não é fortemente conexo pois não existe um caminho orientado de v_3 à v_2.

Um grafo orientado que não é fortemente conexo pode ser dividido, de maneira única, em subgrafos maximais fortemente conexos que chamaremos componentes fortes.

Estas componentes fortes $C_1, \ldots, C_k$ podem ser ordenadas de modo que para $i < j$ não exista caminho dos vértices de C_j para os vértices de C_i.

Definição (4.5.6): Se $PAQ = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$

onde A_{11} e A_{22} são matrizes quadradas de ordem pelo menos 1 e P e Q são matrizes de permutação, então A é bi-redutível. Caso contrário, A é totalmente indecomponível ou irredutível. Se

$$PAQ = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ 0 & A_{22} & \cdots & A_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{KK} \end{bmatrix}$$

com cada A_{ii}, $i = 1, 2, \ldots, k$ totalmente indecomponível, então as submatrizes A_{ii} são as componentes indecomponíveis de A.

Definição (4.5.7): Sejam A uma matriz quadrada de ordem n x n sem zeros na diagonal principal e G(A) seu grafo. Dizemos que A (ou G(A)) tem a propriedade C se são satisfeitas as seguintes condições:

Sobre a matriz: Cada subconjunto de $k < n$ linhas de A tem elementos não nulas em pelo menos $k + 1$ colunas .

Sobre o grafo: Cada conjunto de $k < n$ vértices de G(A) tem pelo menos uma aresta direcionada (ou orientada) a partir de um vértice de um subconjunto para um vértice de outro subconjunto.

unesp

A seguir apresentaremos o teorema principal sobre o qual se baseia este método.

Teorema (4.5.8): Sejam A uma matriz $n \times n$ sem zeros na diagonal principal e G(A) seu grafo. As afirmações abaixo são equivalentes:

(1) A tem a propriedade C.

(2) A é totalmente indecomponível.

(3) G (A) é fortemente conexo.

Demonstração: Mostraremos que não (1) => não (2):

Seja S um subconjunto de $k < n$ linhas de A sem a propriedade C. Em S existem k elementos da diagonal principal de A. Estes elementos não nulos estão em k colunas diferentes. Como a propriedade C não ocorre, as outras colunas de S devem ser nulas. Seja P uma matriz de permutação que coloca as linhas de S em último lugar.

Então:

$$PAP^t = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \quad \text{linhas em S,}$$

e A é bi-redutível.

Mostremos que não (2) => não (1):

Seja $PAQ = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$

Então a propriedade de C não é satisfeita nas linhas de A_{22}; e, portanto isto também ocorre em A.

Mostremos que não (1) => não (3):

Seja S um subconjunto de linhas de A onde a propriedade C não ocorre. Então, não existe caminho em G(A) a partir de vértices correspondentes às linhas em S para vértices fora deste conjunto.

Logo, G(A) não é fortemente conexo.

Mostremos que não (3) => não (1):

Se G(A) não é fortemente conexo ele deve ter uma componente forte C_k sem arestas externas. Então as linhas de A correspondentes a vértices em C_k tem elementos não nulos somente nas colunas correspondentes a vértices em C_k, o que viola a propriedade C.

A seguir, apresentamos um Corolário e um Teorema cuja demonstração pode ser encontrada em Howell, 1982.

Corolário (4.5.9): Sejam R e S matrizes de permutação tais que RA e SA tenham diagonais principais não nulas. Então G(RA) é fortemente conexo se, e somente se, G(SA) é fortemente conexo.

Teorema (4.5.10): Cada diagonal não nula Π de A leva à mesma partição de vértices em componentes fortes em G (A, Π).

Seja A uma matriz esparsa quadrada de ordem n x n, não singular com τ elementos não nulos ($n \leq \tau \leq n^2$). Agora, o algoritmo para a partição pode ser reescrito como segue:

1 - Encontrar uma diagonal não nula Π e a permutação R que a desloca para a diagonal principal.

2 - Formar G(A, Π).

3 - Encontar as componentes fortes de G(A, Π) numa determinada ordem tal que para $i < j$ não exista caminho de C_j para C_i. Tomar a permutação Q^t de modo que a linha k preceda a linha p enquanto $v_k \in C_i$, $v_p \in C_j$ e $i < j$.

4 - Formar $PAQ = Q^tRAQ$ a qual é matriz de bloco triangular superior.

Exemplo: Seja o sistema linear $Ax = b$ tal que:

$$A = \begin{pmatrix} \boxed{2} & . & 4 & . & . \\ 1 & 1 & . & . & \boxed{3} \\ . & . & 3 & \boxed{4} & . \\ 2 & \boxed{2} & . & 1 & 5 \\ . & . & \boxed{1} & 1 & . \end{pmatrix} \qquad b = \begin{pmatrix} . \\ 2 \\ 2 \\ 2 \\ 1 \end{pmatrix}$$

1 - Escolhamos qualquer diagonal não nula. Por exemplo, seja:

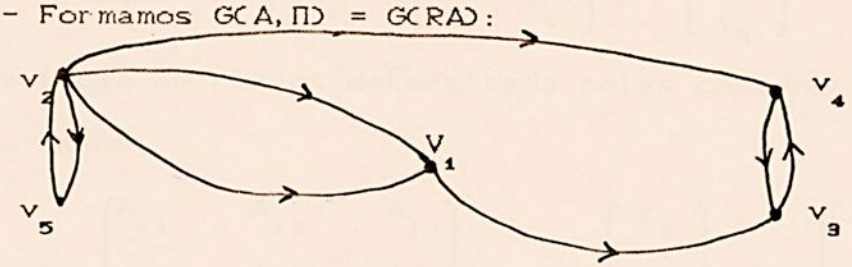
$\Pi = (a_{11}, \ a_{25}, \ a_{34}, \ a_{42}, \ a_{53})$. Então

$$R = \begin{pmatrix} 1 & . & . & . & . \\ . & . & . & 1 & . \\ . & . & . & . & 1 \\ . & . & 1 & . & . \\ . & 1 & . & . & . \end{pmatrix} \qquad \text{ou} \qquad R = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 5 & 4 & 2 & 3 \end{pmatrix}$$

Como $RAx = Rb$, temos:

$$RA = \begin{pmatrix} 2 & . & 4 & . & . \\ 2 & 2 & . & 1 & 5 \\ . & . & 1 & 1 & . \\ . & . & 3 & 4 & . \\ 1 & 1 & . & . & 3 \end{pmatrix} \qquad e \qquad Rb = \begin{pmatrix} . \\ 2 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

2 - Formamos $G(A, \Pi) = G(RA)$:



3 - Ordenamos as componentes fortes; e tomamos a permutação Q^t que reflita esta ordem. Teremos:

$C_1 = (v_2, v_5)$; $C_2 = (v_1)$ e $C_3 = (v_3, v_4)$;

e $Q^t = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 4 & 5 & 2 \end{bmatrix}$. Logo, $Q = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 1 & 3 & 4 \end{bmatrix}$

4 - Determinamos $PAQ = Q^tRAQ$. Teremos:

$$Q^tRA = \begin{bmatrix} 2 & 2 & . & 1 & 5 \\ 1 & 1 & . & . & 3 \\ 2 & . & 4 & . & . \\ . & . & 1 & 1 & . \\ . & . & 3 & 4 & . \end{bmatrix} \quad \text{e}$$

$$Q^tRAQ = \left[\begin{array}{cc|c|cc} 2 & 5 & 2 & . & 1 \\ 1 & 3 & 1 & . & . \\ \hline . & . & 2 & 4 & . \\ \hline . & . & . & 1 & 1 \\ . & . & . & 3 & 4 \end{array}\right] = PAQ$$

Então, como $PAQy = Pb$ onde $Pb = Q^tRb$, temos:

$$\left[\begin{array}{cc|c|cc} 2 & 5 & 2 & . & 1 \\ 1 & 3 & 1 & . & . \\ \hline . & . & 2 & 4 & . \\ \hline . & . & . & 1 & 1 \\ . & . & . & 3 & 4 \end{array}\right] \left[\begin{array}{c} y_1 \\ y_2 \\ \hline y_3 \\ \hline y_4 \\ y_5 \end{array}\right] = \left[\begin{array}{c} 2 \\ 2 \\ \hline . \\ \hline 1 \\ 2 \end{array}\right]$$

com a partição em blocos determinada pelas componentes obtidas em 3.

Ou seja, temos:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ & A_{22} & A_{23} \\ & & A_{33} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} (Pb)_1 \\ (Pb)_2 \\ (Pb)_3 \end{bmatrix}$$

Resolvendo os blocos, concluímos que:

$$y = (3 \quad 1 \quad -4 \quad 2 \quad -1)^t$$

Como $x = Qy$, chegamos à solução $x = (-4 \quad 3 \quad 2 \quad -1 \quad 1)^t$

do sistema.

Observação 1 : Se neste exemplo tivessemos escolhido para Π a diagonal principal de A, não teríamos a permutação R que moveria Π para a diagonal principal. O grafo resultante seria fortemente conexo e nenhuma partição poderia ser feita.

A seguir apresentaremos um exemplo que mostra como a escolha da diagonal não nula pode afetar o custo da eliminação de Gauss sobre uma componente de A.

Suponhamos que:

$$A_{ii} = \begin{pmatrix} X & X & . & X \\ . & X & X & . \\ . & X & X & X \\ X & . & . & X \end{pmatrix}$$

onde x indica um elemento cujo valor é não nulo.

A escolha de $\bar{R}_i = I$ conduz a pelo menos um fill-in independentemente da escolha de Q_i.

Já a escolha de $\bar{R}_i = \begin{pmatrix} . & . & . & 1 \\ . & 1 & . & . \\ . & . & 1 & . \\ 1 & . & . & . \end{pmatrix} \leftrightarrow \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix}$

não conduz a nenhum fill-in se Q_i for escolhido como sendo I, por exemplo.

Observação 2 : Suponhamos que A tenha um zero em sua diagonal principal o qual seria trocado por um elementos não nulo na eliminação de Gauss antes que fosse usado como um divisor. Ainda assim podemos construir G(A) e fazer a partição como antes. Pelo Teorema (4.5.10) este grafo teria a mesma partição de vértices em componentes fortes

que aquela gerada , como segue:

1 - Marque o zero sobre a diagonal principal.

2 - Encontre R tal que RA tenha diagonal principal não nula e construa G(RA) como antes.

3 - Adicione uma aresta correspondente ao zero marcado em 1 o qual está agora fora da diagonal principal de RA. Esta aresta adicionada pode ou não mudar a estrutura de componentes fortes de G(RA). Por ela é possivel fundir duas ou mais componentes de G(RA) em uma única componente.

Um algoritmo eficiente para determinarmos as componentes fortes de um grafo fortemente conexo, numa ordem conveniente é dado por Tarjan, 1972; Szwarafites, 1986; Terada, 1991.

CAPÍTULO 5

MÉTODOS ITERATIVOS PARA RESOLUÇÃO DE SISTEMAS LINEARES ESPARSOS

5.1. INTRODUÇÃO

Os métodos iterativos para a resolução de sistemas de equações algébricas foram revistos em 1977 por Fox. Desde então, um avanço notável nesta área tem sido verificado.

Uma razão para o grande interesse na resolução de sistemas de equações lineares esparsos é sua importância na discretização de equações diferenciais. Pesquisas computacionais nesta área têm sido responsáveis por algumas técnicas para matrizes esparsas que são de uso geral.

Falando de modo mais amplo, existem duas maneiras para solucionar o problema Ax = b com A esparsa. Uma delas é como já vimos a dos métodos diretos, que com estratégias típicas de exploração envolvem um inteligente uso da estrutura de dados e pivotamento estratégico especial, para minimizar o fill-in. Uma outra é a dos métodos iterativos que fornecem uma seqüência de aproximantes da solução, cada um dos quais obtido dos anteriores pela repetição do mesmo tipo de processo.

Um método iterativo é dito estacionário se cada aproximante é obtido do anterior pelo mesmo processo. Quando os processos variam de passo para passo mas se repetem ciclicamente de s em s passos, ele é dito s-cíclico. Agrupando-se os s passos de cada ciclo num único passo composto, obtemos um método estacionário.

Analisemos os métodos iterativos estacionários.



Suponhamos que o sistema linear $Ax = b$ tenha sido transformado no sistema equivalente

$$x = Mx + g.$$

Um método iterativo básico de resolução do sistema linear é escrito na forma

$$x^{(k+1)} = Mx^{(k)} + g \quad , \qquad k = 0, 1, 2, \ldots \qquad (5.1.1)$$

onde a matriz M é chamada matriz de iteração.

Observamos que se a seqüência $\{ x^{(k)} \}$ converge para $\bar{x}$ então $\bar{x}$ coincide com a solução x de $Ax = b$.

O critério geral de convergência de (5.1.1) é dado por:

Teorema (5.1.2): Se $\| \cdot \|$ é uma norma qualquer de matriz, a condição $\| M \| < 1$ é suficiente para a convergência do processo iterativo (5.1.1).

Temos também

Teorema (5.1.3): Uma condição necessária e suficiente para a convergência do processo iterativo (5.1.1) é $\rho(M) < 1$ onde $\rho(M) = \max_{1 \le i \le n} |\lambda_i|$ é o raio espectral de M.

A seguir apresentamos alguns dos métodos iterativos mais usados.

5.2. MÉTODOS ITERATIVOS CLÁSSICOS

Seja o sistema $Ax = b$. Definindo as matrizes D, E, F por

$$d_{ij} = \begin{cases} a_{ij} , & i = j \\ 0 , & i \neq j \end{cases} \quad ; \quad e_{ij} = \begin{cases} -a_{ij} , & i > j \\ 0 , & i \leq j \end{cases} \quad ; \quad f_{ij} = \begin{cases} -a_{ij} , & i < j \\ 0 , & i \geq j \end{cases}$$

podemos escrever $A = D - E - F$.

Supondo que $\det(D) \neq 0$, consideremos $L = D^{-1}E$ e $U = D^{-1}F$.

1) MÉTODO DE JACOBI-RICHARDSON OU DOS DESLOCAMENTOS SIMULTÂNEOS.

De $(D - E - F)x = b$ obtemos $Dx = (E + F)x + b$. O método iterativo resultante

$$x^{(k+1)} = (L + U)\, x^{(k)} + D^{-1}b$$

é denominado método de Jacobi-Richardson.

Assim, neste método a matriz de iteração é $M_J = L + U = D^{-1}(E+F)$.

O algoritmo do método de Jacobi é dado por:

Dada uma aproximação inicial $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ arbitrária, calcule:

Para $k = 0, 1, 2, \ldots$:

  Para $i = 1, 2, \ldots, n$:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}\, x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}\, x_j^{(k)} \right) / a_{ii}$$

2) MÉTODO DE GAUSS-SEIDEL OU DOS DESLOCAMENTOS SUCESSIVOS.

De $(D - E - F)x = b$ obtemos $(D - E)x = Fx + b$. O método iterativo resultante

$$x^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + D^{-1}b$$

ou

$$x^{(k+1)} = (I - L)^{-1}Ux^{(k)} + (D - E)^{-1}b$$

é denominado método de Gauss-Seidel.

Portanto, neste método a matriz de iteração é $M_{Gs} = (I - L)^{-1}U$.

Observamos que este método difere do de Jacobi por utilizar para o cálculo de uma componente de $x^{(k+1)}$ o valor mais recente das demais componentes. Computacionalmente isto é bastante conveniente, pois o novo valor pode ser armazenado na memória sobre o valor antigo, o que minimiza a memória utilizada.

É importante observar também que as componentes $x_i^{(k+1)}$ são

calculadas sem necessidade de se calcular $(I - L)^{-1}U$. Assim, o algoritmo do método de Gauss-Seidel é dado por:

Dada uma aproximação inicial $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ arbitrária, calcule:

Para $k = 0, 1, 2, \ldots$:

  Para $i = 1, 2, \ldots, n$:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) / a_{ii}$$

3) MÉTODO DE SOBRE-RELAXAÇÃO SUCESSIVA (SOR).

O problema $f(x)=0$ pode ser resolvido por iteração na forma $x = x + \omega f(x)$, onde o parâmetro ω é escolhido de modo que a iteração convirja. Analogamente, o sistema $b - Ax = 0$ pode ser reduzido à forma $x = x + \omega D^{-1}(b - Ax)$, com $\omega \in \mathbb{R}$ apropriadamente escolhido. Este procedimento corresponde à decomposição

$$A = \frac{1}{\omega}(D - \omega E) - \frac{1}{\omega}(\omega F + (1 - \omega) D)$$

que nos fornece o método iterativo de sobre-relaxação (SOR) definido por:

$$x^{(k+1)} = (1 - \omega)x^{(k)} + \omega (Lx^{(k+1)} + Ux^{(k)} + D^{-1}b)$$

ou

$$x^{(k+1)} = x^{(k)} + \omega D^{-1} \left(b - Dx^{(k)} + Ex^{(k+1)} + Fx^{(k)} \right).$$

Como $\det(I - \omega L)^{-1} = 1$ segue que

$$x^{(k+1)} = M_{SOR} x^{(k)} + \omega (D - \omega E)^{-1} b$$

onde $M_{SOR} = (I - \omega L)^{-1} \left[\omega U + (1-\omega) I \right]$.

Este método converge significativamente mais rápido que os anteriores, mas a convergência depende do fator de relaxação ω. O parâmetro ω é escolhido de modo a minimizar $\rho(M_{SOR})$ para que $\{x^{(k)}\}$ convirja para x o mais rápido possível. Este valor de ω é chamado

valor ótimo e é denotado ω^*. O cálculo de ω^* é difícil exceto nos casos mais simples. Usualmente ele é obtido aproximadamente, baseado em experimentos com valores de ω e em seus efeitos na velocidade de convergência. Apesar de não ser simples calculá-lo, o efeito de ω^* na velocidade de convergência é significativo e compensa seu custo de cálculo.

Foi demonstrado (ver, por exemplo, Young, 1971) que $0 < \omega < 2$ é condição necessária para a convergência. Observamos que se $\omega = 1$ o método SOR se reduz ao método de Gauss-Seidel. Se $\omega > 1$, então temos uma correção por excesso pois $x_i^{(k+1)} = x_i^{(k)} + \omega (x_i^{(k+1)} - x_i^{(k)})$ onde $x_i^{(k+1)}$ é o valor calculado pelo método de Gauss-Seidel. Por isso, neste caso, o método é denominado sobre-relaxação. Similarmente, se $\omega < 1$ teremos o método denominado sub-relaxação.

4) MÉTODO SIMULTÂNEO OVER-RELAXAÇÃO (JOR)

Este método está relacionado com o método de Jacobi. Nele escolheremos um parâmetro real ω e fazemos:

$$x^{(k+1)} = M_{JOR} x^{(k)} + \omega D^{-1} b$$

onde $M_{JOR} = \omega M_J + (1 - \omega) I$.

Se $\omega = 1$ teremos o método de Jacobi.

Se $\omega > 1$ teremos uma correção por excesso (sobre-relaxação) desde que $x^{(k+1)} = x^{(k)} + \omega (x^{(k+1)} - x^{(k)})$ onde $x_i^{(k+1)}$ é obtida pelo método de Jacobi. Similarmente, se $\omega < 1$ temos correção por falta (sub-relaxação). Posteriormente faremos algumas considerações sobre ω.

5) MÉTODO DE SOBRE-RELAXAÇÃO SUCESSIVA SIMÉTRICA (SSOR)

O sistema $b - Ax = 0$ pode ser resolvido pelo método SSOR em duas etapas (semi-iterações).

unesp

A primeira semi-iteração é idêntica à realizada pelo método SOR (sobre-relaxação sucessiva) para obter $x^{(k+1/2)}$. A segunda semi-iteração para obter $x^{(k+1)}$ também utiliza o método SOR porém, com as equações tomadas na ordem inversa.

Assim, primeiramente determinamos $x^{(k+1/2)}$ da seguinte maneira

$$x^{(k+1/2)} = (1 - \omega)\, x^{(k)} + \omega\,(Lx^{(k+1/2)} + Ux^{(k)} + D^{-1}b$$

ou

$$x^{(k+1/2)} = x^{(k)} + \omega\, D^{-1}(b - Dx^{(k)} + Ex^{(k+1/2)} + Fx^{(k)})$$

e segue que

$$x^{(k+1/2)} = M_{SOR}x^{(k)} + \omega\,(D - \omega E)^{-1}b,$$

onde

$$M_{SOR} = (I - \omega L)^{-1}\,[\omega U + (1 - \omega)\, I].$$

Agora, determinamos $x^{(k+1)}$ fazendo

$$x^{(k+1)} = M^{*}_{SOR}\, x^{(k+1/2)} + (1 - \omega U)^{-1}b$$

onde

$$M^{*}_{SOR} = (I - \omega U)^{-1}\,[\omega L + (1 - \omega)\, I].$$

5.3. MÉTODOS BLOCO-ITERATIVOS

Consideremos a matriz quadrada A de ordem n particionada em blocos na forma

$$A_B = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & & & \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{pmatrix}$$

onde as matrizes A_{ii} são quadradas de ordem m_i, $i = 1, 2, \ldots, p$ e $\sum_{j=1}^{p} m_i = n$.

Se aplicamos os métodos iterativos anteriores ao sistema

$$\sum_{j=1}^{p} A_{ij} x_j = b_i , \qquad i = 1, 2, \ldots, p ,$$

onde b_i e x_i são vetores de dimensão m_i, obteremos novos métodos iterativos que possuem a forma

$$x^{(k+1)} = M^{-1} N x^{(k)} + M^{-1} b.$$

Denominaremos D_B à matriz

$$\begin{bmatrix} A_{11} & & & 0 \\ & A_{22} & & \\ & & \ddots & \\ 0 & & & A_{pp} \end{bmatrix}$$

Seja $A_B = D_P - E_B - F_B$, $L_B = D_B^{-1} E_B$, $U_B = D_B^{-1} F_B$

onde E_B e L_B são matrizes bloco triangular estritamente inferiores; e, F_B e U_B são matrizes bloco triangular estritamente superiores.

Com estas matrizes blocos obteremos vários métodos bloco iterativos que apresentaremos a seguir.

Em cada passo k do processo iterativo é preciso resolver p sistemas lineares por métodos diretos, pois a inversão de D_B não se reduz mais a uma simples divisão como ocorria nos métodos não blocos.

Em geral é aconselhável determinar as inversas A_{ii}^{-1} antes do início da iteração, para facilitar a resolução dos p sistemas em cada passo. Entretanto, se os A_{ii} são tridiagonais é preferível a resolução sucessiva pela eliminação de Gauss. Este caso é frequente no

tratamento de Equações Diferenciais Parciais.

1) O método ITERATIVO BLOCO DE JACOBI é formado pela decomposição

$$A_B = D_B - E_B - F_B,$$

e é definido por

$$x^{(k+1)} = (L_B + U_B)\, x^{(k)} + D_B^{-1} b.$$

A matriz $J_B = L_B + U_B$ é chamada matriz bloco de Jacobi.

2) O método ITERATIVO BLOCO DE GAUSS-SEIDEL é fornecido pela decomposição. $A_B = (D_B - E_B) - F_B$ e é definida por

$$x^{(k+1)} = L_B x^{(k+1)} + U_B x^{(k)} + D^{-1} b$$

ou

$$x^{(k+1)} = (I - L_B)^{-1}\, U_B x^{(k)} + (D_B - E_B)^{-1} b.$$

A matriz $GS_B = (I - L_B)^{-1} U_B$ é chamada matriz bloco de Gauss-Seidel.

3) O método ITERATIVO BLOCO SOBRE-RELAXAÇÃO SUCESSIVA é fornecido pela decomposição

$$A_B = (1 - \omega)\, x^{(k)} + \omega\, (L_B x^{(k+1)} + U_B x^{(k)} + D_B^{-1} b)$$

e é definido por

$$x^{(k+1)} = (I - \omega L_B)^{-1} (\omega U_B + (1 - \omega) I) x^{(k)} + \omega\, (D_B - \omega E_B)^{-1} b.$$

A matriz $SOR_B = (I - \omega L_B)^{-1}\, (\omega U_B + (1 - \omega)\, I)$ é chamada matriz bloco de sobre-relaxação sucessiva.

A iteração de blocos é usada quando o sistema é muito grande para ser armazenado na memória do computador. Em alguns casos, estes métodos convergem mais rapidamente que os métodos iterativos básicos.

unesp

A seguir, apresentamos um exemplo que ilustra a aplicação do método iterativo bloco de Jacobi.

Exemplo: Seja o sistema linear onde

$$A_B = \left[\begin{array}{cc|cc|c} 2 & -1 & . & . & . \\ -1 & 2 & -1 & . & . \\ \hline . & -1 & 2 & -1 & . \\ . & . & -1 & 2 & -1 \\ \hline . & . & . & -1 & 1 \end{array}\right] \quad e \quad b = \left[\begin{array}{c} 3 \\ -5 \\ \hline 9 \\ -6 \\ \hline 2 \end{array}\right]$$

Com esta partição em blocos, o método iterativo bloco de Jacobi resulta em

$$\begin{cases} 2x_1^{(k+1)} - x_2^{(k+1)} = 3 \\ -x_1^{(k+1)} + 2x_2^{(k+1)} = x_3^{(k)} - 5 \end{cases}$$

$$\begin{cases} 2x_3^{(k+1)} - x_4^{(k+1)} = x_2^{(k)} + 9 \\ -x_3^{(k+1)} + 2x_4^{(k+1)} = x_5^{(k)} - 6 \end{cases}$$

$$\begin{cases} x_5^{(k+1)} = x_4^{(k)} + 2 \end{cases}$$

Resolvendo estes sistemas obtemos:

$$x_1^{(k+1)} = \frac{1}{3} x_3^{(k)} + \frac{1}{3}$$

$$x_2^{(k+1)} = \frac{2}{3} x_3^{(k)} - \frac{7}{3}$$

$$x_3^{(k+1)} = \frac{2}{3} x_2^{(k)} + \frac{1}{3} x_5^{(k)} + 4 \; ; \qquad k = 0, 1, \ldots .$$

$$x_4^{(k+1)} = \frac{1}{3} x_2^{(k)} + \frac{2}{3} x_5^{(k)} - 1$$

$$x_5^{(k+1)} = x_4^{(k)} + 2$$

Observamos que estas equações correspondem à iteração

$$x^{(k+1)} = J_B\, x^{(k)} + D_B^{-1} b$$

do método bloco de Jacobi, onde

$$J_B = \left[\begin{array}{cc|cc|cc} . & . & 1/3 & . & . & . \\ . & . & 2/3 & . & . & . \\ \hline . & 2/3 & . & . & . & 1/3 \\ . & 1/3 & . & . & . & 2/3 \\ \hline . & . & . & 1 & . & . \end{array}\right] \quad \text{e} \quad D_B^{-1} = \frac{1}{3}\left[\begin{array}{cc|cc|cc} 2 & 1 & . & . & . & . \\ 1 & 2 & . & . & . & . \\ \hline . & . & 2 & 1 & . & . \\ . & . & 1 & 2 & . & . \\ \hline . & . & . & . & 3 & \end{array}\right]$$

CONVERGÊNCIA DOS MÉTODOS ITERATIVOS

Todos os métodos iterativos descritos anteriormente são da forma

$$x^{(k+1)} = Mx^{(k)} + g \qquad k = 0, 1, \ldots \qquad (5.1.4)$$

A seguir apresentaremos alguns teoremas de convergência para os métodos iterativos básicos.

Se no critério de convergência apresentado no Teorema (5.1.2) escolhermos as normas $\| \cdot \|_\infty$ e $\| \cdot \|_1$ e considerarmos $a^*_{ij} = a_{ij}/a_{ii}$ obtemos:

Corolário (5.1.5): O método de Jacobi-Richardson converge se

$$\max_i \sum_{\substack{j=1 \\ j \neq i}}^{n} | a^*_{ij} | < 1 \qquad \text{(critério das linhas)}$$

ou se

$$\max_j \sum_{\substack{i=1 \\ i \neq j}}^{n} | a^*_{ij} | < 1 \qquad \text{(critério das colunas)}$$

Corolário (5.1.6): O método de Gauss-Seidel converge se o critério das linhas for satisfeito.

Corolário (5.1.7): O método de Gauss-Seidel converge se $\max_i \beta_i < 1$

onde $\beta_i = \sum_{j=1}^{i-1} | a^*_{ij} | \beta_j + \sum_{j=i+1}^{n} | a^*_{ij} |$ (critério Sassenfeld).

Em geral o método de Gauss-Seidel converge assintoticamente mais rápido que o método de Jacobi-Richardson. No entanto, dado um sistema Ax = b pode ocorrer que o método de Jacobi-Richardson aplicado a ele resulte convergente enquanto que o de Gauss-Seidel resulte divergente e vice-versa.

Definição (5.1.8): Sejam M_1 e M_2 matrizes de iteração. Então a iteração (5.1.4) com $M = M_1$ converge assintoticamente mais rápido que a iteração com $M = M_2$, se $\rho(M_1) < \rho(M_2) < 1$.

Teorema (5.1.9): Seja $M_{SOR} = (I - \omega L)^{-1} [\omega U + (1 - \omega) I]$.

Então, para todo real ω, $\rho(M_{SOR}) \geq |\omega - 1|$. (Albrechet, 1973).

Deste teorema e do Teorema (5.1.3) decorre que $0 < \omega < 2$ é uma condição necessária para a convergência do método de sobre-relaxação (SOR).

Teorema (5.1.10): Se a matriz A do sistema é simétrica e positiva definida então o método de Gauss-Seidel é sempre convergente.

Teorema (5.1.11): Se o método de Jacobi converge então o método de Jacobi over-relaxação (JOR) converge se $0 < \omega \leq 1$.

Demonstração: De M_{JOR} temos que os autovalores λ_i do método JOR são dados em função dos autovalores μ_i do método de Jacobi por

$\lambda_i = \omega\mu_i + 1 - \omega$

Se $\mu_i = re^{i\theta}$, $r < 1$ e $o < \omega \leq 1$, então

$| \lambda_i |^2 = \omega^2 r^2 + 2 \omega r \cos\theta (1-\omega) + (1-\omega)^2 \leq (\omega r + 1 - \omega)^2 < 1$

e o método de Jacobi over-relaxação converge.

Definição (5.1.12): Uma matriz A de ordem n x n é <u>diagonalmente dominante</u> se

$$| a_{ii} | \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} | a_{ij} | , \qquad i = 1, 2, \ldots, n$$

e para pelo menos um i,

$$| a_{ii} | > \sum_{\substack{j=1 \\ j \neq i}}^{n} | a_{ij} | . \qquad (5.1.13)$$

Se (5.1.13) se verifica para todo i, dizemos que A é estritamente diagonalmente dominante.

Teorema (5.1.14): Seja A uma matriz irredutível e diagonalmente dominante. Então

(a) O método de Jacobi converge e o método JOR converge para $0 < \omega \leq 1$.

(b) O método de Gauss-Seidel e o método SOR convergem para $0 < \omega \leq 1$ (Young, 1971).

Teorema (5.1.15): Seja A uma matriz simétrica com elementos diagonais positivos. Então o método SOR converge se, e somente se, A é positiva definida e $0 < \omega < 2$ (Young, 1971).

Teorema (5.1.16): Se os métodos de Jacobi e Gauss-Seidel forem aplicados a um sistema tridiagonal então ou ambos convergem ou ambos divergem. Se ambos convergirem, o método de Gauss-Seidel converge mais rapidamente que o de Jacobi.

Observamos que os métodos iterativos são muito utilizados na resolução de grandes sistemas lineares esparsos. Há inúmeros pacotes computacionais tais como ITPACK 2C, ITPACKV 2C, ITPACK 3B, ITPACK 4 ou NSPCG, ORTHOMIN e outros que utilizam estes métodos.

unesp

BIBLIOGRAFIA

(Aho,1974) - Aho,A.V.; J.E.Hopcroft; J.D.Ulman; The design and analysis of computer algoritms. Copyright - 1974 - USA.

(Albretch,1973) - Albrecht,P. Análise Numérica: um curso moderno. Editora USP - Rio de Janeiro - 1973 - Br.

(————,1982) - Métodos iterativos para sistemas lineares e sua aplicação na solução de equações diferenciais parciais PVC - Rio de Janeiro - 1982 - Br.

(Allwood,1971) - Allwood,R.J. Matrix methods of structural analysis. In "Large sparse sets of linear equations". (J.K.Reid, ed.), pp. 17-24. Academic Press. New York.

(Akyuz,1968) - Akyuz,F.A.; Utku,S. An automatic relabeling scheme for bandwidth minimization of stiffness matrices. AIAA J. 6,728-730.

(Baker,1962) - Baker,J.M. - A note on multiplying Boolean matrices. Comm. ACM 5,102.

(Buneman,1969) - Buneman, O. A compact non iterative Poisson solver, Report 294, Stanford University Institute for Plasma Research, Stanford. California, 1969 - USA.

(Clansen,1966) - Clansen,R.J. - Techniques for automatic tolerance control in linear programming. Comm. ACM 9,802.

(Chen,1972) - Chen,Y.T.; Permutation of irreducible sparse matrices to upper triangular form.IMA J. **10**,15-18.

(Comstock,1964) - Comstock,D.R. A note on multiplying boolean matrices II. Comm. ACM 7,13.

(Cunha,1993) - Cunha Cristina - Métodos numéricos para as engenharias e ciências aplicadas. Editora Unicamp. Campinas. Br.

(Cuthill,1969) - Cuthill,E.H.; Mckee,J. Reducing the bandwidth of sparse symmetric matrices. Tech. note AML 40-69. Appl. Math. Lab., Naval Ship Res. and Develop. Center, Washington, D.C.

(Cuthill,1971) - Cuthill,E.H. Several strategies for reducing the bandwidth of matrices. Tech. note CMD 42-71, Naval Ship Res. and Develop. Center, Bethesda, Maryland.

(Dantzig,1963) - Dantzig,G.B. Linear Programming and extensions. Princeton Univ. Press. Princeton, New Jersey.

(Duff,1972) - Duff,I. On a factored form of the inverse for sparse matrices. D.Phil. thesis, Oxford University - USA.

(———,1981) - Sparce matrices and their uses. The exploration of sparsity by iterative methods - DAH Jacobs.

(Dulmage,Mendelsohn,1963) - Dulmage,A.L.; N.S. Mendelsohn. Two algorithms for bipartite graphs. SIAM J. Appl.Math. **11**,183-194.

(Fox,1977) - Fox,L. "Introduction to numerical linear algebra". Oxford University Press, (Clarendon), London and New York.

(Golub,1985) - Golub G.H.; V.Loan. Matrix computations. Baltimore, Johns Hopkins University Press.

(Hageman,1981) - Hageman,L.A.; D.M.Young. Applied iterative Methods - Academic Press, Inc. Orlando. Florida.

(Harary,1962) - Harary,F. A graph theoric approach to matrix inversion by partitioning. Numer. math. **4**, 128-135.

(————,1971) - Sparse matrices and graph theory. In "Large sparse sets of linear equations". (J.K.Reid ed.), pp 139-150. Academic Press. New York.

(Howell,1982) - Howell T.D. Partitioning using PAQ. In "Sparce matrix computations". pp 23-37.

(Ingerman,1962) - Ingerman,P.Z. Path natrix, Algorithm 141. Comm. ACM 5, 556.

(Kettler e Weil,1969) - Kettler,P.C.; R.L.Weil. An algorithm to provide structure for decomposition. In "Sparse matrix proceedings". Rep. No. RA1 (11707), pp 11-24. IBM, Yorktown Heigths, New York.

(Liniger e Willoughby,1969) - Liniger,W; Willoughby,R.A. Efficient numerical integration of stiff systems of differential equations. SIAM J. Numer. anal. 7, 47-66.

(Markowitz,1957) - Markowitz,H.M. - The elimination form of the inverse and its application to linear programming. Management Sci 3,255-269.

(Rabinowitz,1968) - Rabinowitz,P. Applications of linear programming to numerical analysis. SIAM Rev. **10**, 121-159.

(Rose e Bunch,1972) - Rose R.J.; Bunch,J.R. The role of partitioning in the numerical solution of sparse systems. In "Sparse matrices and their applications", pp 177-190. Plenum Press. New York.

(Rosen,1968) - Rosen,R. Matrix bandwidth minimization. Proc. 23rd Nat. Conf. ACM Publ. **P-68** pp 585-595. Brandon Systems Press, Princeton, New Jersey.

(Schwarz,1989) - Schwarz,H.R. Numerical Analysis - A comprehensive introduction. Typeset by Thonson Press (India) Ltd. New Delhi. India.

(Steward,1962) - Steward D.V. On an approach to techniques for the analysis of the structure of large systems of equation. SIAM Rev. **4**,321-342.

(————,1965) - Partitioning and tearing systems of equations. SIAM J. Numer. Anal. 2,345-365.

(————,1969) - Tearing analysis of the structure of disorderly sparse matrices. In "Sparse matrix proceedings". (R.A.Willoughby ed.). Rep. No. RA1 (11707) pp 65-74. IBM. Yorktown Heights. New York.

(Sweet,1977) - Sweet,R.A. A cyclic reduction algorithm for solving block tridiagonal systems of arbitrary dimension. SIAM J. Num. Anal. **14**, 706-720.

(Szwarcfiter,1986) - Szwarcfiter,J.L. Grafos e algoritmos computacionais. Editora Campus Ltda. Rio de Janeiro - RJ.

(Tarjan,1972) - Tarjan,R. Depht-first search and linear graph algorithms. SIAM J. Comput - Vol I - Junho 1972.

(Terada,1991) Terada,R. Desenvolvimento de algoritmo e estruturas de dados. Editora MacGraw-hill Ltda e Makron Books do Brasil Editora Ltda. São Paulo - S.P.

(Tewarson,1966) - Tewarson,R.P. On the product form of inverses of sparse matrices. SIAM Rev **8**,336-342.

(————,1967a) - Row column permutation of sparse matrices. Comput. J. **10**,300-305.

(————,1967b) - Solution of a system of simultaneous linear equations with a sparse coefficient matrix by elimination methods. Nordisk. Tidskr. Informations Behadling (BIT) 7, 226-239.

(————,1969) - The Gaussian elimination and sparse systems. In "Sparse matrix proceedings". (R.A.Weloughby, ed.), Report No. RA1 (11707), pp. 35-42. IBM, Yorktown Heights. New York.

(————,1971) - Sorting and ordering sparse linear systems. In "Large sparse sets of linear equations". (J.K.Reid, ed.), pp 151-168. Academic Press, New York.

(————,1972) - On the Gaussian elimination for inverting sparse

matrices. Computing (Arch. Elektron. Rechnen) 9,1-7.

(————,1973) - Sparse matrices. Academic Press, New York.

(Warshall,1962) - Warshall,S. A theorem on boolean matrices. J. ACM 9,11-12.

(Wilkinson,1965) - Wilkinson,J.H. The algebraic eigenvalue problem. Oxford Univ. Press, London and New York.

(Young,1971) - Young,D.M. Iterative solution of large linear systems. Copyright by Academic Press, Inc. Orlando. Florida.

(————,1981) - Young,D.M.; L.A.Hageman - Applied iterative methods. Copyright by Academic Press, Inc. Orlando. Florida.

(————,1987) - Young,D.M.; M.Tsun-Zee - Iterative algorithms and software for solving large sparse Linear Systems. CNA-215. The University of Texas, Austin. Texas.