

**Antonio Fernando Rodrigues**

**SOLUÇÃO DE SISTEMAS ESPARSOS DE  
EQUAÇÕES ALGÉBRICAS LINEARES  
POR MÉTODOS DIRETOS**

**Dissertação apresentada à  
Escola Politécnica da  
Universidade de São Paulo, para  
a obtenção do Título de Mestre  
em Engenharia.**

**São Paulo  
1979**

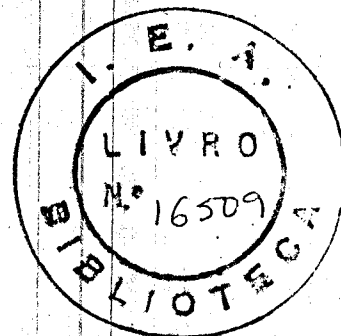
**Antonio Fernando Rodrigues**

Eng. Eletricista, Escola de Engenharia Mauá, 1971

**SOLUÇÃO DE SISTEMAS ESPARSOS DE  
EQUAÇÕES ALGÉBRICAS LINEARES  
POR MÉTODOS DIRETOS**

Dissertação apresentada à  
Escola Politécnica da  
Universidade de São Paulo  
para a obtenção do título  
de Mestre em Engenharia

Orientador: Prof. Dr. Aydin Konuk  
Professor colaborador do Departamento de  
Engenharia Química da UNICAMP



**São Paulo**

**1979**

À dona Beatriz, minha mãe.

## Agradecimentos

Ao dr. Aydin Konuk, orientador deste trabalho

Ao Instituto de Pesquisas Energéticas e Nucleares

Ao Pessoal do Centro de Processamento de Dados do IPEN

A meus colegas da Coordenadoria de Engenharia Nuclear

A todos que direta ou in  
diretamente contribuíram  
na realização deste traba  
lho.

## RESUMO

Foram desenvolvidas quatro subrotinas com o objetivo de calcular sistemas de equações algébricas lineares, aplicando técnicas para matrizes esparsas. As subrotinas SPAMAT (SPAMA1 e SPAMA2) realizam a solução pelo método de eliminação de Gauss, e as subrotinas - MASP (MASP1 e MASP2) através do método da fatorização. As subrotinas SPAMA1 e SPAMA2, e também MASP1 e MASP2, são semelhantes, distinguindo-se somente pelo fato de que a SPAMA2 e MASP2 utilizam operações, armazenadas em seu primeiro processamento, para efetuar cálculos de sistemas  $AX=B$ , onde A permanece fixo e B é variável.

Os testes realizados evidenciaram a vantagem da utilização da SPAMAT e MASP em sistemas onde a matriz dos coeficientes é esparsa. Constatou-se também uma dependência na performance dessas subrotinas com a distribuição dos elementos não nulos dentro da matriz, o que implica numa aplicação mais eficiente de cada uma de acordo com o sistema a ser calculado.

Uma outra subrotina, denominada ORDEM, que realiza uma ordenação inicial da matriz dos coeficientes com o objetivo de evitar a criação de elementos, foi desenvolvida, podendo ser utilizada em conjunto com qualquer uma das outras quatro. Os testes com a ORDEM mostraram que a eficiência de sua utilização depende também da forma da matriz do sistema de equações.

## ABSTRACT

Four subroutines, named SPAMAT( SPAM1 and SPAMA2) and MASP( MASP1 and MASP2), were developed to solve algebraic linear equations by sparse matrix techniques. Solutions are obtained, in SPAMAT, using the Gauss elimination method and in MASP by the factorization method. The SPAM1 and SPAMA2 subroutines( and also the MASP1 and MASP2 subroutines) differ in the method used to resolve a series of equations of the type  $AX=B$ , where the A coefficients are fixed and the B coefficients are variable. In SPAMA2 or MASP2, results obtained for a given system are stored and used in the solution of subsequent systems.

Advantages of the utilisation of SPAMAT and MASP subroutines in systems with sparse matrices are discussed with reference to results obtained for representative tests. The relative efficiencies of the subroutines was shown to be dependent on the distribution of the non zero elements within the matrix.

An additional subroutine, ORDEM, was developed for use with the SPAMAT and MASP subroutines. ORDEM orders the coefficient matrix so as to avoid fill in. Its efficiency depends also on the non zero elements distribution within the matrix.

## Í N D I C E

I - <u>INTRODUÇÃO</u>	.1.
II - <u>TEORIA</u>	.4.
A - Descrição dos métodos diretos	.4.
1. O método de eliminação de Gauss	.4.
2. O método da fatorização	.7.
B - Utilização dos métodos diretos com matrizes esparsas	.10.
1. Armazenamento	.12.
2. Ordenação para manter a esparsidade	.13.
3. Condensação Pivotal	.15.
III - <u>DESCRIÇÃO DAS SUBROTINAS</u>	.17.
A - As subrotinas SPAMAT	.17.
1. Memória de armazenamento	.17.
2. Operações	.18.
3. Condensação Pivotal	.20.
4. Ordenação para manter a esparsidade	.21.
5. Armazenamento para a subrotina SPAMA2	.22.
B - As subrotinas MASP	.23.
1. Memória de armazenamento	.23.
2. Operações	.24.
3. Condensação Pivotal	.26.
4. Ordenação para manter a esparsidade	.27.
5. Armazenamento para a subrotina MASP2	.27.
IV - <u>OS PROGRAMAS DE TESTE</u>	.29.
1. Cálculo da distribuição de temperatura numa barra unidimensional(PT1)	.29.
2. Cálculo da distribuição de temperatura numa placa bidimensional(PT2)	.31.
3. Cálculo da distribuição de temperatura numa placa bidimensional(PT3)	.36.

4. Cálculo da distribuição de temperatura numa placa bidimensional(PT4)	.38.
5. Estudo do escoamento do fluido de carcaça de um trocador de calor equipado com chicanas(PT5)	.40.
6. Dinâmica e controle de um gerador de vapor tipo "passo único"(PT6)	.40.
<u>V - RESULTADOS OBTIDOS E CONCLUSÕES</u>	.43.
A - Resultados obtidos	.43.
1. Resultados obtidos com o PT1	.45.
2. Resultados obtidos com o PT2	.46.
3. Resultados obtidos com os PT3 e PT4	.48.
4. Utilização da ORDEM2 pela subrotina SPAMA1	.49.
5. Resultados obtidos com o PT5	.49.
6. Resultados obtidos com o PT6	.50.
7. Teste com a subrotina SIMQ	.51.
B - Conclusões	.52.
<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	.70.
<u>APÊNDICE I</u>	.73.
subrotina SPAMA1	.74.
subrotina SPAMA2	.78.
<u>APÊNDICE II</u>	.83.
subrotina MASPI	.84.
subrotina MASP2	.88.
<u>APÊNDICE III</u>	.93.
subrotina ORDEM	.94.
<u>APÊNDICE IV: os programas de teste</u>	.96.
PT1 aplicado às subrotinas SPAMAT e MASP	.97.
PT1 aplicado à subrotina GELB	.99.
PT2 aplicado às subrotinas SPAMAT e MASP	.101.
PT2 aplicado à subrotina GELB	.103.
PT3	.105.
PT4	.109.

## LISTA DE FIGURAS

Fig.(II-1) - Obtenção de um elemento $u_{ik}$	.8.
Fig.(II-2) - Obtenção de um elemento $l_{ik}$	.9.
Fig.(II-3) - Representação esquemática de uma matriz com primeira linha e coluna, e diagonal principal, preenchidas por elementos não nulos	.14.
Fig.(IV-1) - Barra AB com comprimento L	.29.
Fig.(IV-2) - Configuração da matriz dos coeficientes para o programa de teste PT1	.30.
Fig.(IV-3) - Placa plana com comprimentos L1 e L2	.31.
Fig.(IV-4) - Configuração da placa para a aplicação do método das diferenças finitas	.33.
Fig.(IV-5) - Enumeração das equações e temperaturas para o programa de teste PT2	.34.
Fig.(IV-6) - Configuração da matriz A para o programa de teste PT2	.35.
Fig.(IV-7) - Enumeração das equações e temperaturas para o programa de teste PT3	.36.
Fig.(IV-8) - Configuração da matriz A para o programa de teste PT3	.37.
Fig.(IV-9) - Enumeração das equações e temperaturas para o programa de teste PT4.	.38.
Fig.(IV-10) - Configuração da matriz A para o programa de teste PT4	.39.
Fig.(IV-11) - Configuração da matriz A para o programa de teste PT5	.41.
Fig.(IV-12) - Configuração da matriz A para o programa de teste PT6	.42.

## LISTA DE TABELAS

Tab.(V-1) - Resultados das subrotinas SPAM1(IS=2), MASPL(IS=2) e GELB para o programa de teste PT1	.54.
Tab.(V-2) - Resultados das subrotinas SPAM1, nas opções IS=1 e IS=2, para o programa de teste PT1	.55.
Tab.(V-3) - Resultados das subrotinas SPAMA2 e GELB para o programa de teste PT1	.56.
Tab.(V-4) - Resultados das subrotinas SPAM1(IS=2), MASPL(IS=2) e GELB para o programa de teste PT2	.57.
Tab.(V-5) - Resultados da subrotina SPAM1, nas opções IS=1 e IS=2, para o programa de teste PT2	.58.
Tab.(V-6) - Resultados da subrotina MASP, nas opções IS=1 e IS=2, para o programa de teste PT2	.59.
Tab.(V-7) - Resultados das subrotinas SPAMA2(IS=2), MASP2(IS=2) e GELB para o programa de teste PT2	.60.
Tab.(V-8) - Resultados da sub.SPAM1, com e sem utilização da sub.ORDEM, para o programa de teste PT2	.61.
Tab.(V-9) - Resultados da sub.SPAM1, com e sem utilização da sub.ORDEM, para o programa de teste PT3	.62.
Tab.(V-10) - Resultados da sub.SPAM1, com e sem utilização da sub.ORDEM, para o programa de teste PT4	.63.
Tab.(V-11) - Resultados da sub.MASPL, com as opções IS=1 e IS=2 para o programa de teste PT3	.64.
Tab.(V-12) - Resultados da sub.MASPL, empregando as opções IS=1, IS=2 e a sub.ORDEM, para o programa de teste PT4	.65.
Tab.(V-13) - Resultados da sub.SPAM1(IS=2) aplicando a ordenação ORDEM2, para os prog. de teste PT2, PT3 e PT4	.66.
Tab.(V-14) - Resultados obtidos com as sub. SPAM1, MASPL e SIMQ, para o programa de teste PT5.	.67.
Tab.(V-15) - Resultados das sub. SPAM1, IS=1 e IS=2, e MASPL, IS=1, para o programa de teste PT6	.68.
Tab.(V-16) - Resultados da subrotina SIMQ nos programas de teste PT1, PT2, PT3 e PT4	.69.

## NOTAÇÃO

- Foi utilizada a representação de matrizes e vetores por letras maiúsculas. A distinção entre ambos é caracterizada no próprio texto.

- Elementos de matrizes e vetores seguem a notação normal, isto é, letras minúsculas com índices minúsculos, um, quando indicando um elemento de um vetor, e dois, de uma matriz.

- Utilizou-se também a representação de elementos de matrizes e vetores por letras maiúsculas com índices, entre parênteses, - maiúsculos. Isso ocorre durante a descrição das subrotinas, sendo essa notação utilizada por analogia aos formatos de impressão dos computadores.

## I - INTRODUÇÃO

Muitos modelos matemáticos requerem a solução por computador de sistemas de equações algébricas lineares onde, frequentemente, a ordem dessas equações torna impraticável sua solução, pois nem sempre se tem à disposição uma máquina com capacidade suficiente para comportar esses sistemas, ou então, por questões econômicas, pois o número de operações envolvidas nesses cálculos, e a necessidade de utilização de memórias auxiliares, acarretam em um tempo de processamento capaz de onerar em demasia o trabalho que se pretende desenvolver.

Felizmente, a maioria desses sistemas possui a particularidade de ter uma porcentagem bem grande de zeros como valores dos coeficientes de suas incógnitas, dando origem a matrizes densas de elementos nulos. Essas matrizes são denominadas de matrizes esparsas, sendo que na solução desses sistemas são empregadas as chamadas "técnicas com matrizes esparsas", cujo objetivo é operar essas matrizes aproveitando sua esparsidade.

Sistemas de equações onde a matriz dos coeficientes é esparsa ocorrem em diversas áreas de aplicação. São encontrados trabalhos realizados em problemas de Análise Numérica, como em soluções de equações diferenciais, estatística e programação dinâmica, de Engenharia, como em estudos de redes elétricas, circuitos de potência, redes hidráulicas, análise estrutural e fotogrametria, e mesmo em Economia e Desenvolvimento Urbano. Essas aplicações justificam o número de trabalhos, relacionados com esse assunto, que têm sido desenvolvidos. Duff / 8 /, por exemplo, apresenta uma bibliografia com cerca de 600 referências, correlatas à pesquisa com matrizes esparsas, publicadas até 1977. Alguns trabalhos significativos são encontrados nas referências /23, 24, 25/, e nas referências / 10,27 / é feita uma análise geral desse assunto.

A motivação para a realização deste trabalho, na Área de Fluidodinâmica do IPEN, ocorreu em virtude de terem sido iniciados estudos, relacionados com termohidráulica / 2,31 / e controle de rea

tores /13/, que deram origem a sistemas de equações onde a matriz dos coeficientes é esparsa. Subrotinas para esse fim já foram desenvolvidas por outros Centros, como, por exemplo, Harwell / 6 /, Maryland, Yale e mesmo pela IBM(SLMATH). No entanto, nenhuma delas se encontra implantada em nosso Centro de Processamento, o que nos levou a optar pelo desenvolvimento das subrotinas aqui apresentadas.

O objetivo deste trabalho, portanto, foi desenvolver, testar e comparar um conjunto de subrotinas para a solução, aplicando técnicas com matrizes esparsas, de sistemas de equações algébricas - através de métodos diretos: Método de Eliminação de Gauss e Método da Fatorização. Esse conjunto é composto de cinco subrotinas, todas escritas em FORTRAN IV. As subrotinas SPAMAT utilizam o método de Gauss, e são constituídas pelas subrotinas SPAM1 e SPAM2. A primeira realiza as operações, de eliminação e substituição regressiva, necessárias para a solução de um único sistema de equações, enquanto que a segunda permite a solução sucessiva, através de armazenamentos apropriados realizados na primeira solução, de sistemas onde somente é variado o lado direito das equações, permanecendo invariável a matriz dos coeficientes, proporcionando uma economia substancial - no tempo de processamento. As subrotinas MASP utilizam o método da fatorização, sendo constituídas pelas subrotinas MASPL e MASP2, cujas finalidades são as mesmas das SPAM1 e SPAM2. Cada uma delas é empregada independentemente, e possuem duas opções de uso: na SPAMAT, com e sem utilização da Condensação Pivotal, e na MASP, com e sem utilização de ordenação de linhas. A quinta subrotina, denominada - ORDEM, é empregada em conjunto com qualquer uma das outras quatro e tem como objetivo manter, através da ordenação de linhas e colunas da matriz dos coeficientes, sua esparsidade.

No cap.II é feita uma descrição dos métodos de eliminação de Gauss e da fatorização de uma forma geral e de sua utilização em matrizes esparsas, com a análise de algumas formas de armazenamento, meios para preservar a esparsidade dessas matrizes e para evitar erros de arredondamento. Estes últimos presentes em todos métodos diretos de solução por computador. No cap.III é feita uma descrição - sucinta da lógica de cada uma das subrotinas. No cap.IV são descritos os programas de teste propostos com o objetivo de verificar o -

funcionamento das subrotinas, procurando abranger os problemas comumente encontrados em soluções desse tipo. Em alguns casos foram utilizadas, para comparação, as subrotinas GELB e SIMQ, constantes da SSP, Biblioteca de Subrotinas Científicas, da IBM. Todos os testes foram realizados em um computador IBM/370, usando a versão OPT=2, com compilador H. No cap.V é feita uma análise dos resultados obtidos, com o emprego dos programas de teste. Finalmente, nos Apêndices, são apresentadas as listagens das subrotinas e feita uma breve descrição de seu modo de utilização.

## II - TEORIA

Neste capítulo é feita uma breve descrição dos métodos diretos de Eliminação de Gauss e da Fatorização, e introduzidas as modificações necessárias considerando a utilização de técnicas com matrizes esparsas, sendo analisados os principais aspectos tendo em vista a utilização dessas técnicas.

### A - Descrição dos Métodos Diretos

Os métodos diretos mais utilizados na solução de sistemas de equações algébricas lineares são o de Eliminação de Gauss, o da Fatorização e o de Gauss-Jordan. Embora exista uma similaridade entre eles, ocorrem diferenças em suas performances, mais pronunciadas em sistemas com matrizes esparsas / 8 /: O que oferece maior vantagem no que se refere ao esforço computacional e que não será utilizado neste trabalho é o de Gauss-Jordan /26 /. Para sistemas com um número extremamente grande de equações, também é utilizado o Método de Eliminação por Bloco /11, 5, 25 /, em que a matriz dos coeficientes é dividida em partições que são processadas independentemente.

#### 1 - O Método de Eliminação de Gauss

O método de Gauss consiste em transformar a matriz A do sistema  $AX=B$  em uma matriz triangular superior, através de operações entre suas linhas dadas pelas expressões:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} \cdot a_{kj}^{(k-1)} / a_{kk}^{(k-1)} \quad (\text{II-1})$$

com  $a_{ij}$  pertencente a A,  $k=1, N-1$ ,  $i=k+1, N$ ,  $j=k, N$ , onde N é o número de equações do sistema, e

$$b_i^{(k)} = b_i^{(k-1)} - a_{ik}^{(k-1)} \cdot b_k^{(k-1)} / a_{kk}^{(k-1)} \quad (\text{II-2})$$

com  $k=1, N-1$ ,  $i=k+1, N$  e  $b$  pertencente ao vetor B.

Chamaremos a linha com o índice  $k$  de linha base, enquanto que  $a_{kk}$  é comumente denominado elemento pivô, devendo ser necessá-

mente diferente de zero. Conforme mostra a expressão (II-1), para cada linha base  $k$ , serão feitas combinações com as linhas, designadas - pelo índice  $i$ , situadas abaixo de  $k$ . As operações entre os elementos das linhas  $k$  e  $i$  são realizadas a partir dos elementos indicados pelo índice  $j$ , localizados na coluna  $k$  pois, aqueles situados em colunas - anteriores foram eliminados (colocados iguais a zero) em passos de eliminação anteriores. Para  $j=k$ ,  $a_{ik}^{(k)}=0$ , para cada linha  $i$  escolhida, o que significa que todos os elementos  $a_{ik}$  situados na coluna  $k$ , abaixo do pivô, são eliminados. Da mesma forma, na eq. (II-2), para cada linha  $i$ , o termo correspondente,  $b_i$ , situado no vetor B, será modificado conforme mostra a expressão, sendo  $b_k$  o elemento correspondente na linha base.

Exemplificando, seja um sistema de N equações, com A e B - conforme mostrados abaixo:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2k} & \dots & a_{2N} \\ a_{31} & a_{32} & \dots & a_{3k} & \dots & a_{3N} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{k1} & a_{k2} & \dots & a_{kk} & \dots & a_{kN} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{N1} & a_{N2} & \dots & a_{Nk} & \dots & a_{NN} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdot \\ b_k \\ \cdot \\ b_N \end{bmatrix}$$

Após o 1º passo de eliminação teríamos,

$$A^{(1)} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1N} \\ 0 & a_{22}^{(1)} & \dots & a_{2k}^{(1)} & \dots & a_{2N}^{(1)} \\ 0 & a_{32}^{(1)} & \dots & a_{3k}^{(1)} & \dots & a_{3N}^{(1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & a_{k2}^{(1)} & \dots & a_{kk}^{(1)} & \dots & a_{kN}^{(1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & a_{N2}^{(1)} & \dots & a_{Nk}^{(1)} & \dots & a_{NN}^{(1)} \end{bmatrix} \quad B^{(1)} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(1)} \\ \cdot \\ b_k^{(1)} \\ \cdot \\ b_N^{(1)} \end{bmatrix}$$

Após um k-ésimo passo,

$$A^{(k)} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1N} \\ 0 & a_{22}^{(1)} & \dots & a_{2k}^{(1)} & \dots & a_{2N}^{(1)} \\ 0 & 0 & \dots & a_{3k}^{(2)} & \dots & a_{3N}^{(2)} \\ \cdot & \cdot & & \cdot & & \cdot \\ 0 & 0 & \dots & a_{kk}^{(k-1)} & \dots & a_{kN}^{(k-1)} \\ \cdot & \cdot & & \cdot & & \cdot \\ 0 & 0 & \dots & 0 & \dots & a_{NN}^{(k)} \end{bmatrix} \quad B^{(k)} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \cdot \\ b_k^{(k-1)} \\ \cdot \\ b_N^{(k)} \end{bmatrix}$$

E, finalmente, após o (N-1)ésimo passo,

$$A^{(N-1)} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} & \dots & a_{1N} \\ 0 & a_{22}^{(1)} & \dots & a_{2k}^{(1)} & \dots & a_{2N}^{(1)} \\ 0 & 0 & \dots & a_{3k}^{(2)} & \dots & a_{3N}^{(2)} \\ \cdot & \cdot & & \cdot & & \cdot \\ 0 & 0 & \dots & a_{kk}^{(k-1)} & \dots & a_{kN}^{(k-1)} \\ \cdot & \cdot & & \cdot & & \cdot \\ 0 & 0 & \dots & 0 & \dots & a_{NN}^{(N-1)} \end{bmatrix} \quad B^{(N-1)} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \cdot \\ b_k^{(k-1)} \\ \cdot \\ b_N^{(N-1)} \end{bmatrix}$$

Uma vez triangularizada a matriz A, as incógnitas serão obtidas por substituição regressiva através das expressões:

$$x_N = b_N / a_{NN} \quad (\text{II-3})$$

$$x_i = 1/a_{ii} (b_i - \sum_{j=i+1}^N a_{ij} \cdot x_j) \quad (\text{II-4})$$

com  $i=N-1, 1$ .

Conforme pode ser determinado através das operações acima mostradas, para cada passo  $k$  de eliminação são realizadas  $N-(k+1)$  - combinações lineares da equação  $k$  do sistema com as equações subsequentes. Efetuando-se os (N-1) passos serão necessárias  $(N^3 + 3N^2 - N)/3$  operações de multiplicação e soma entre os elementos de A, e que definem o esforço de computação requerido. Além disso, é necessário armazenar os  $N^2$  elementos de A, que definem a memória utilizada para a solução do sistema.

2 - O Método da Fatorização

O método da fatorização consiste em transformar a matriz dos coeficientes A, de um sistema  $AX=B$ , no produto  $A=LU_1$ , onde L é uma matriz triangular inferior e  $U_1$ , matriz triangular superior com 1's em sua diagonal principal. Similarmente, A poderia ser transformada no produto  $L_1U$ , onde, nesse caso,  $L_1$  seria formada com 1's em sua diagonal principal (o primeiro é conhecido como método de Crout e, o segundo, Gauss-Doolittle). Assim, um sistema de equações algébricas  $AX=B$  é resolvido através das expressões matriciais:

$$LU_1X = B, \text{ com} \tag{II-5}$$

$$Y = L^{-1}B, \text{ e} \tag{II-6}$$

$$U_1X = Y \tag{II-7}$$

Seja, por exemplo, uma matriz A conforme esquematizada abaixo:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1,N-1} & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2,N-1} & a_{2N} \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ a_{N-1,1} & a_{N-1,2} & \dots & a_{N-1,N-1} & a_{N-1,N} \\ a_{N1} & a_{N2} & \dots & a_{N,N-1} & a_{NN} \end{bmatrix}$$

As matrizes L e U (fica subtendido U como sendo a matriz  $U_1$  acima definida) teriam a forma:

$$L = \begin{bmatrix} 1_{11} & 0 & \dots & 0 & 0 \\ 1_{21} & 1_{22} & \dots & 0 & 0 \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ 1_{N-1,1} & 1_{N-1,2} & \dots & 1_{N-1,N-1} & 0 \\ 1_{N1} & 1_{N2} & \dots & 1_{N,N-1} & 1_{NN} \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & u_{12} & u_{13} & \dots & u_{1,N-1} & u_{1N} \\ 0 & 1 & u_{23} & \dots & u_{2,N-1} & u_{2N} \\ 0 & 0 & 1 & \dots & u_{3,N-1} & u_{3N} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & u_{N-1,N} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix},$$

sendo cada elemento  $a_{ij}$  de A igual ao produto de uma linha  $i$  de L por uma coluna  $j$  de U. Os valores dos coeficientes de L e U, que se desejam obter, são determinados pelas expressões:

$$l_{ii} = a_{ii}, \quad i=1,N \tag{II-8}$$

$$u_{li} = a_{li}/l_{ii}, \quad i=1,N \tag{II-9}$$

$$u_{ik} = 1/l_{ii} \cdot (a_{ik} - \sum_{i_1=1}^{i-1} l_{ii_1} \cdot u_{i_1k}), \tag{II-10}$$

com  $k=i+1,N$  e  $i=2,N-1$

$$u_{NN} = 1 \tag{II-11}$$

$$l_{ik} = a_{ik} - \sum_{i_1=1}^{k-1} l_{ii_1} \cdot u_{i_1k}, \tag{II-12}$$

com  $k=2,i$  e  $i=2,N$ , onde N é o número de equações do sistema.

A obtenção de cada  $u_{ik}$  de U e  $l_{ik}$  de L pode ser esquematizada conforme mostram as figuras (II-1) e (II-2).

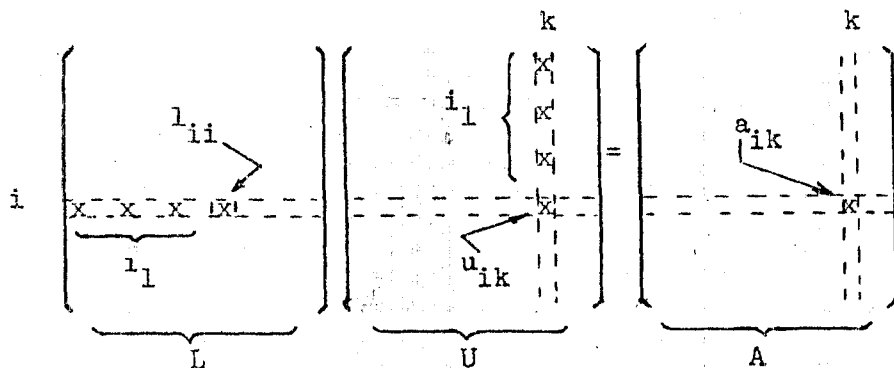


Fig.(II-1): obtenção de um elemento  $u_{ik}$

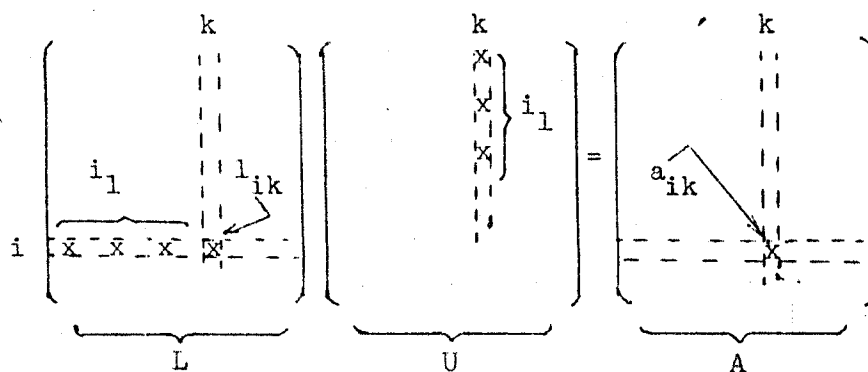


Fig. (II-2): obtenção de um elemento  $l_{ik}$

As expressões (II-8) e (II-9) são decorrentes, respectivamente, de (II-12) e (II-10), considerando-se que, para  $i=1$  as somatórias em (II-12) e (II-10) são nulas. A expressão (II-11) provém da própria definição dada a U.

Obtidas as matrizes L e U, determina-se Y, em (II-6), por substituição direta de L sobre B, através das expressões:

$$y_1 = b_1 / l_{11} \quad e, \quad (II-13)$$

$$y_i = 1/l_{ii} \cdot (b_i - \sum_{j=1}^{i-1} l_{ij} \cdot b_j), \quad \text{com } i=2, N \quad (II-14)$$

Uma vez determinado Y, obtem-se X por substituição regressiva de U sobre Y, através das expressões:

$$x_N = y_N \quad e, \quad (II-15)$$

$$x_i = y_i - \sum_{j=i+1}^N u_{ij} \cdot y_j, \quad \text{com } i=2, N-1 \quad (II-16)$$

Examinando-se as expressões que definem o método da fatorização, verifica-se que o esforço de computação para a solução de um sistema de equações através desse processo é o mesmo que o do método de eliminação de Gauss, ou seja, são necessárias efetuar  $(N^3 + 3N^2 - N)/3$  operações de multiplicação e adição. Considerando as matrizes L e U formadas sobre a matriz A, são necessárias também  $N^2$  posições para o armazenamento dos elementos de L e U.

## B - Utilização dos Métodos Diretos com Matrizes Esparsas

Considerando agora as mesmas soluções vistas no caso geral aplicadas a sistemas onde A é uma matriz esparsa, muitas das operações realizadas envolvem multiplicações por zero, que não influem na solução final.

Analisando-se a exp. (II-1), aplicada no método de eliminação de Gauss, podemos observar que:

- para elementos a serem eliminados,  $a_{ik}$ , com valor nulo, os elementos  $a_{ij}$  permanecerão inalterados, não havendo portanto necessidade de realizar operações entre os elementos dessa linha  $i$  com a linha base  $k$ , ou seja, não é necessário eliminar os zeros.

- se  $a_{ik}$  for diferente de zero, as operações entre elementos da linha  $i$  com a linha  $k$  também não serão necessárias para valores de  $a_{ij}$  e  $a_{kj}$  iguais a zero, ou quando somente  $a_{kj}$  é nulo pois o valor de  $a_{ij}$  permanecerá inalterado. O mesmo não ocorre se  $a_{ij}$  e  $a_{kj}$  forem não nulos ou, ainda, se  $a_{ij}$  for nulo e  $a_{kj}$  não o for. Nesse caso  $a_{ij}$  assumirá um novo valor, agora diferente de zero, ocorrendo a "criação de um novo elemento".

Tendo em vista essas observações, podemos reescrever as expressões utilizadas no método de Gauss, levando em consideração a não utilização de elementos nulos. Assim teríamos:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} / a_{kk}^{(k-1)} \cdot a_{kj}^{(k-1)} \quad (\text{II-17})$$

com  $k=1, N-1$ ,  $i=k+1, N$ ,  $j=k, N$ ,  $a_{ik} \neq 0$  e  $a_{kj} \neq 0$ ,

$$b_i^{(k)} = b_i^{(k-1)} - a_{ik}^{(k-1)} \cdot b_k^{(k-1)} / a_{kk}^{(k-1)} \quad (\text{II-18})$$

com  $k=1, N-1$ ,  $i=k+1, N$  e  $a_{ik} \neq 0$ , e:

$$x_i = 1/a_{ii} \cdot (b_i - \sum_{j=i+1}^N a_{ij} \cdot x_j) \quad (\text{II-19})$$

com  $i=N-1, 1$  e  $a_{ij} \neq 0$ , esta última aplicada à substituição regressiva.

Aplicando-se raciocínio análogo nas expressões utilizadas

no método da fatorização obtem-se o seguinte conjunto de equações:

$$l_{il} = a_{il} , \text{ com } i=1, N \text{ e } a_{il} \neq 0 \quad (\text{II-20})$$

$$u_{li} = a_{li}/l_{11} , \text{ com } i=1, N \text{ e } a_{li} \neq 0 \quad (\text{II-21})$$

$$l_{ik} = a_{ik} - \sum_{i_1=1}^{k-1} l_{ii_1} \cdot u_{i_1k} , \quad (\text{II-22})$$

com  $k=2, i, i=2, N, l_{ii_1} \neq 0$  e  $u_{i_1k} \neq 0$

$$u_{ik} = 1/l_{ii} \cdot (a_{ik} - \sum_{i_1=1}^{i-1} l_{ii_1} \cdot u_{i_1k}) , \quad (\text{II-23})$$

com  $k=i, i=1, N, i=2, N-1, l_{ii_1} \neq 0$  e  $u_{i_1k} \neq 0$

$$u_{NN} = 1 \quad (\text{II-24})$$

$$y_1 = b_1/l_{11} \quad (\text{II-25})$$

$$y_i = 1/l_{ii} \cdot (b_i - \sum_{j=1}^{i-1} l_{ij} \cdot b_j) , \quad (\text{II-26})$$

com  $i=2, N$  e  $l_{ij} \neq 0$ . As duas últimas aplicadas na substituição direta, e

$$x_N = y_N \text{ e} \quad (\text{II-27})$$

$$x_i = y_i - \sum_{j=1}^N u_{ij} \cdot y_j , \quad (\text{II-28})$$

com  $i=N-1, 1$  e  $u_{ij} \neq 0$ , aplicadas na substituição regressiva.

No método da fatorização, a "criação de um elemento" ocorre quando para um  $a_{ik} = 0$  resultem valores de  $l_{ik}$  ou  $u_{ik}$  diferentes de zero, devido à coincidência de  $l_{ii_1}$  e  $u_{i_1k}$  com valores não nulos nas expressões (II-22) e (II-23) para a obtenção, respectivamente, de  $l$  e  $u$ .

## 1 - Armazenamento

Ao se realizar o armazenamento de matrizes esparsas, leva-se em conta a vantagem da utilização de somente elementos não nulos da matriz. Isso, naturalmente, implica numa considerável economia de memória. Entretanto, outras informações, concernentes ao posicionamento real desses elementos e que estão implicitamente definidas quando todos os elementos, nulos ou não, da matriz são considerados, são necessárias de serem fornecidas. Assim, existem dois tipos de armazenamento associados na definição dos coeficientes de matrizes esparsas: um, armazenamento primário, usado para os valores dos elementos não nulos da matriz, e outro, armazenamento secundário, requerido por indicadores de posicionamento e outras informações sobre a estrutura da matriz. Portanto, faz-se necessário, ao definir-se um esquema de armazenamento, levar em conta uma forma capaz de conciliar esses dois tipos. Uma economia excessiva de informações contidas no armazenamento primário como, por exemplo, eliminando-se ao máximo o armazenamento de elementos nulos, pode, na maioria das vezes, requerer uma quantidade maior de informações para o armazenamento secundário, o que é inconveniente tanto sob o ponto de vista de memória de computação como em tempo de processamento.

Segundo alguns autores / 8 , 5 /, não existe um esquema de armazenamento capaz de satisfazer plenamente em cada caso em que é aplicado. O problema maior encontrado no manuseio de matrizes em sua forma esparsa reside, sem dúvida, na criação de elementos. Essa criação, como mostrado no item a seguir, depende da forma da matriz utilizada. É difícil saber, de antemão, quantos e aonde elementos nulos tornar-se-ão não nulos e, sempre que isso ocorrer, é necessário ter-se posições previstas tanto para armazenar esse novo valor como para definir a sua posição, além das modificações necessárias de posicionamento desses elementos criados entre os outros já existentes, de acordo com a lógica do armazenamento utilizado.

Vários esquemas de armazenamento têm sido propostos. Basicamente, podem ser classificados em dois grupos: armazenamentos estáticos e dinâmicos. A diferença fundamental entre esses dois tipos encontra-se justamente na facilidade de adoção de elementos criados.

Entre os armazenamentos estáticos, o mais comumente empregado é o chamado "indicador de coluna/ indexador de linha" ou, semelhantemente, "indicador de linha/ indexador de coluna" /7, 4/. Nessses esquemas utilizam-se um vetor real para armazenamento dos valores não nulos da matriz em seqüência crescente de suas linhas, um vetor inteiro com a posição das colunas às quais pertencem cada elemento armazenado e um outro vetor inteiro com o endereço, em cada posição  $i$ , do primeiro elemento não nulo localizado na linha  $i$  correspondente.

Os esquemas dinâmicos de armazenamento utilizam as chamadas listas de ligação (linked lists) /28, 9, 6/. Nesses esquemas não existe uma ordem de armazenamento dos elementos, sendo fornecidas informações adicionais que propiciem a localização de cada elemento a ser processado à medida que sejam necessários. Assim, por exemplo, os elementos localizados numa determinada linha da matriz não necessitam estar em posições consecutivas pois, a cada elemento esta associada a posição daquele que o sucede.

Na escolha de um esquema de armazenamento para as subrotinas desenvolvidas neste trabalho procurou-se encontrar uma solução visando eliminar, o quanto possível, informações e trabalhos requeridos para a localização dos elementos da matriz, permitindo-se um acréscimo na memória de armazenamento. Foi utilizado um esquema estático, com elementos dispostos em forma matricial, com cada linha dessa matriz correspondendo à mesma linha na matriz dos coeficientes, sendo reservadas posições, em cada linha, para a introdução de elementos criados.

## 2 - Ordenação para manter a esparsidade

Para obter as vantagens de armazenamento e tempo de processamento encontradas na solução de sistemas com matrizes esparsas, é importante reduzir ao mínimo a criação de elementos durante o processo de eliminação.

Essa criação depende essencialmente da distribuição dos elementos dentro da matriz. Por exemplo, considerando-se uma matriz

esparsa A, conforme visto na fig. (II-3) abaixo, onde sua primeira -  
 linha e coluna possuem todos seus elementos não nulos, a criação de  
 elementos, durante as eliminações, ocasionará um preenchimento total  
 de A. Por outro lado, se a linha l for escolhida como última linha  
 base, não haverá criação alguma, sendo mantida a esparsidade de A du-  
 rante todo processamento.

$$A = \begin{bmatrix} x & x & x & x & x & \dots & x & x \\ x & x & & & & & & \\ x & & x & & & & & \\ x & & & x & & & & \\ \cdot & & & & \cdot & & & \\ \cdot & & & & & \cdot & & \\ \cdot & & & & & & \cdot & \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ x & & & & & & & x \end{bmatrix}$$

Fig. (II-3): representação esquemática de uma matriz com primeira linha e coluna, e diagonal principal, preenchidas por elementos não nulos.

Conforme visto no exemplo, uma maneira de otimizar a criação de elementos, normalmente utilizada, é através da realização de arranjos entre linhas ou colunas da matriz.

A maioria das técnicas utilizadas para ordenação estão enquadradas dentro de duas categorias / 7 /: métodos à priori e métodos locais. Os primeiros realizam a ordenação de linhas e colunas inicialmente ou antes de cada passo de eliminação, colocando-as em ordem crescente de número de elementos, podendo ser feita tanto para linhas como para colunas, ou para ambas. Esses métodos têm sido muito utilizados / 29,20 , 30 /, mostrando resultados que, se não chegam a ser ideais, possuem vantagens sobre seu não utilização.

Nos métodos locais, a ordenação é feita escolhendo como linha base, através da verificação da distribuição dos elementos na

matriz, aquela que ocasionará menor criação de elementos. Vários métodos usando ordenação local têm sido propostos / 14, 15, 16, 19, 3 / porém, o mais popular é o de Markowitz / 18 /. No critério de Markowitz, basicamente, são realizadas mudanças, em cada passo  $k$  de eliminação, entre linhas  $i$  e colunas  $j$ , com  $i > k$  e  $j > k$ , de maneira a minimizar o produto  $N(i)N(j)$ , onde  $N(i)$  dá o número de não zeros numa linha  $i$  e  $N(j)$  o número de não zeros numa coluna  $j$ .

Conforme comenta Duff / 8 /, não existe um algoritmo capaz de realizar uma ordenação ótima em matrizes esparsas de maneira geral, ou seja, o que pode ser bem aplicado em um determinado tipo de matriz poderá não ser em outro tipo.

### 3 - Condensação Pivotal

Os métodos diretos para solução de sistemas de equações, embora exatos, estão sujeitos eventualmente a erros de arredondamento que podem conduzir a soluções errôneas. Para evitar a ocorrência desses erros é importante manter, durante o processo de eliminação, os valores de  $A$  baixos em relação aos valores iniciais dessa matriz - / 22 /. A forma mais comum de atingir essa condição é escolhendo, em cada passo de eliminação, o maior elemento, em módulo, de  $A$  como pivô.

Geralmente são utilizadas duas formas de condensação pivotal / 32 /, ambas baseadas no princípio acima, denominadas Condensação Pivotal Parcial e Condensação Pivotal Completa. A primeira escolhe como pivô o maior elemento situado na coluna  $k$ , correspondente ao estágio de eliminação considerado, em linhas de  $k$  a  $N$ , isto é,

$$|a_{kk}^{(k)}| = \max_i |a_{ik}^{(k)}|, \quad k \leq i \leq N.$$

Essa escolha pode também ser feita em colunas, de  $k$  a  $N$ . A segunda, faz a escolha do maior elemento dentre aqueles situados em colunas e linhas a partir de  $k$  até  $N$ .

$$|a_{kk}^{(k)}| = \max_{i,j} |a_{ij}^{(k)}|, \quad k \leq i, j \leq N.$$

A utilização simultânea da condensação pivotal e dos métodos de ordenação, baseados na efetuação de mudanças entre linhas e colunas, poderá ocasionar uma ineficiência na utilização de um ou outro, ou de ambos, tornando-os inoperantes. Nesse caso, abre-se mão, quando possível, da condensação pivotal, empregando-se para a escolha dos pivôs a chamada Tolerância Relativa / 7,25,28 /. Em geral, define-se essa tolerância através da multiplicação do maior elemento da matriz, em módulo, por um fator EPS,  $0 < \text{EPS} < 1$ , de tal forma que, um elemento da matriz para ser escolhido como pivô deverá ser maior que essa tolerância. EPS é escolhido levando-se em conta a máquina onde é aplicado o algoritmo, possuindo um valor típico de  $10^{-7}$ . Naturalmente, o rigor para a escolha do pivô dependerá do valor assumido por EPS: para valores próximos a 1, esse rigor será maior, o contrário ocorrendo para valores próximos a zero. A Tolerância Relativa pode ser definida apenas uma vez, antes do início do processo de eliminação ou, antes de cada passo  $k$  de eliminação, aqui escolhendo o maior elemento de A nesse passo, em linhas e colunas a partir de  $k$ .

### III - DESCRIÇÃO DAS SUBROTINAS

Neste capítulo são descritas as subrotinas SPAMAT e MASP que utilizam, respectivamente, o método de eliminação de Gauss e da fatorização. São abordados aspectos da lógica dessas subrotinas, considerando-se sua aplicação com matrizes esparsas, baseados no desenvolvimento realizado no Cap.II.

#### A - As subrotinas SPAMAT

Duas subrotinas foram desenvolvidas para uso da eliminação de Gauss em soluções de sistemas de equações com matrizes esparsas. A primeira, denominada SPAM1, resolve um sistema  $AX=B$  uma única vez. A segunda, denominada SPAM2, semelhante à primeira, é composta de duas partes: a primeira parte resolve um sistema  $AX=B_1$  e, enquanto triangulariza a matriz A, armazena as posições e valores dos elementos pivô e eliminados, respectivamente  $a_{kk}$  e  $a_{ik}$  na exp. (II-1), o que permite as operações sobre B sem repetir a triangularização, como pode-se ver na exp. (II-2). Para um outro sistema  $AX=B_2$ , onde A permanece invariável, é utilizada a segunda parte da subrotina, sendo sua solução realizada efetuando-se somente as operações sobre B e utilizando A, já triangularizada, para a substituição regressiva.

A forma de utilização dessas subrotinas, bem como suas listagens, são encontradas no Apêndice I.

#### 1 - Memória de armazenamento

As subrotinas SPAMAT utilizam o armazenamento estático dos coeficientes da matriz A, realizado através de A(I,J), IC(I,J) e INZ(I). A matriz A(I,J) armazena os valores não nulos dos elementos da matriz dos coeficientes. O acesso às linhas de A é direto, isto é, cada linha I de A(I,J) corresponde a uma linha I de A. As colunas a que pertencem os elementos da matriz dos coeficientes A são definidas através da matriz IC(I,J), sendo que a correspondência entre o valor de um coeficiente de A e de sua posição na coluna é feita atra

vés de J. O vetor  $INZ(I)$  é usado para indicar o número total de elementos não nulos armazenados em cada linha de  $A(I,J)$ , ou seja,  $A(I,INZ(I))$  corresponde ao último elemento não nulo localizado numa linha I de  $A(I,J)$ , enquanto que,  $IC(I,INZ(I))$  indica a coluna à qual pertence esse elemento. Os valores do vetor B, lado direito do sistema, são armazenados no vetor  $B(I)$ . Finalmente, são usados  $X(I)$  e  $IX(I)$  como vetores auxiliares nas operações de eliminação, sendo que  $X(I)$  é também empregado como saída para os resultados obtidos.

## 2 - Operações

Considerando-se as expressões que descrevem o método de Gauss levando em consideração a utilização de matrizes esparsas, vistas no cap. II, e os arranjos vistos no ítem anterior, podemos, mais uma vez, reescrever aquelas expressões, agora sob o ponto de vista de sua utilização nas subrotinas SPAMAT, obtendo-se:

$$A^{(K)}(K, J_1) = A^{(K-1)}(K, J_1) / A^{(K-1)}(K, J_2), \quad (III-1)$$

com  $K=1, N-1$ ,  $J_1=1, INZ(K)$  e  $J_2$  tal que  $IC(K, J_2)=K$ .

$$A^{(K)}(I, J_3) = A^{(K-1)}(I, J_3) - A^{(K-1)}(I, J_4) \cdot A^{(K)}(K, J_1), \quad (III-2)$$

com  $K=1, N-1$ ,  $I=K+1, N$ ,  $J_3=1, INZ(I)$ ,  $J_1=1, INZ(K)$  e  $J_4$  tal que  $IC(I, J_4)=K$

$$B^{(K)}(K) = B^{(K-1)}(K) / A^{(K-1)}(K, J_2), \quad (III-3)$$

com  $K=1, N-1$  e  $J_2$  tal que  $IC(K, J_2)=K$

$$B^{(K)}(I) = B^{(K-1)}(I) - A^{(K-1)}(I, J_4) \cdot B^{(K)}(K), \quad (III-4)$$

com  $K=1, N-1$ ,  $I=K+1, N$  e  $J_4$  tal que  $IC(I, J_4)=K$

$$X(I) = B(I) - X(IC(I, J)) \cdot A(I, J) \quad (III-5)$$

com  $I=N-1, 1$  e  $J=1, INZ(I)$ , tal que  $IC(I, J)=I$

$$X(N) = B(N) / A(N, J) \quad (III-6)$$

com J talque  $IC(N, J)=N$ .

Notar que as expressões (II-17) e (II-18), mostradas no •

cap. II, foram desmembradas, cada uma delas em duas, respectivamente em (III-1) e (III-2), (III-3) e (III-4). Esse desmembramento mostra que, inicialmente, toda a linha base escolhida e o termo independente correspondente são divididos por seu elemento pivô. Como consequência, nas expr. (III-2) e (III-4) são utilizados os coeficientes da linha base  $K$  e do termo independente já modificados, enquanto que na exp. (III-5), para a substituição regressiva, não são feitas as divisões pelo pivô, pois esse terá o valor 1 após as operações em (III-1).

A obtenção do pivô,  $A(K, J_2)$  em (III-1) e (III-3), requer a procura em IC da posição  $J_2$  em que está localizado, o mesmo ocorrendo com o elemento a ser eliminado,  $A(I, J_4)$  em (III-2) e (III-4). Além do mais, para cada passo de eliminação, será necessário verificar em cada linha  $I$ ,  $I=K+1, N$ , a existência de  $A(I, J_4)$ , para que sejam feitas as operações com a linha base. Esses fatos adicionados acarretarão um trabalho extra de computação requerido para a procura desses elementos e que, certamente, ocasionará um aumento no tempo de processamento da subrotina. Para amenizar essa procura, as matrizes  $A(I, J)$  e  $IC(I, J)$  são montadas em ordem crescente de coluna para cada linha  $I$  pois, considerando-se que os elementos  $A(I, J_4)$  são realmente eliminados de suas posições de armazenamento, através do deslocamento de todos os elementos pertencentes à linha  $I$  correspondente, cada elemento  $A(K, J_2)$  e  $A(I, J_4)$ , se existir, estará localizado na primeira posição da linha considerada, isto é,  $A(K, 1)$  e  $A(I, 1)$ .

Por outro lado, levando em conta essa disposição dos elementos em ordem crescente, para cada novo elemento criado numa linha  $I$  será necessário realizar deslocamentos de elementos de  $A(I, J)$  e  $IC(I, J)$ , com o objetivo de abrir-se espaço para seu armazenamento.

Finalmente, na exp. (III-2), para cada valor  $A(K, J_1)$  será necessário verificar a existência ou não de um  $A(I, J_3)$  localizado numa mesma coluna, para que sejam efetuadas as operações entre esses elementos, ou a criação de um novo elemento, o que também acarretará em procuras e comparações entre elementos de  $IC(I, J_3)$  e  $IC(K, J_1)$ .

Visando diminuir o tempo gasto com as operações acima, é

utilizado um vetor  $X(I)$ , onde as linhas  $A(I, J_3)$  e  $A(K, J_1)$ , em (III-2), são transportadas em sua forma expandida, isto é, com seus elementos situados em posições de  $X(I)$  correspondentes às colunas a qual pertencem em  $A$  para, em seguida, serem realizadas as operações. A utilização desse vetor  $X$  ocorre da seguinte maneira:

- Inicialmente  $X(I)$  é colocado igual a zero na faixa onde estão situados os elementos das linhas  $K$  e  $I$ , desde a coluna  $K$  até a coluna na posição mais extrema, ou seja,

$$X(L)=0, \text{ para } L=K, KN \text{ com } KN=IC(K, INZ(K)) \text{ se } IC(K, INZ(K)) > IC(I, INZ(I)), \text{ ou } KN=IC(I, INZ(I)) \text{ se } IC(I, INZ(I)) \geq IC(K, INZ(K)).$$

- Em seguida, são transportados para  $X(I)$ , em forma expandida, os elementos situados na linha  $I$ ,

$$X(IC(I, J_3))=A(I, J_3), \text{ com } J_3=1, INZ(I).$$

- Da mesma forma, são transportados os elementos da linha base  $K$ , e efetuadas as operações com os valores já armazenados em  $X(I)$ ,

$$X(IC(K, J_1))=X(IC(K, J_1)) - A(I, J_4) \cdot A(K, J_1), \text{ com } J_1=1, INZ(K),$$

sendo que, para  $X(IC(K, J_1))=0$ , no 2º termo, será criado um novo elemento.

- Por fim, os valores resultantes em  $X$ , não nulos, na faixa de operação considerada, e que correspondem aos valores modificados da linha  $I$ , são retransportados para a matriz  $A(I, J)$ .  $IC(I, J)$  e  $INZ(I)$  acompanham as transformações efetuadas nessa linha,

$$A(I, J_3)=X(L), \text{ com } L=K, KN \text{ tal que } X(L) \neq 0,$$

$$IC(I, J_3)=L, \text{ com } L=K, KN \text{ tal que } X(L) \neq 0,$$

$$INZ(I)=NZ, \text{ onde } NZ \text{ é o número de valores } X(L) \neq 0 \text{ que foram transportados para } I.$$

### 3 - Condensação Pivotal

Da mesma forma como utilizado nas subrotinas GELB e SIMQ, para cálculo de sistemas de equações algébricas, adotou-se a Condensação Pivotal Parcial nas subrotinas SPAM1 e SPAM2. Assim, em cá-

da passo de eliminação, é escolhido como pivô o maior elemento em módulo da coluna correspondente ao pivô atual em linhas situadas a partir da linha base atual. Uma vez determinado esse elemento, é feita uma troca de linhas, passando para a posição de linha base aquela que contém o novo pivô. Essa mudança entre linhas é efetuada indiretamente, através do vetor auxiliar IX(I).

Considerando-se que nem sempre haverá a necessidade de utilização da Condensação Pivotal, existem duas opções de funcionamento das subrotinas SPAMAT: uma realizando a Condensação, e a outra utilizando somente uma Tolerância Relativa como teste de aceitação de um pivô.

Para evitar a escolha de zeros como pivô e para verificar a existência de singularidades no sistema de equações, compara-se cada valor de pivô escolhido, em ambas opções, com essa Tolerância Relativa (TOL). TOL é obtido pela expressão / 17/:

$$TOL = EPS \cdot |A(I,J)|, \quad (III-7)$$

onde A(I,J) é o maior valor, em módulo, da matriz A inicial.

#### 4 - Ordenação para manter a esparsidade

Foram adotadas duas formas de ordenação, ambas a priori, chamadas ORDEM1 e ORDEM2. A primeira é apresentada como subrotina, denominada ORDEM, e, a segunda, foi desenvolvida dentro da própria SPAMAT. A subrotina ORDEM efetua inicialmente um arranjo nas colunas de A, enumerando-as em ordem crescente do número de elementos em suas colunas, ou seja, as primeiras colunas de A terão menor número de elementos, enquanto que as últimas maior. Depois, procedimento semelhante é realizado com as linhas de A, agora modificada em virtude da ordenação das colunas. As novas posições das linhas de A, após a ordenação, são armazenadas no vetor IY(I), I=1,N, pois não são feitas mudanças efetivas entre suas linhas. Dessa forma, as linhas K, I e N utilizadas nas expressões (III-1) a (III-6), são definidas através do vetor IY, da seguinte maneira:

$$K = IY(J), J=1, N-1 \quad (\text{III-8})$$

$$I = IY(L), L=J+1, N \quad (\text{III-9})$$

$$N = IY(N) \quad (\text{III-10})$$

As mudanças efetuadas com as colunas de A necessitam ser acompanhadas pelas incógnitas X do sistema. Sendo assim, define-se um vetor IZ(I), de forma que:

$$X(I) = X(IZ(I)), I=1, N \quad (\text{III-11})$$

A forma de utilização dessa subrotina e sua listagem encontram-se no Apêndice III.

A ORDEM2 foi utilizada dentro da SPAM1 na opção onde não é feita a Condensação Pivotal. Para cada passo de eliminação é escolhida a linha base entre as que possuem elementos situados na coluna K correspondente ao pivô e que tenham o menor número de elementos. Dessa forma, nessa ordenação, a escolha da linha base é feita ao mesmo tempo em que são procurados os elementos a serem eliminados, situados em linha I e coluna K, na exp. (III-2). Sendo a definição de um pivô realizada antes do início, propriamente dito, de cada passo de eliminação, a informação das linhas I que serão processadas é armazenada no vetor auxiliar IX(I), I=1, N, evitando-se que sejam procuradas novamente.

##### 5. Armazenamento para a subrotina SPAMA2

A subrotina SPAMA2 realiza o processo de eliminação de forma idêntica à SPAM1. No entanto, para obter vantagens no tempo de processamento, são armazenados valores que serão posteriormente utilizados. Com isso, além de A(I, J), IC(I, J), INZ(I) e IX(I), são empregados também os vetores AB(I) e IBM(I), com I=1, LB, e, IBO(I) e IB(I), com I=1, N.

IBM dá a sequência de linhas onde estão localizados os elementos pivô e eliminados e AB, os seus valores, ou seja, IBM definirá K e I nas exp. (III-3) e (III-4), enquanto que AB, A(K, J2) e A(I, J4). Esses vetores têm dimensão LB, soma do número total de pivôs e

elementos eliminados. IBO realiza o endereçamento das posições de - IBM e AB que são usadas em cada passo de eliminação. Para , por exemplo, um passo de eliminação K, o endereço das posições dos vetores - IBM e AB estarão localizados a partir de IBO(K-1) a IBO(K), onde IBO(K-1) é a somatória de todos IBO anteriores a IBO(K). Finalmente , IB acompanha as mudanças de linhas ocorridas na 1ª parte do processamento da SPAMA2 e que também serão realizadas na 2ª parte.

## B - As subrotinas MASP

Como no método de Gauss, foram também desenvolvidas duas subrotinas para a utilização do método da fatorização em soluções de sistemas de equações com matrizes esparsas. A primeira, denominada MASPL, resolve um sistema  $AX=B$  uma única vez. A segunda, denominada MASP2, é composta de duas partes. A primeira parte resolve um sistema  $AX=B1$ , de forma idêntica à MASPL, armazenando todos os valores e posições da matriz L. Para um outro sistema  $AX=B2$ , onde foi somente mudado seu lado direito, é utilizada a segunda parte da MASP2. Agora, as matrizes L e U já foram obtidas, restando resolver os sistemas  $LY= B2$  e  $UX=Y$ .

A forma de utilização dessas subrotinas e suas listagens encontram-se no Apêndice II.

## 1 - Memória de armazenamento

As subrotinas MASP utilizam o armazenamento estático dos coeficientes das matrizes A e U. São utilizados  $A(I,J)$ ,  $IC(I,J)$  e  $INZ(I)$  para armazenar, respectivamente, valores não nulos, as posições nas colunas e o número desses valores em cada linha I, para armazenar A. Em virtude de não haver modificações na distribuição inicial da matriz, as dimensões desses arranjos é conhecida de antemão. Para operar a matriz U são usados  $XU(I,J)$ ,  $ICX(I,J)$  e  $IRZ(I)$ , com XU armazenando seus valores não nulos, ICX a posição de suas colunas e IRZ o número de elementos não nulos para cada linha I. Aquí, a segunda dimensão de XU e ICX não é conhecida inicialmente, sendo defi-

nida através do número máximo de elementos ocorridos em uma linha de U durante o processo.

Os elementos não nulos de L não são totalmente armazenados. Cada linha de L é utilizada uma única vez, não havendo necessidade de guardar esses valores. Assim, utiliza-se um vetor XL(I), onde cada linha de L é formada em sua forma expandida, isto é, com as posições de seus elementos em XL iguais às posições das colunas a qual pertencem.

Finalmente, utilizam-se B(I) para armazenar o vetor B e X(I), os valores das soluções do sistema, além de servir como vetor auxiliar.

## 2 - Operações

Considerando-se as exp. (II-20) a (II-28), vistas no cap. II, que descrevem o processo da fatorização, podemos reescreve-las utilizando agora os arranjos definidos no ítem anterior, obtendo:

- para determinação de L e U,

$$XL(IC(I,1))=A(I,1), \quad I=1,N \quad (\text{III-12})$$

$$XU(1,J)=A(1,J)/XL(1), \quad J=1,INZ(1) \quad (\text{III-13})$$

$$XL(K)=A(I,J1) - \sum_{I1=1}^{K-1} XL(I1).XU(I1,J2), \quad (\text{III-14})$$

com  $I=2,N$ ,  $J1=1,INZ(I)$  tal que  $IC(I,J1)=K$  e  $J2=1,IRZ(I1)$  tal que  $ICX(I1,J2)=K$ ,  $K=2,I$

$$XU(I,J)=1/XL(I). \left( A(I,J1) - \sum_{I1=1}^{K-1} XL(I1).XU(I1,J2) \right), \quad (\text{III-15})$$

com  $I=2,N-1$ ,  $J1=1,INZ(I)$  tal que  $IC(I,J1)=K$  e  $J2=1,IRZ(I1)$  tal que  $ICX(I1,J2)=K$ ,  $K=I+1,N$ .

$$XU(N,1)=1 \quad (\text{III-16})$$

- para a substituição direta,

$$B(1)=B(1)/XL(1) \quad (III-17)$$

$$B(I)=1/XL(I) \cdot (B(I) - \sum_{J=1}^{I-1} XL(J) \cdot B(J)), \quad (III-18)$$

com  $I=2, N$  e  $XL(J) \neq 0$

- para a substituição regressiva,

$$X(N)=B(N) \quad (III-19)$$

$$X(I)=B(I) - \sum_{J=1}^{IRZ(I)} XU(I,J) \cdot X(ICX(I,J)), \quad (III-20)$$

com  $I=N-1, 1$ .

O procedimento de cálculo segue a ordem das equações mostradas acima, determinando-se cada linha I de L e U alternadamente.

No cálculo de cada elemento de uma linha I de L, exp(III-14), situado numa coluna K,  $K=2, I$ , é feita a somatória dos produtos de elementos  $IL$ ,  $IL=1, K-1$ , dessa linha por elementos de U situados na coluna K de linhas  $IL$ . Em virtude de L ser formado em forma expandida sobre XL, o acesso a seus elementos é direto. Por outro lado, a obtenção de cada  $XU(IL, J2)$  acarretará a procura, em cada linha  $IL$  de uma posição  $J2$  cujo XU se encontrará na coluna K. Para contornar o trabalho adicional de computação causado por essas procuras, a matriz U é formada em XU na sua forma transposta, isto é, uma coluna I de U corresponderá a uma linha I de XU. Além do mais, transportando-se para o vetor X a linha I de  $A(I, J)$  em sua forma expandida, o acesso a seus valores também será direto. Sendo assim, a exp.(III-14) assumirá a forma:

$$XL(K)=X(K) - \sum_{J=1}^{IRZ(IL)} XL(ICX(IL, J)) \cdot XU(IL, J) \quad (III-21)$$

com  $I=2, N$ ,  $K=2, I$  e  $IL=1, K-1$ .

Uma vez que U é formada em sua forma transposta, na exp.(III-15) serão obtidas colunas ao invés de linhas de XU. Sendo assim, considerando-se também  $A(I, J1)$  transportado em sua forma expandida -

para X, essa expressão tornar-se-á:

$$XU(K,J)=1/XL(I) \cdot (X(K) - \sum_{J_1=1}^{IRZ(I)} XL(ICX(I_1, J_1)) \cdot XU(I_1, J_1)), \quad (III-22)$$

com  $I=2, N-1$ ,  $K=I+1, N$  e  $I_1=1, I-1$ .

Para cada  $XU(K, J)$  formado corresponderá um  $ICX(K, J)=1$  que determinará a coluna desse  $XU$ . Por outro lado,  $J=IRZ(I)+1$ , onde  $IRZ(I)$  indica o número de não zeros formados na linha  $I$  de  $XU$  em passos anteriores. Por sua vez  $IRZ$  será incrementado, ou seja,  $IRZ(I) = IRZ(I)+1$ .

Uma vez determinada uma linha de  $L$  em  $XL$  é feita a substituição direta obtendo-se o valor  $Y(I)$  da exp. (II-6). Essa substituição esta representada em (III-17) e (III-18), sendo que cada  $Y(I)$  resultante fica armazenado no próprio vetor  $B$ .

Obtidas as matrizes  $L$  e  $U$  é feita a substituição regressiva através das exp. (III-19) e (III-20), que consideram  $XU$  formada em sua forma normal, ou seja, não transposta. Sendo assim, é necessário antes de obter  $X$ , realizar a transformação de  $XU$  para essa forma normal. A transformação é imediata pois, há uma correspondência entre linhas de  $XU$  transposta e colunas de  $XU$  não transposta, e vice-versa.

Observando-se ainda as expressões que definem os coeficientes  $L$  e  $U$ , podemos verificar em (III-21), ou em (III-14), que, devido à dependência na obtenção de um elemento  $XL(K)$  de outros, dessa mesma linha, anteriormente obtidos, o primeiro elemento  $XL(K)$  não nulo para cada linha  $I$  coincidirá com o primeiro não nulo localizado em cada linha correspondente de  $A(I, J)$ , o que é mostrado na exp. (III-12).

### 3 - Condensação Pivotal

Para evitar divisões por zero e para verificar a existência de singularidades no sistema de equações, utiliza-se uma tolerância relativa para comparar com cada valor de  $XL(I)$ , das exp. (III-15) e (III-18), obtido. Esse valor de  $XL$  corresponde ao elemento locali

zado na diagonal principal da matriz L, sendo o equivalente ao pivô definido no método de Gauss.

Essa tolerância é obtida da mesma forma como feito para a subrotina SPAMAT, na exp. (III-7), isto é, através da multiplicação do maior elemento, em módulo, da matriz A inicial por um fator EPS pré-definido.

#### 4 - Ordenação para manter a esparsidade

Da mesma forma como nas subrotinas SPAMAT foram adotadas, para a MASP, duas formas de ordenação, ambas a priori. A primeira forma é a mesma ORDEMI, descrita na SPAMAT no ítem(A-4) deste capítulo, que utiliza a subrotina ORDEM para fazer um arranjo inicial das linhas e colunas da matriz A.

A segunda forma de ordenação é utilizada como uma opção de funcionamento da subrotina, sendo realizada no início de seu processamento, consistindo de um arranjo inicial das linhas da matriz A. Para cada coluna J,  $J=1, N-1$ , as linhas I,  $I=J, N$ , de A que possuem elementos não nulos nessa coluna J, são ordenadas de tal forma que aquela com o menor número de elementos passe a ter a posição J, e assim sucessivamente, restando nas linhas mais inferiores da matriz aquelas que possuírem maior número de elementos. As novas posições das linhas de A, após a ordenação, são armazenadas no vetor IX(I),  $I=1, N$ , pois não são realizadas mudanças efetivas entre essas linhas.

#### 5 - Armazenamento para a subrotina MASP2

A subrotina MASP2 realiza o processo de fatorização de forma idêntica à MASPI. Para obter vantagem no tempo de processamento, a matriz L, da exp.  $LJ=A$ , é totalmente armazenada, quando da utilização da 1ª parte da subrotina. Sendo assim, quando utilizada sua 2ª parte, L e U já foram obtidas, restando somente realizar as operações direta e de substituição regressiva para a solução do sistema.

Além dos arranjos usados pela MASPL, são necessários também os vetores XML(I) e JB(I), com  $I=1, LB$ , que armazenam, respectivamente, os elementos não nulos de L e as colunas às quais pertencem - esses elementos, e o vetor IZL(I), com  $I=1, N$ , onde  $IZL(I) - IZL(I-1)$  dá o número de elementos para cada linha de L. XML e JB possuem a dimensão JB, que corresponde ao número total de elementos não nulos da matriz L.

IV - OS PROGRAMAS DE TESTE

Neste capítulo são descritos os problemas escolhidos para a realização de testes com as subrotinas SPAMAT e MASP.

1 - Cálculo da distribuição de temperatura numa barra unidimensional  
(PT1)

Para uma barra AB com comprimento L, conforme mostra a fig. (IV-1), em que o fluxo de calor é considerado somente na direção x, onde k é a condutividade térmica da barra, constante, TA e TB as temperaturas aplicadas em suas extremidades, com TA maior que TB, e HA e HB, os coeficientes de transmissão de calor dos meios A e B, a

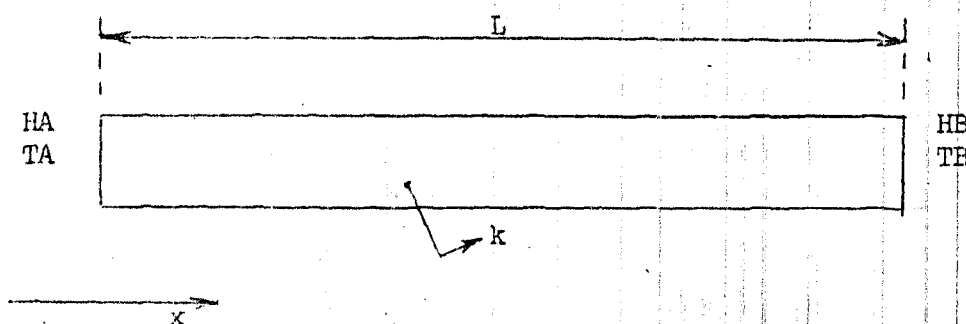


Fig. (IV-1): barra AB com comprimento L.

distribuição de temperatura em seu interior é calculada através da equação de Laplace

$$\frac{\partial^2 T}{\partial x^2} = 0, \quad (IV-1)$$

com as condições de contorno nas extremidades,

$$HA(TA - T) + k \cdot \frac{\partial T}{\partial x} = 0, \text{ e} \quad (IV-2)$$

$$HB(TB - T) + k \cdot \frac{\partial T}{\partial x} = 0. \quad (IV-3)$$

Aplicando-se o método das diferenças finitas /21 /, dividindo-se a barra em (N-1) segmentos, obtém-se o seguinte sistema com N equações:

$$(HA + k/DX)T_1 - k/DX \cdot T_2 = HA \cdot TA \quad (IV-4)$$

$$DX^2 T_{i-1} - 2DX^2 \cdot T_i + DX^2 \cdot T_{i+1} = 0, \quad (IV-5)$$

com  $i=2, N-1$ , e

$$(HB + k/DX)T_N - k/DX \cdot T_{N-1} = HB \cdot TB, \quad (IV-6)$$

onde, nas expressões acima,  $DX=L/(N-1)$

Dessa forma, o sistema de equações diferenciais foi transformado em um sistema de equações algébricas lineares  $AX=B$ , cujo número poderá variar dependendo das divisões aplicadas à barra. Considerando-se, por exemplo,  $N=30$ , a matriz dos coeficientes A terá a forma mostrada na fig. (IV-2), com os espaços em branco representando os elementos de A iguais a zero.

A =

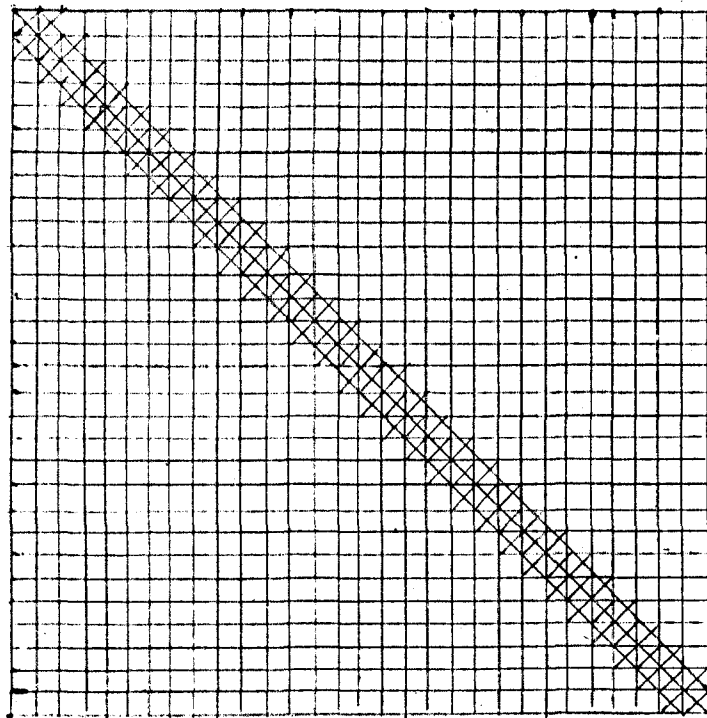


Fig. (IV-2): Configuração da matriz dos coeficientes para o PTL.

A distribuição dos coeficientes de A, qualquer que seja N, será semelhante à mostrada na fig. (IV-2), apresentando a particularidade de ser uma matriz esparsa com uma estrutura de banda bem definida.

As listagens dos dois programas usados, respectivamente, com a subrotina GELB e com as subrotinas SPAMAT e MASP, para cálculos desse sistema, encontram-se no Apêndice IV.

## 2 - Cálculo da distribuição de temperatura numa placa bidimensional

(PT2)

Considerando-se uma placa plana, conforme mostra a fig. (IV-3), em que o fluxo de calor ocorre sómente nas direções x e y ,

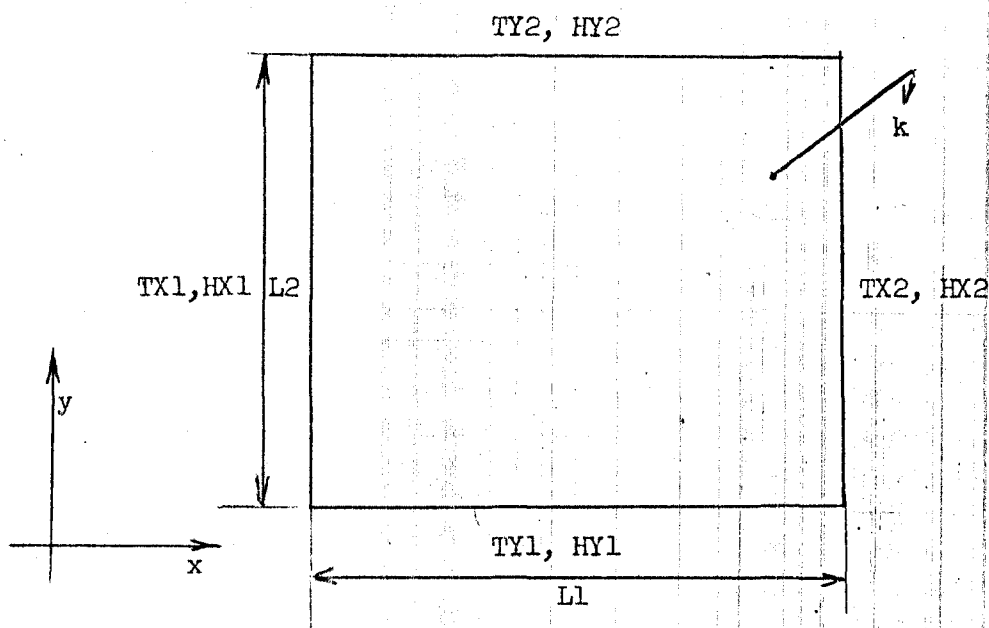


Fig. (IV-3): placa plana com comprimentos  $L_1$  e  $L_2$ .

sendo  $k$  sua condutividade térmica, constante, e,  $T_{X1}, T_{X2}, T_{Y1}, T_{Y2}, H_{X1}, H_{X2}, H_{Y1}$  e  $H_{Y2}$ , as temperaturas e coeficientes de transmissão de calor entre paredes e meios adjacentes à placa, a distribuição de temperatura em seu interior é calculada pela eq. de Laplace

$$\partial^2 T / \partial y^2 + \partial^2 T / \partial x^2 = 0, \quad (\text{IV-7})$$

com as condições de contorno,

$$\text{HX1}(T_{X1} - T) + k \cdot \partial T / \partial x = 0, \quad (\text{IV-8})$$

para a parede vizinha ao meio com temperatura TX1

$$\text{HX2}(T - T_{X2}) + k \cdot \partial T / \partial x = 0 \quad (\text{IV-9})$$

para a parede vizinha ao meio com temperatura TX2

$$\text{HY1}(T - T_{Y1}) + k \cdot \partial T / \partial y = 0, \quad (\text{IV-10})$$

para a parede vizinha ao meio com temperatura TY1

$$\text{HY2}(T - T_{Y2}) + k \cdot \partial T / \partial y = 0, \quad (\text{IV-11})$$

para a parede vizinha ao meio com temperatura TY2.

Dividindo-se a placa em (m-1) segmentos na direção x e, (n-1) segmentos na direção y, num total de N=m.n pontos, e aplicando-se o método das diferenças finitas, baseando-se na configuração da placa mostrada na fig. (IV-4), obtém-se um sistema de equações algébricas lineares:

$$\text{DY}^2 \cdot T_{i,j-1} + \text{DX}^2 \cdot T_{i-1,j} - 2(\text{DX}^2 + \text{DY}^2)T_{i,j} + \text{DX}^2 \cdot T_{i+1,j} + \text{DY}^2 \cdot T_{i,j+1} = 0, \quad (\text{IV-12})$$

com  $i=2, m-1$  e  $j=2, n-1$ , para pontos interiores à placa

$$(\text{HX1} + k/\text{DX})T_{1,j} - k/\text{DX} \cdot T_{2,j} = \text{HX1} \cdot T_{X1}, \quad (\text{IV-13})$$

com  $j=2, n-1$ , para pontos na parede vizinha à região de temp. TX1

$$(\text{HX2} + k/\text{DX})T_{m,j} - k/\text{DX} \cdot T_{m-1,j} = \text{HX2} \cdot T_{X2} \quad (\text{IV-14})$$

com  $j=2, n-1$ , para pontos na parede vizinha à região de temp. TX2

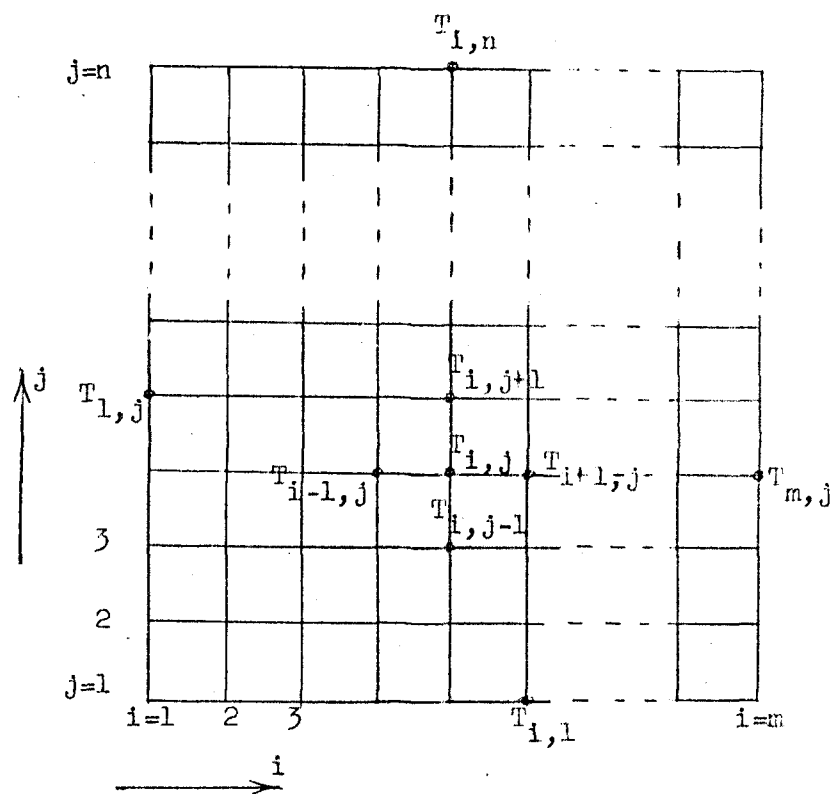
$$(\text{HY1} - k/\text{DY})T_{i,1} + k/\text{DY} \cdot T_{i,2} = \text{HY1} \cdot T_{Y1}, \quad (\text{IV-15})$$

com  $i=1, m$ , para pontos na parede vizinha à região de temp. TY1

$$(\text{HY2} - k/\text{DY})T_{i,n} + k/\text{DY} \cdot T_{i,n-1} = \text{HY2} \cdot T_{Y2}, \quad (\text{IV-16})$$

com  $i=1,m$ , para pontos na parede vizinha à região de temp.  $T_2$ . Sendo que ,

$$DX=L1/(m-1) \text{ e } DY=L2/(n-1).$$



FIG(IV-4): configuração da placa para a aplicação do método das diferenças finitas.

Para esse problema, a enumeração das equações e temperaturas, nos pontos da placa, seguiu a ordem mostrada na fig. (IV-5), para um exemplo com  $m=5$  e  $n=10$ , num total de 50 equações. A matriz  $A$ , dos coeficientes, após a montagem do sistema, com a aplicação das expressões (IV-12) a (IV-16) para  $5 \times 10 = 50$  equações, assumirá a forma mostrada na fig.(IV-6).

Como o ocorrido com o PTL, a matriz  $A$  apresenta uma distribuição de elementos não nulos em estrutura de banda, com uma largura  $F$  dada, por  $(m+n+1)$ .

46	47	48	49	50
41	42	43	44	45
36	37	38	39	40
31	32	33	34	35
26	27	28	29	30
21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

Fig. (IV-5): enumeração das equações e temperaturas para o PT2.

As listagens dos dois programas usados, respectivamente, com a subrotina GELB e com as subrotinas SPAMAT e MASP, para cálculos desse sistema, encontram-se no apêndice IV.

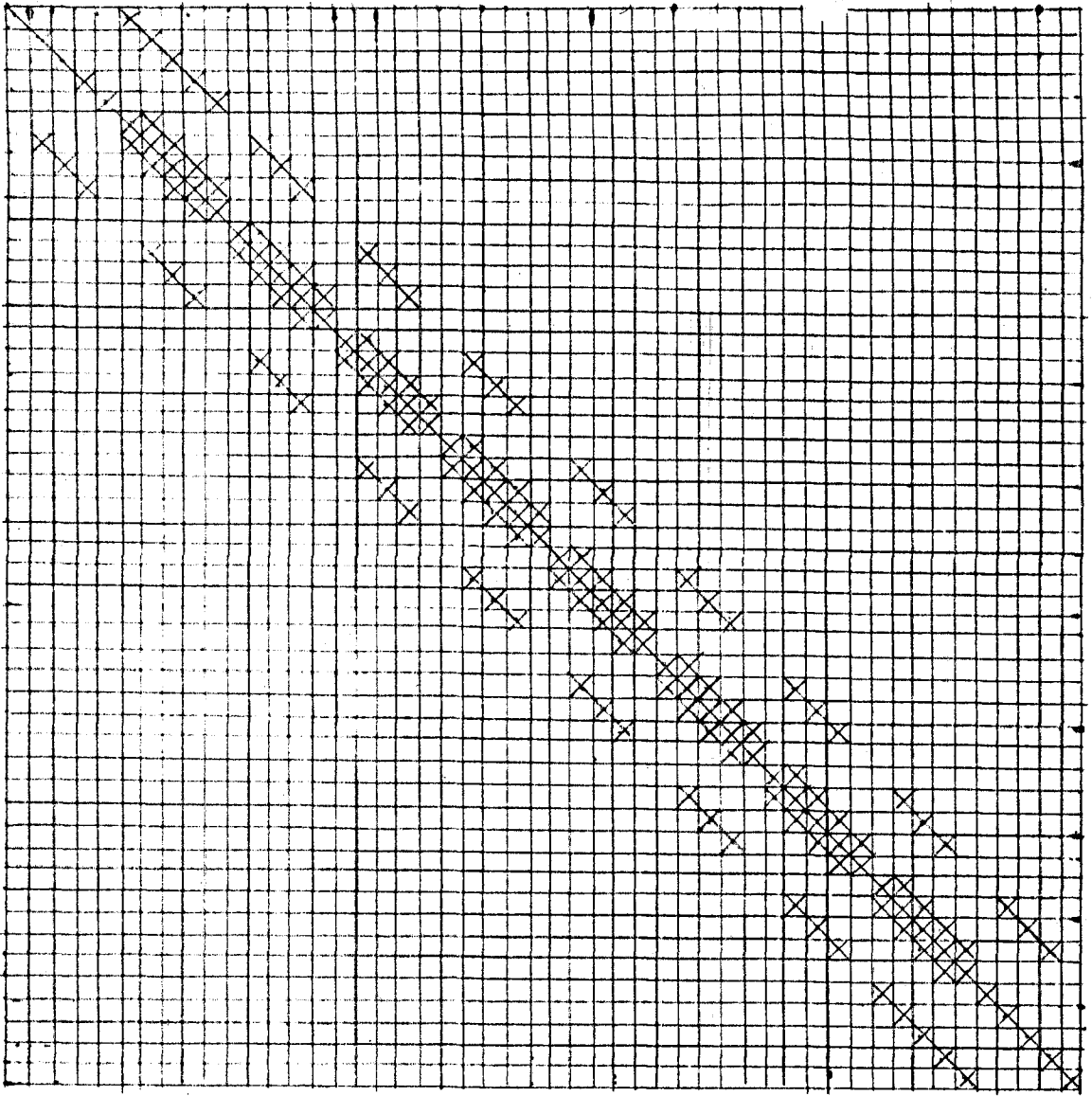


Fig. (IV-6): configuração da matriz A para o programa de teste PI2.

### 3 - Cálculo da distribuição de temperatura numa placa bidimensional (PT3)

Aqui, é calculada a distribuição de temperatura em uma placa da mesma forma como no PT2, o que significa que este problema receberá o mesmo tratamento e terá as mesmas equações diferenciais - que as daquele problema. No entanto, as temperaturas e equações foram enumeradas de forma diversa, primeiro para os pontos externos da placa e depois para os internos, conforme pode ser visto na fig. (IV-7), para um sistema com  $m=5$  e  $n=10$ , num total de 50 equações.

18	17	16	15	14
19	48	49	50	13
20	45	46	47	12
21	42	43	44	11
22	39	40	41	10
23	36	37	38	9
24	33	34	35	8
25	30	31	32	7
26	27	28	29	6
1	2	3	4	5

Fig. (IV-7): enumeração das equações e temperaturas para o programa de teste PT3.

Essa enumeração acarreta em uma configuração da matriz dos coeficientes A, conforme mostra a fig. (IV-8), diferente daquela obtida no PT2, onde podemos notar um sensível aumento da largura de banda F, o que torna a matriz praticamente desprovida dessa característica.

A listagem do programa utilizado com as subrotinas SPAMAT e MASP encontra-se no apêndice IV.

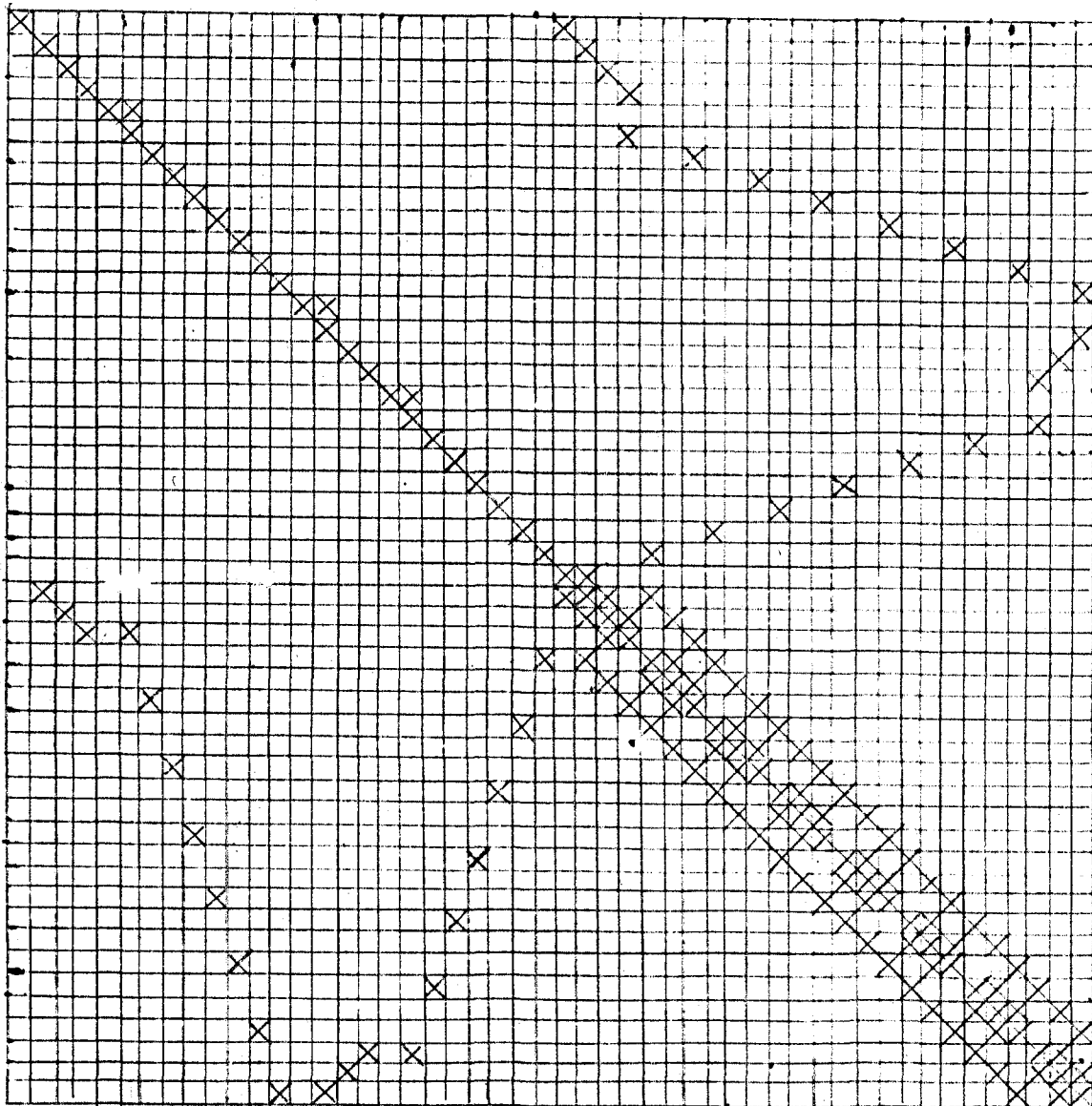


Fig.(IV-8): configuração da matriz A para o programa de teste PT3.

4 - Cálculo da distribuição de temperatura numa placa bidimensional

(PT4)

Nesse problema é realizado o mesmo cálculo descrito pelos PT2 e PT3. No entanto, a enumeração das equações e temperaturas é feita de uma terceira forma, primeiro para os pontos interiores à placa e depois para os exteriores, conforme pode ser visto na fig. (IV-9), para  $m=5$  e  $n=10$ , num total de 50 equações.

42	41	40	39	38
43	22	23	24	37
44	19	20	21	36
45	16	17	18	35
46	13	14	15	34
47	10	11	12	33
48	7	8	9	32
49	4	5	6	31
50	1	2	3	30
25	26	27	28	29

Fig. (IV-9): enumeração das equações e temperaturas para o programa de teste PT4.

A enumeração dos pontos nessa ordem acarreta em uma configuração da matriz A, conforme mostra a fig. (IV-10), diferente da obtida nos dois problemas anteriores. Percebe-se que A perdeu totalmente a característica de banda, tornando-se uma matriz esparsa geral.

A listagem desse programa, utilizado com as subrotinas - MASP e SPAMAT, encontra-se no apêndice IV.

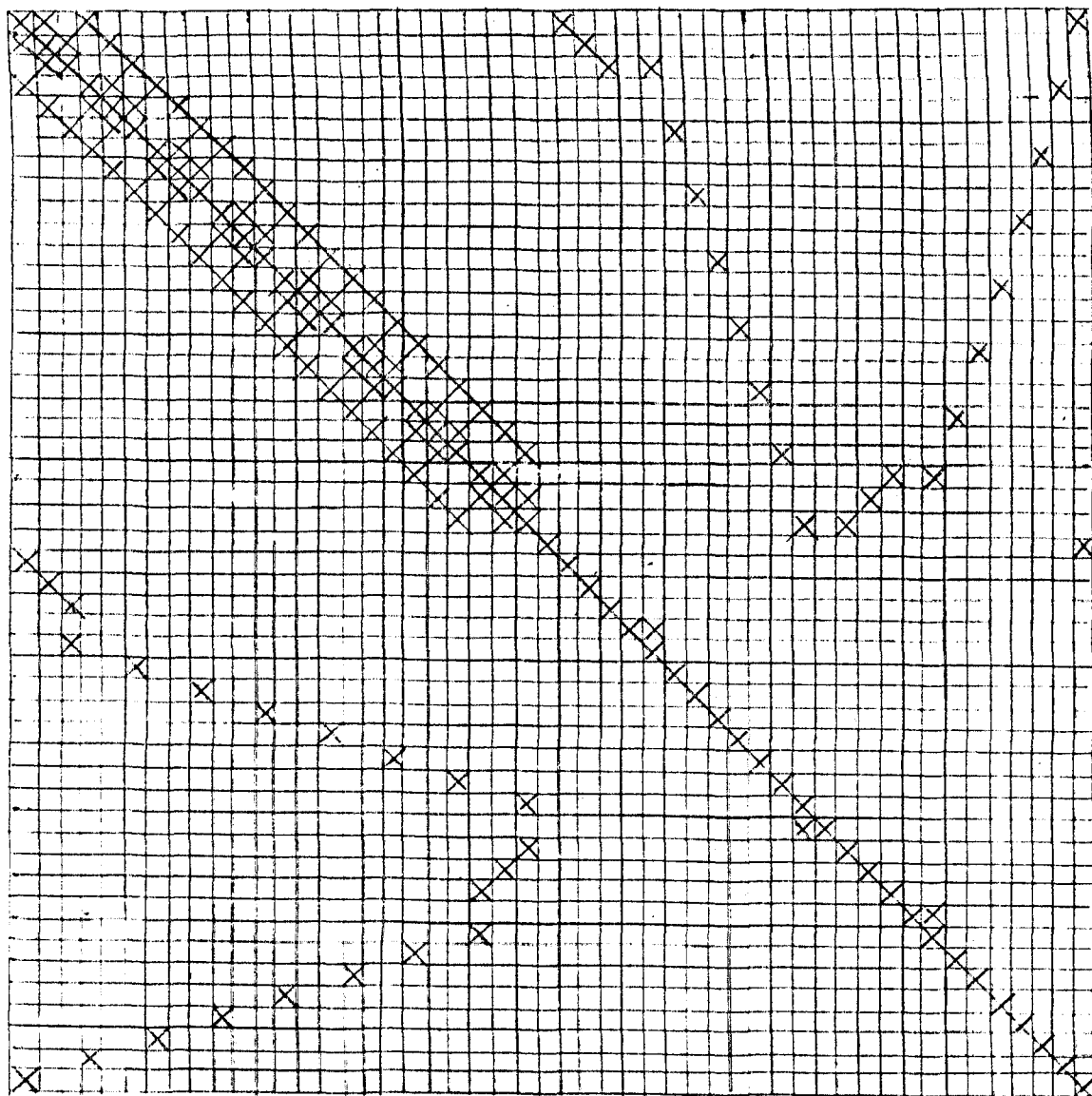


Fig. (IV-10): configuração da matriz A para o programa de teste P14.

5 - Estudo do escoamento do fluido de carcaça de um trocador de calor equipado com chicanas (PT5)

O objetivo desse programa é a obtenção da distribuição de pressão e velocidade em um trecho entre duas chicanas de um trocador de calor de carcaça e tubos com chicanas segmentais / 2 /.

São utilizadas equações algébricas não lineares baseadas em balanços de massa e força, em volumes de controle gerados por divisões axiais (subcanais) e transversais ao trocador. Esse equacionamento é denominado "Análise de sub-canais", e é a base dos códigos para a análise termodinâmica de reatores nucleares.

Na fig. (IV-11), está esquematizada a distribuição dos elementos na matriz dos coeficientes do sistema para um caso com  $N=268$  equações, utilizadas nesse modelo. Como pode ser observado, a matriz A possui uma característica de banda em quase sua totalidade. No entanto, a dispersão dos elementos, ocorrida no final da matriz, faz com que essa característica seja perdida, tornando-se ela uma matriz esparsa geral.

6 - Dinâmica e controle de um gerador de vapor tipo "passo único"

(PT6).

O objetivo desse programa é o cálculo de um modelo, não linear com parâmetros distribuídos, aplicado na dinâmica e controle de um trocador de calor em contra-corrente, projetado como gerador de vapor para uma central nuclear reprodutora. Foi desenvolvido um método implícito e convergente para resolver simultaneamente as equações de conservação de massa, momentum e energia / 13 /.

Na fig. (IV-12), está esquematizada a distribuição dos elementos na matriz dos coeficientes do sistema para um caso com  $N=100$  equações. Pode-se perceber que, nesse problema, a matriz A possui uma configuração de banda bem definida, assemelhando-se a do programa PT1, visto anteriormente.

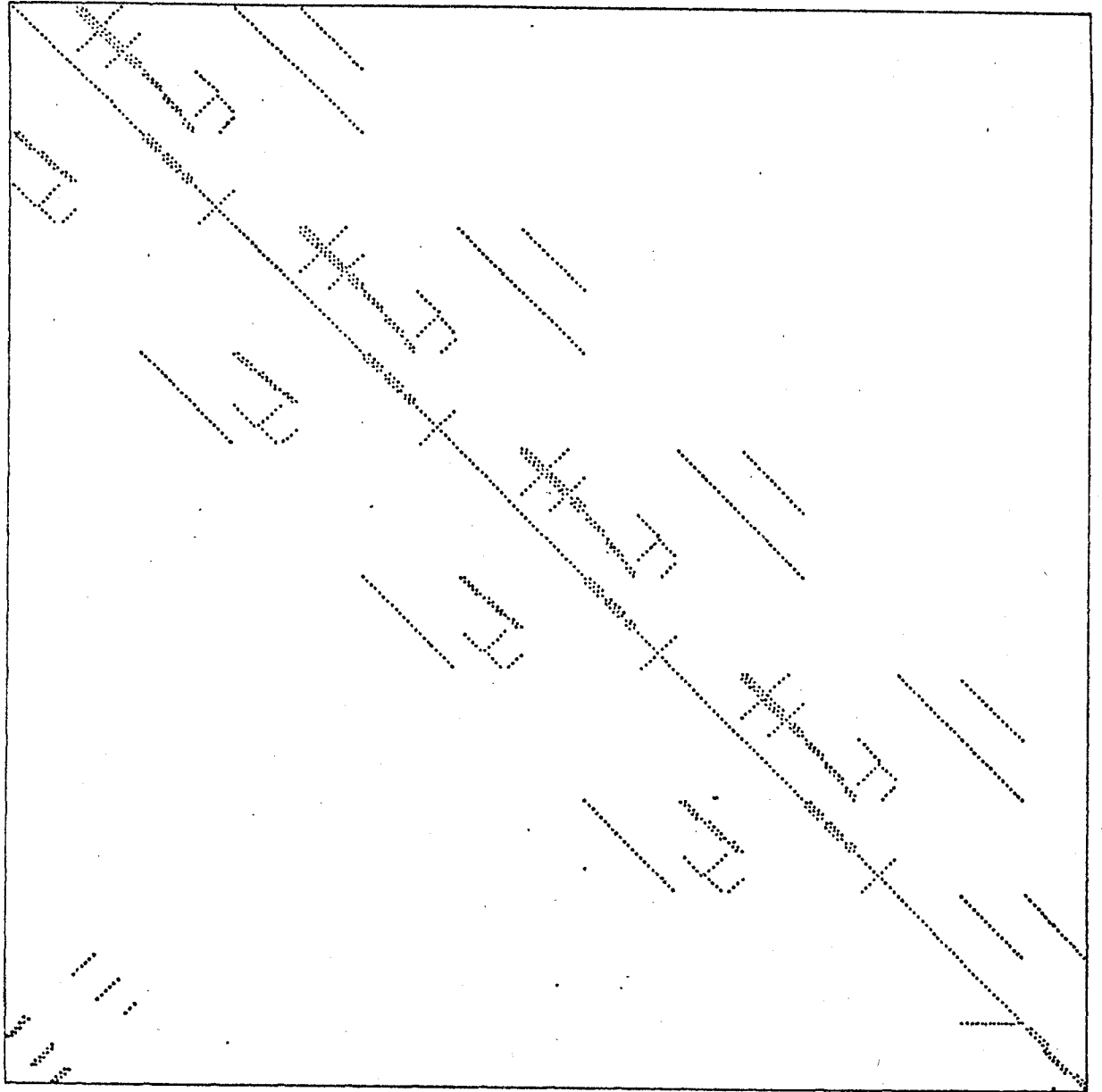


Fig. (IV-11): configuração da matriz A para o programa de teste PT5.

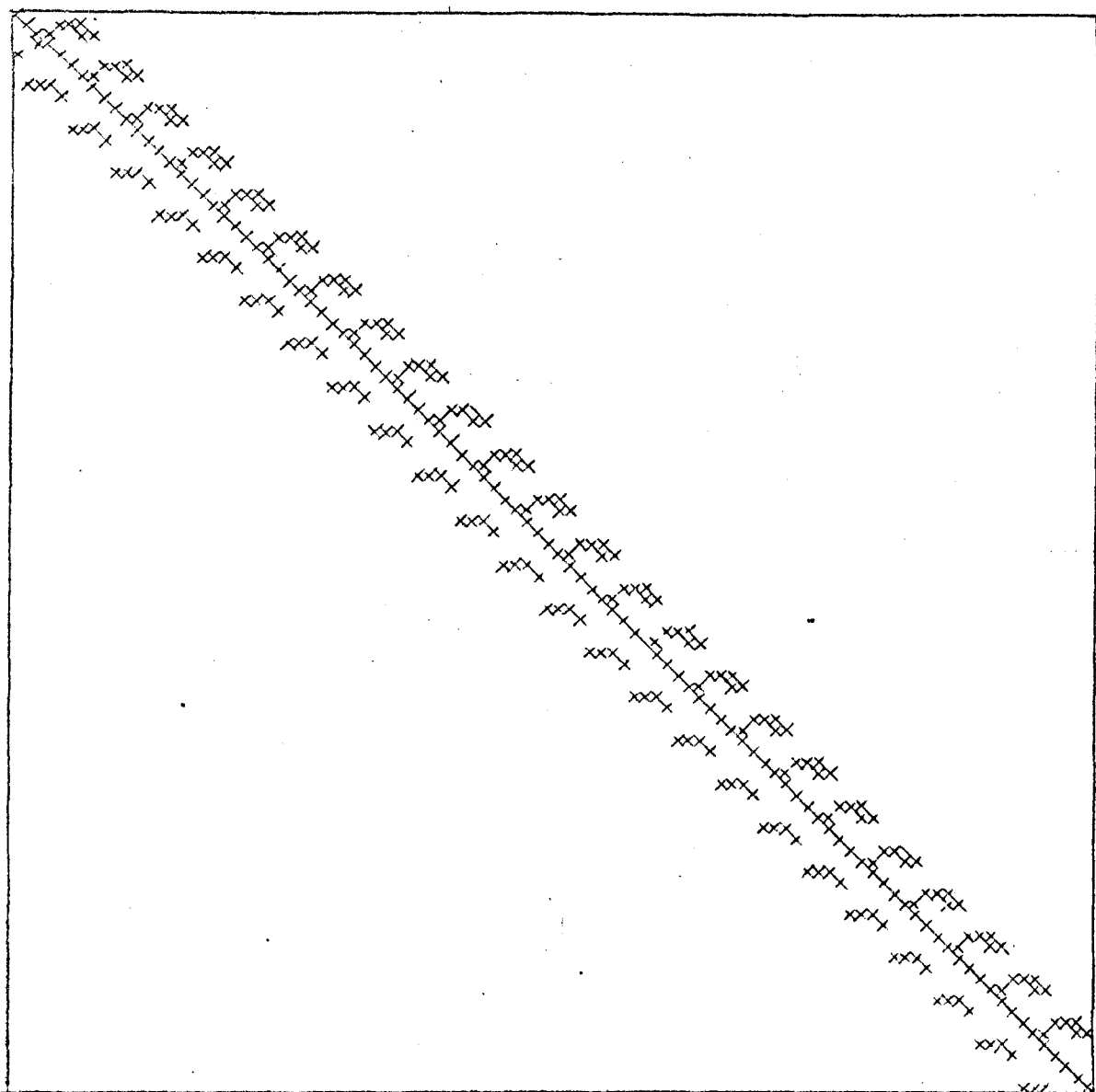


Fig. (IV-12): configuração da matriz A para o programa de teste PT6.

## V - RESULTADOS OBTIDOS E CONCLUSÕES

Neste capítulo são mostrados os resultados obtidos empregando os programas de teste descritos no item anterior. Foram feitos testes com as subrotinas SPAM1 e SPAM2, MAS1 e MAS2. Para efeito de comparação foram empregadas também as subrotinas GELB e SIMQ, ambas constantes da Biblioteca de Subrotinas Científicas (SSP) desenvolvidas pela IBM. A primeira é usada na solução de sistemas de equações algébricas em que a matriz dos coeficientes possui uma estrutura de faixa e, a segunda, é aplicada em sistemas de equações algébricas com matrizes gerais.

Os testes com as subrotinas SPAMAT e MASP englobaram também a utilização de suas opções, para realização de ordenação, opção IS=2 nas subrotinas MASP, e condensação pivotal, IS=1 nas subrotinas SPAMAT. Foram realizados também testes com a subrotina ORDEM, tanto com a SPAM1 como com a MAS1. Um teste adicional foi feito com a SPAM1, utilizando-se a ordenação ORDEM2, citada no cap.III.

### A- Resultados obtidos

Os resultados obtidos, apresentados em forma de tabela, procuram mostrar a performance das subrotinas no que se refere a tempo de processamento, memória de armazenamento utilizada, número de operações realizadas e precisão nos valores das incógnitas para cada sistema resolvido.

Nessas tabelas são apresentados os seguintes valores:

CPU - tempo de execução, em segundos, da subrotina para o teste realizado. A obtenção desse tempo é feita através da subtração do tempo total, que inclui o processamento da subrotina e programa principal, do tempo de execução desse último, obtido através de seu processamento isolado.

Memória - quantidade em kbytes da memória do computador utilizada pelos arranjos definidos nas subrotinas, para cada teste

realizado.

Desvio Máximo Relativo - razão em módulo da máxima diferença obtida entre os lados direito e esquerdo de cada equação, após a substituição dos valores das incógnitas do sistema. É calculado - pela expressão:

$$\text{DRM} = \left| (B(I) - A(I,J) \cdot X(J)) / B(I) \right|_{\max}, \quad (V-1)$$

com  $I=1,N$  e  $J=1,N$ , onde  $A(I,J)$  representa os coeficientes da matriz  $A$ ,  $B(I)$  o lado direito do sistema de equações e  $X(J)$ , os valores das incógnitas do sistema.

NZM - número máximo de elementos ocorridos em uma linha de  $A$  para a SPAMAT ou de  $U$  para a MASP. Indica o efeito da criação de - elementos na eficiência de memória.

NTOT - número total de operações do tipo A-B.C realizadas durante as operações de eliminação ou fatorização. Indica o esforço de computação requerido para a solução do sistema.

KOE - número total de coeficientes de  $A$  que são eliminados durante o processo. Engloba também os elementos criados e que são posteriormente eliminados.

NEC - número total de elementos criados na matriz  $A$  durante o processo de eliminação.

CPU/NTOT - dá o tempo gasto para cada operação básica nos processos de eliminação e da fatorização, indicando a eficiência de - programação no teste realizado. Essas operações englobam, entre outras, operações do tipo A-B.C, procuras, deslocamento de elementos e transferência de posições.

MJD - número total de mudanças entre linhas ocorridas devido a utilização da Condensação Pivotal. No teste utilizando ORDEM2, MJD dá o número de trocas entre linhas para a realização da ordenação.

## 1 - Resultados obtidos com o P11

As tabelas (V-1), (V-2) e (V-3) mostram os resultados obtidos no cálculo da distribuição de temperatura em uma placa unidimensional.

Na tab. (V-1) são apresentadas comparações de performance entre as subrotinas SPAMA1, MASPL e GELB. Nota-se, observando-se as colunas relativas a CPU e memória, que a subrotina GELB apresenta vantagens sobre as outras duas. Isso pode ser justificado considerando-se, conforme mostra a fig. (IV-2), que nesse tipo de sistema a matriz A possui uma distribuição de seus coeficientes numa faixa bem estreita, composta de 3 elementos, o que condiz melhor com a GELB, especial para matrizes com estrutura de faixa. Ao contrário do esperado, a relação CPU/NTOT permaneceu alta em todos os casos, o que mostra a ineficiência de programação para esse problema, embora seja, em termos absolutos, baixo o tempo de processamento das subrotinas. Esses valores altos podem ser explicados considerando-se que o número de operações de soma e multiplicação é baixo, cabendo a maior parcela do tempo de processamento às operações secundárias como procuras e deslocamentos. Finalmente, é importante observar o gradual decréscimo da precisão dos resultados à medida que aumenta o número de equações do sistema, conforme mostra a coluna relativa aos desvios.

Na tab. (V-2) são feitas comparações entre as duas opções de utilização da SPAMA1, com e sem Condensação Pivotal, respectivamente, IS=1 e IS=2. Para todos sistemas calculados, houve um equilíbrio na performance da subrotina nessas opções. Conforme mostra a coluna dos desvios, o uso da Condensação Pivotal não introduziu melhorias na precisão dos cálculos para esses sistemas. Pelo contrário, na maioria dos casos os valores foram maiores do que aqueles utilizados sem a condensação.

Na tab. (V-3) são apresentados resultados da SPAMA2 e GELB. A subrotina GELB pode ser utilizada definindo-se de início mais que um sistema de equações em que a matriz A permanece fixa e o vetor B possui valores distintos para cada sistema. Nessa subrotina entretanto, os valores que definem o lado direito de cada sistema devem ser conheci-

dos de antemão, enquanto que, na SPAMA2, o novo vetor B pode ser redefinido após o cálculo de cada sistema de equações. Nos casos apresentados nessa tabela foram efetuados cálculos com dois sistemas, ou seja, foram definidos dois conjuntos de valores para B. Percebe-se que as diferenças de tempo de processamento entre as execuções da 1ª parte e 2ª parte não são muito pronunciadas. No entanto, a vantagem da utilização dessas subrotinas fica evidenciada se considerarmos um número grande de sistemas resolvidos por sua 2ª parte.

## 2 - Resultados obtidos com o PT2.

Nas tabelas (V-4) a (V-8) são mostrados os resultados obtidos no cálculo da distribuição de temperatura numa placa bidimensional, tendo a matriz dos coeficientes a distribuição de seus elementos conforme mostrado na fig. (IV-6).

Na tab. (V-4) são apresentadas comparações entre as subrotinas SPAMAL (IS=2), MASPL (IS=1) e GELB, para sistemas com número de equações que variam de 100 a 1225. Ao contrário do ocorrido no PTL verifica-se aqui uma melhor performance da subrotina MASPL, principalmente no que se refere ao tempo de processamento. A SPAMAL, a mais lenta das três, foi a que menos memória requereu. Embora a matriz dos coeficientes para esse sistema ainda possua uma estrutura de banda, a subrotina GELB não mostrou a mesma eficiência apresentada no PTL, onde a banda da matriz era mais estreita. Os desvios obtidos, mostrados na coluna desvio máximo relativo, indicam um gradual decréscimo da precisão nos cálculos à medida que aumenta o número de equações do sistema.

Na tab. (V-5) são mostrados resultados utilizando a subrotina SPAMAL com e sem Condensação Pivotal (opções IS=1 e IS=2). Conforme mostram os valores dos desvios, não foi sentida a influência da utilização da condensação, a não ser por pequenas melhorias para sistemas com 900 e 1225 equações. Por outro lado, o trabalho adicional requerido para efetuar essa condensação praticamente não causou diferenças nos tempos de processamento. Pelo contrário, embora insignificante, a opção IS=1 apresentou mais vantagens que a IS=2, isso porque as mudanças entre linhas ocorridas acarretaram em um menor esforço de computação (NTOT), compensando o tempo adicional requerido para a Conden-

sação Pivotal. Esta por sua vez, diminuiu a eficiência do programa, aumentando a relação CPU/NTOT.

Na tab. (V-6) encontram-se os resultados obtidos com a utilização das opções da MASPL sem ordenação, IS=1, e com ordenação, IS=2. A utilização da ordenação nesse programa de teste melhorou ligeiramente a memória de armazenamento requerida, porém o trabalho adicional de computação realizado para essa ordenação acarretou em um aumento no tempo de processamento. Além do mais, percebe-se um pequeno aumento dos desvios com a utilização da opção IS=2, o que mostra um certo antagonismo entre a utilização da ordenação e da condensação pivotal.

Na tab. (V-7) são apresentados resultados obtidos com a SPAMA2, MASP2 e GELB para dois sistemas de equações onde somente é variado o lado direito, vetor B, dos sistemas. Aqui, da mesma forma como o visto na tab. (V-3), a subrotina GELB necessita, de início, de todos os valores que serão utilizados por B para cada sistema. No confronto entre a SPAMA2 e MASP2, embora sem uma vantagem acentuada, a primeira apresentou uma melhor performance em relação à memória de armazenamento, havendo um equilíbrio no tempo de processamento da 2ª parte das subrotinas. Conforme mostra a relação CPU/NTOT para a 2ª parte fica evidenciada a eficiência dessas subrotinas em cálculos de sistemas de equações desse tipo.

Na tab. (V-8) são mostrados resultados da SPAMAL com e sem a utilização da subrotina ORDEM para sistemas com 100, 150 e 600 equações. Para esse tipo de configuração da matriz a utilização da ordenação, efetuada pela ORDEM, não cumpriu o objetivo desejado. Pelo contrário, piorou de forma evidente a performance da SPAMAL.

Os valores de NZM obtidos no cálculo deste programa de teste mostram que, tanto na SPAMAL quanto na MASPL, há, a partir de um determinado passo de eliminação, o preenchimento total de meia faixa da matriz por elementos não nulos e, pelo número de operações realizadas, próximo aquele obtido para a GELB, esse preenchimento, nos sistemas com um número grande de equações, ocorre próximo ao início dos processos. Finalmente, fica evidenciado, na coluna CPU/NTOT, uma eficiência maior da utilização dessas subrotinas nesse tipo de problema,

atingindo essa relação, em alguns casos, valores próximos a 30  $\mu$ seg.

### 3 - Resultados obtidos com os PT3 e PT4

Foram realizados alguns testes empregando as subrotinas - SPAMAL e MASPL para os programas de teste PT3 e PT4, que calculam a - distribuição de temperatura numa placa plana, tendo sido feitas as - montagens dos sistemas de equações de formas diferentes, acarretando distribuições diferentes dos elementos na matriz dos coeficientes, - conforme mostram as figuras (IV-8) e (IV-10). Essas matrizes, embora esparsas, não apresentam uma característica de banda como a encontrada nos PT1 e PT2, o que torna ineficiente a utilização da subrotina - **GELB** nesses problemas. Os resultados obtidos encontram-se nas tabelas (V-9) a (V-12).

Na tab. (V-9) é feita uma comparação da utilização da SPAMAL no PT3 com e sem a utilização da subrotina ORDEM. Os resultados apresentados mostram permanecer inalterada a memória de armazenamento requerida para qualquer das duas opções, não havendo sido introduzida - melhoria alguma. Além do mais, ficou evidenciado, através dos tempos de processamento, o trabalho adicional requerido para a utilização - dessa ordenação.

Na tab. (V-10) são também feitas comparações da SPAMAL com a utilização da subrotina ORDEM, agora no PT4. Ao contrário do ocorrido nos outros programas de teste, os resultados mostram a eficiência na utilização da ordenação realizada pela ORDEM em matrizes com uma distribuição de seus elementos não nulos aleatória. Além da diminuição sensível da memória de armazenamento requerida, houve também uma diminuição do tempo de processamento da SPAMAL, apesar do trabalho de computação adicional realizado para a efetuação dessa ordenação, como pode ser visto na coluna CPU/NTOT, agora superado pelas vantagens adquiridas com a diminuição do esforço de computação (NTOT). Embora de uma forma não muito sensível, a diminuição do número de operações efetuadas acarretou numa maior precisão nos resultados, como é mostrado na coluna dos desvios.

Na tab.(V-11) são mostrados os resultados obtidos com a subrotina MASPL, utilizando as opções IS=1 e IS=2 no programa de teste - PT3. A não ser pelo acréscimo no tempo de computação, a ordenação efetuada com a opção IS=2 em nada influenciou no comportamento de cálculo da MASPL nesse programa.

As duas possíveis formas de ordenação efetuadas pela MASPL, opção IS=2 e utilização da subrotina ORDEM, foram empregadas no PT4, conforme mostrado na tab.(V-12). A exemplo do ocorrido com a SPAMAL nesse programa, as formas de ordenação empregadas melhoraram a performance da subrotina, tanto em memória como em tempo de processamento. Entretanto, observando-se os valores de NZM, bem acima daqueles obtidos com a SPAMAL, conclui-se não ser essa subrotina tão eficiente como aquela, em problemas com a distribuição de elementos na matriz em forma aleatória.

#### 4 - Utilização da ORDEM2 pela subrotina SPAMAL

Os resultados obtidos com a ordenação ORDEM2, descrita no cap. III, item A-4, aplicada à subrotina SPAMAL, são mostrados na tab. (V-13), para os programas PT2, PT3 e PT4. Além da inoperância dessa ordenação nos problemas apresentados, como pode ser constatado através dos valores de NZM, houve problemas de precisão no cálculo, conforme mostra a coluna dos desvios. Para um número pequeno de equações os resultados obtidos foram corretos, o mesmo não ocorrendo para sistemas com um número maior de equações.

#### 5 - Resultados obtidos com o PT5

No sistema de equações utilizado no Estudo do escoamento do fluido de Carcaça de um Trocador de Calor equipado com Chicanas, cuja forma da matriz é mostrada na fig.(IV-11), foram efetuados cálculos com as subrotinas SPAMAL e MASPL, em simples e dupla precisão, e também com a subrotina SIMQ, em simples e dupla precisão, em suas formas normal e modificada, conforme pode ser visto na tab.(V-14). - A SIMQ em sua forma normal realiza os cálculos sem considerar a exis

tência de elementos nulos, isto é, elimina todos  $a_{ij}$ , situados abaixo do pivô, mesmo se  $a_{ij} = 0$ . Na forma modificada foi feita uma mudança - em sua lógica de maneira que as operações de eliminação não são efetuadas sempre que o elemento a ser eliminado,  $a_{ij}$ , for igual a zero. Nesses testes não foi utilizada a GELB pois, no dimensionamento dessa subrotina, deve ser considerada uma faixa com largura  $N$ , número de equações, o que descarta, de imediato, o seu emprego.

Comparando-se os resultados da SIMQ, em suas duas formas, - com a SPAMAL e MASPL, percebe-se um melhor comportamento dessas últimas em matrizes esparsas, tanto em tempo de processamento como em memória utilizada. Mesmo usando-se a forma modificada para a SIMQ, a diferença no CPU é ainda bem acentuada.

No confronto entre a SPAMAL e MASPL, existe vantagem na utilização da segunda pois, embora necessite mais memória, seu tempo de processamento é quase metade do requerido por aquela, o que é importante considerando-se o cálculo de sistemas não lineares como este, - que requerem soluções sucessivas do sistema linearizado. Embora o grau de precisão atingido por todas versões tenha sido bom, observa-se que os desvios obtidos pela SPAMAL, com o uso da Condensação Pivotal (IS=1), foram menores que aqueles sem o uso dessa condensação.

Finalmente, a utilização da ordenação realizada pela subrotina ORDEM não surtiu o efeito desejado. Pelo contrário, aumentou a memória requerida e o tempo de processamento, mesmo diminuindo o número de elementos criados e número de operações de eliminação.

#### 6 - Resultados obtidos com o PT6

A tab. (V-15) apresenta os resultados dos cálculos efetuados pelas subrotinas SPAMAL e MASPL do sistema de equações utilizado no - Estudo da Dinâmica e Controle de um Gerador de Vapor tipo Passo Único, cuja forma da matriz é mostrada na fig. (IV-12). Os valores obtidos - mostram um razoável equilíbrio no comportamento de ambas subrotinas - nesse problema. Os sistemas estudados são não lineares, havendo portanto iterações para a obtenção da solução, conforme mostra a coluna

"num. de iterações". Com relação à memória e tempo de CPU percebe-se uma ligeira vantagem da SPAMAL sobre a MASPL, o que era de se esperar pois a estrutura de banda, estreita, dessa matriz é semelhante à do PTL. Por outro lado, há uma maior precisão nos cálculos obtidos pela MASPL, conforme mostra a coluna dos desvios. Um outro detalhe, que se faz presente nesse problema são as diferenças de desvio obtidas pela SPAMAL com e sem a utilização da Condensação Pivotal. Da mesma forma como o ocorrido no PTL, são grandes os valores de CPU/NTOT.

#### 7 - Teste com a subrotina SIMQ

Na tab.(V-16) são mostrados os testes realizados com a subrotina SIMQ nos programas PTL,PT2,PT3 e PT4. Como era de se esperar, os requisitos de memória e de tempo de processamento são bem maiores que os necessários para as subrotinas empregando "técnicas com matrizes esparsas". É importante observar, nesses resultados, o valor praticamente constante, em torno de 30 microsegundos, da relação CPU/NTOT. É razoável, pois, nessa subrotina, o tempo de processamento depende exclusivamente do número de operações realizadas, NTOT, não havendo utilização das operações requeridas devido ao armazenamento secundário, citado no cap.II, ítem B-1.

Tendo em vista o exposto podemos, de antemão, para qualquer sistema com N equações, prever os requisitos de memória e de tempo de processamento necessários para a utilização da SIMQ, dados por:

$$\text{memória} = 4N^2/1024 \quad |\text{kby}|, \quad (\text{V-2})$$

$$\text{CPU} = 30.10^{-6}.NTOT \quad |\text{seg}|, \quad (\text{V-3})$$

$$\text{com } NTOT = (N^3 + 3N^2 - N)/3.$$

## B - Conclusões

Nos testes realizados neste trabalho, procurou-se utilizar matrizes esparsas cuja distribuição de seus elementos não nulos assumisse várias formas, abrangendo matrizes em forma de banda estreita, como nos PT1 e PT6, com banda larga, como no PT2, com banda larga de maneira a ocupar praticamente toda a largura da matriz, como nos PT3 e PT5, e, com matrizes esparsas gerais, como no PT4, onde já não é distinguida uma banda. Esses testes evidenciaram ser difícil definir um método de solução de sistemas com matrizes esparsas capaz de satisfazer plenamente cada um dos problemas onde venha a ser aplicado.

De maneira geral, qualquer subrotina susceptível de ser utilizada em um determinado sistema de equações com matrizes esparsas, empregando técnicas aplicadas a matrizes esparsas para sua solução, apresenta vantagens acentuadas sobre a não utilização dessas técnicas na solução do sistema. Isso pode ser confirmado comparando-se, em alguns testes realizados, os resultados apresentados pela SIMQ com os das outras subrotinas. Isso fica mais evidenciado à medida que aumenta o número de equações do sistema. Se considerarmos, por exemplo, um sistema  $20 \times 30$ , igual a 600 equações, visto no PT2, e aplicarmos as equações (V-2) e (V-3) obteríamos, aproximadamente, 1400 kbytes e 36 minutos como memória e tempo de processamento necessários para sua solução pela SIMQ, enquanto que, da ordem de 90 kbytes e 9 segundos, seriam suficientes para a solução do mesmo sistema através de qualquer uma das outras subrotinas aqui apresentadas.

O tempo de processamento e a memória utilizada depende, em subrotinas para a solução de equações gerais, exclusivamente do número de equações do sistema. Para subrotinas aplicadas à solução de sistemas com matrizes esparsas essa dependência fica ligada também à distribuição dos coeficientes não nulos da matriz, pois essa distribuição é que comandará a "criação de elementos" durante o processo, que, por sua vez, interferirá no número total de operações, de eliminação ou fatorização, realizadas, e na memória utilizada.

As ordenações, cujo objetivo é evitar a "criação de elementos", propostas neste trabalho, feitas através da subrotina ORDEM, ou

através da opção IS=2 para a MASP, apresentaram resultados satisfatórios quando aplicadas à matrizes esparsas com uma distribuição de seus elementos de uma forma aleatória, como a apresentada no PT4. No entanto, considerando-se os resultados obtidos nos PT2, PT3 e PT4, - utilizados para a análise de um mesmo problema, pode-se afirmar que, sempre que possível, é mais vantajoso procurar uma forma de enumeração das incógnitas e equações do sistema capaz de evitar ao máximo - essa criação, sem dispor da ordenação da matriz.

Finalizando, tendo em vista os resultados apresentados podemos definir, para cada tipo de matriz esparsa, uma subrotina que melhor performance apresente:

- a subrotina GELB, em matrizes com distribuição de elementos não nulos numa banda estreita.

- a subrotina MASP, em matrizes com distribuição de seus elementos em banda larga.

- a subrotina SPAMAT, em matrizes esparsas gerais.

Equações	Subrotina	CPU (seg)	Memória (kby)	Desvio Relativo Máximo	NZM	NTOT	NEC	KOE	CPU/NTOT (useg)
100	SPAMAL	0,12	3,32	$0,89 \times 10^{-7}$	2	198	0	99	606
	MASPL	0,11	4,10	$0,89 \times 10^{-7}$	1	198	-	-	555
	GELB	0,07	1,56	$0,48 \times 10^{-5}$	-	198	-	-	353
200	SPAMAL	0,17	6,64	$0,12 \times 10^{-5}$	2	398	0	199	427
	MASPL	0,25	8,20	$0,12 \times 10^{-5}$	1	398	-	-	628
	GELB	0,17	3,12	$0,95 \times 10^{-5}$	-	398	-	-	427
400	SPAMAL	0,38	13,28	$0,49 \times 10^{-5}$	2	798	0	399	476
	MASPL	0,63	16,41	$0,49 \times 10^{-5}$	1	798	-	-	789
	GELB	0,36	6,25	$0,97 \times 10^{-5}$	-	798	-	-	451
800	SPAMAL	0,71	26,56	$0,19 \times 10^{-4}$	2	1598	0	799	444
	MASPL	1,80	32,81	$0,19 \times 10^{-4}$	1	1598	-	-	1126
	GELB	0,68	12,50	$0,14 \times 10^{-4}$	-	1598	-	-	425
1200	SPAMAL	1,22	39,84	$0,15 \times 10^{-4}$	2	2398	0	1199	509
	MASPL	3,51	49,22	$0,15 \times 10^{-4}$	1	2398	-	-	1464
	GELB	1,15	18,75	$0,63 \times 10^{-4}$	-	2398	-	-	479
2000	SPAMAL	2,10	66,41	$0,78 \times 10^{-4}$	2	3998	0	1999	525
	MASPL	7,02	82,03	$0,78 \times 10^{-4}$	1	3998	-	-	1756
	GELB	1,70	31,25	$0,15 \times 10^{-3}$	-	3998	-	-	425

Tab. (V-1): Resultados das subrotinas SPAMAL (IS=2), MASPL (IS=2) e GELB para o programa de teste PT1.

Equações	Opção	CPU (seg)	Memória (kby)	Desvio Máximo Relativo	NZM	NTOT	NEC	KOE	CPU/NTOT (useg)
100	IS=1	0,13	3,32	$0,48 \times 10^{-5}$	2	198	0	99	656
	IS=2	0,12	3,32	$0,89 \times 10^{-5}$	2	198	0	99	606
200	IS=1	0,17	6,64	$0,49 \times 10^{-5}$	2	398	0	199	427
	IS=2	0,17	6,64	$0,12 \times 10^{-5}$	2	398	0	199	427
400	IS=1	0,36	13,26	$0,98 \times 10^{-5}$	2	798	0	399	451
	IS=2	0,38	13,28	$0,49 \times 10^{-5}$	2	798	0	399	476
800	IS=1	0,88	26,56	$0,24 \times 10^{-4}$	2	1598	0	799	551
	IS=2	0,71	26,56	$0,19 \times 10^{-4}$	2	1598	0	799	444
1200	IS=1	1,33	39,84	$0,63 \times 10^{-4}$	2	2398	0	1199	555
	IS=2	1,22	39,84	$0,15 \times 10^{-4}$	2	2398	0	1199	509
2000	IS=1	2,23	66,41	$0,19 \times 10^{-4}$	2	3998	0	1999	558
	IS=2	2,10	66,41	$0,78 \times 10^{-4}$	2	3998	0	1999	525

Tab.(V-2): resultados da sub. SPAM1, nas opções IS=1 e IS=2, para o prog. de teste P11, utilizando  $EPS=10^{-12}$ .

Equações	Subrotina	Memória (kby)	CPU (seg)		NTOT	CPU/NTOT (useg)	
			1ª parte	2ª parte		1ª parte	2ª parte
100	SPAMA2	4,87	0,12	0,05	198	606	252
	GELB	1,95	0,07	0,04	198	354	202
200	SPAMA2	9,36	0,17	0,10	398	427	251
	GELB	3,90	0,17	0,10	398	427	251
400	SPAMA2	18,74	0,39	0,22	798	314	275
	GELB	7,81	0,36	0,22	798	302	275
800	SPAMA2	37,48	0,72	0,45	1598	450	282
	GELB	15,63	0,68	0,48	1598	425	300
1200	SPAMA2	51,55	1,23	0,75	2398	513	313
	GELB	23,44	1,15	0,83	2398	479	346
2000	SPAMA2	93,74	2,11	1,25	3998	528	313
	GELB	39,06	1,70	1,21	3998	425	303

Fig. (V-3): resultados das subrotinas SPAMA2 (IS=2) e GELB para o prog. de teste PTL.

Equações	Subrotina	Memória (kby)	CPU (seg)	Desvio Relativo Máximo	NZM	NTOT	NEC	KOE	CPU/NTOT (useg)
10x10	SPAM1	8,01	0,26	$0,52 \times 10^{-4}$	11	4439	951	626	59
	MASPL	10,55	0,15	$0,18 \times 10^{-4}$	10	5391	-	-	31
	GELB	8,38	0,37	$0,18 \times 10^{-3}$	-	9000	-	-	41
10x15	SPAM1	12,01	0,48	$0,95 \times 10^{-5}$	11	7209	1521	996	67
	MASPL	15,82	0,30	$0,95 \times 10^{-5}$	10	8731	-	-	34
	GELB	12,67	0,50	$0,61 \times 10^{-4}$	-	13500	-	-	37
20x30	SPAM1	83,20	9,14	$0,12 \times 10^{-4}$	21	164774	17281	9706	56
	MASPL	98,44	6,80	$0,44 \times 10^{-5}$	20	182056	-	-	37
	GELB	97,62	8,34	$0,14 \times 10^{-4}$	-	228000	-	-	36
30x20	SPAM1	118,36	20,15	$0,20 \times 10^{-4}$	31	392389	27391	14766	51
	MASPL	130,08	13,29	$0,55 \times 10^{-5}$	30	419781	-	-	32
	GELB	143,50	17,57	$0,24 \times 10^{-4}$	-	522000	-	-	34
30x30	SPAM1	177,54	31,53	$0,20 \times 10^{-4}$	31	613329	42531	22906	51
	MASPL	200,39	21,78	$0,59 \times 10^{-5}$	30	655861	-	-	33
	GELB	216,15	27,25	$0,19 \times 10^{-4}$	-	783000	-	-	35
35x35	SPAM1	277,54	59,30	$0,24 \times 10^{-4}$	36	1182239	69926	37226	50
	MASPL	308,64	40,80	$0,65 \times 10^{-5}$	35	1252166	-	-	33
	GELB	342,07	50,90	$0,13 \times 10^{-4}$	-	1457750	-	-	35

Fig. (V-4): resultados das sub. SPAM1 (IS=2), MASPL (IS=2) e GELB para o prog de teste PT2.

Equações	Opção	Memoria (kby)	CPU (seg)	Desvio Máximo Relativo	NZM	MUD	NTOT	KOE	NEC	CPU/NTOT (useg)
10x10	IS=1	8,01	0,22	$0,84 \times 10^{-11}$	11	17	3436	611	610	64
	IS=2	8,01	0,26	$0,52 \times 10^{-4}$	11	0	4439	626	951	59
10x15	IS=1	12,01	0,45	$0,56 \times 10^{-4}$	11	22	5886	981	1340	76
	IS=2	12,01	0,48	$0,95 \times 10^{-5}$	11	0	7209	996	1521	67
20x30	IS=1	83,20	8,82	$0,16 \times 10^{-4}$	21	47	150006	9671	16420	59
	IS=2	83,20	9,14	$0,12 \times 10^{-4}$	21	0	164774	9706	17281	55
30x20	IS=1	118,36	19,51	$0,19 \times 10^{-4}$	31	47	356676	14711	26050	55
	IS=2	118,36	20,15	$0,20 \times 10^{-4}$	31	0	392389	14766	37391	51
30x30	IS=1	177,54	30,97	$0,18 \times 10^{-4}$	31	57	569776	22851	40910	54
	IS=2	177,54	31,53	$0,20 \times 10^{-4}$	31	0	613329	22906	42531	51
35x35	IS=1	277,54	54,43	$0,13 \times 10^{-4}$	36	67	1110861	37161	67685	49
	IS=2	277,54	59,30	$0,24 \times 10^{-4}$	36	0	1182239	37226	69926	50

Tab. (V-5): resultados da sub. SPAM1, nas opções IS=1 e IS=2, para o prog. de teste PT2.

Equações	Opção	Memória (kby)	CPU (seg)	Desvio Maximo Relativo	NLM	ITOT	CPU/ITOT (useg)
10x10	IS=1	10,55	0,15	$0,16 \times 10^{-4}$	10	5391	31
	IS=2	9,38	0,19	$0,70 \times 10^{-4}$	9	4343	44
10x15	IS=1	15,82	0,30	$0,95 \times 10^{-5}$	10	8731	34
	IS=2	14,94	0,42	$0,53 \times 10^{-4}$	9	7323	57
20x30	IS=1	98,44	6,80	$0,44 \times 10^{-5}$	20	182056	37
	IS=2	94,92	8,84	$0,13 \times 10^{-4}$	19	166918	53
30x20	IS=1	133,59	13,29	$0,55 \times 10^{-5}$	30	419781	32
	IS=2	130,07	16,32	$0,19 \times 10^{-4}$	29	383913	42
30x30	IS=1	200,39	22,29	$0,56 \times 10^{-5}$	30	655861	34
	IS=2	195,12	26,88	$0,13 \times 10^{-4}$	29	611873	44
35x35	IS=1	308,64	40,88	$0,68 \times 10^{-5}$	35	1252166	33
	IS=2	301,46	51,70	$0,15 \times 10^{-4}$	34	1180193	44

Tab. (V-6): resultados da sub. MASPl, nas opções IS=1 e IS=2, para o prog. de teste PT2.

Equações	Subrotina	Memória (kby)	CPU(seg)		NTOT	CPU/NTOT(useg)	
			1ª parte	2ª parte		1ª parte	2ª parte
10x10	SPAMA2	11,74	0,26	0,07	4439	59	16
	MASP2	14,99	0,15	0,07	5391	31	13
	GELB	8,77	0,37	0,07	9000	41	8
10x15	SPAMA2	19,30	0,48	0,08	7209	67	11
	MASP2	22,82	0,30	0,14	8731	34	16
	GELB	13,26	0,50	0,10	13500	37	8
20x30	SPAMA2	145,92	9,14	1,47	164774	55	9
	MASP2	159,99	6,80	0,91	182056	37	5
	GELB	99,96	8,31	1,59	228000	37	7
30x20	SPAMA2	210,73	20,15	1,77	392389	51	4,5
	MASP2	221,28	13,29	1,80	419781	32	4,4
	GELB	145,84	17,57	2,19	522000	34	4,2
30x30	SPAMA2	320,54	31,53	3,33	613320	51	5,3
	MASP2	341,63	21,78	1,90	655861	33	2,9
	GELB	219,66	27,25	3,50	783000	35	4,5
35x35	SPAMA2	407,62	59,30	4,14	1182239	50	3,5
	MASP2	536,32	40,88	4,72	1252166	33	3,8
	GELB	346,85	50,90	4,64	1457750	35	3,2

Tab .(V-7): resultados das sub. SPAMA2(IS=2), MASP2(IS=2) e GELB para o prog. de teste PT2.

Equações	Opção	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	MZM	NTOT	NEC	KOE	CPU/NTOT (useg)
10x10	s/ORDEM	8,01	0,26	$0,520 \times 10^{-4}$	11	4439	951	626	59
10x10	c/ORDEM	13,87	0,89	$0,207 \times 10^{-3}$	21	9284	1200	744	36
10x15	s/ORDEM	12,01	0,48	$0,950 \times 10^{-5}$	11	7209	1521	996	67
10x15	c/ORDEM	25,19	2,11	$0,820 \times 10^{-4}$	26	17992	2244	1351	117
20x30	s/ORDEM	83,20	9,09	$0,120 \times 10^{-4}$	21	164774	17281	9706	55
20x30	c/ORDEM	209,76	62,55	$0,789 \times 10^{-4}$	57	542062	30384	16246	115

Tab. (V-8): resultados da sub.SPAMAL, com e sem a utilização da sub.ORDEM, para o prog. de teste PT2.

Equações	Opção	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	NZM	MTOT	NEC	KOE	CPU/MTOT (useg)
10x10	s/ORDEM	7,42	0,30	$0,52 \times 10^{-4}$	10	3313	687	488	90
10x10	c/ORDEM	7,42	0,63	$0,244 \times 10^{-3}$	10	3312	686	487	190
20x30	s/ORDEM	79,68	11,59	$0,110 \times 10^{-4}$	20	149193	15607	8858	77
20x30	c/ORDEM	79,68	18,98	$0,110 \times 10^{-4}$	20	149192	15606	8857	127
35x35	s/ORDEM	270,36	63,4	$0,240 \times 10^{-4}$	35	1108713	65537	35013	57
35x35	c/ORDEM	270,36	99,4	$0,274 \times 10^{-3}$	35	1108712	65536	35012	89

Tab. (V-9): resultados da sub.SPAMAL, com e sem utilização da sub.ORDEM, para o prog. de teste PT3.

Equações	Opção	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	NZM	NTOT	NEC	KOE	CPU/NTOT (useg)
10x10	s/ORDEM	20,90	2,85	$0,353 \times 10^{-4}$	33	43451	3568	1940	66
10x10	c/ORDEM	13,87	1,00	$0,231 \times 10^{-4}$	21	8899	1178	743	149
15x10	s/ORDEM	40,14	8,07	$0,694 \times 10^{-4}$	43	127326	7653	4125	63
15x10	c/ORDEM	30,47	3,75	$0,671 \times 10^{-4}$	32	43142	3597	2136	87
20x30	s/ORDEM	336,33	178	$0,140 \times 10^{-3}$	93	2330431	68893	35655	76
20x30	c/ORDEM	192,19	67	$0,101 \times 10^{-3}$	52	610937	32525	17974	110

Tab. (V-10): resultados da sub.SPAM1 (IS=2), com e sem a utilização da sub.ORDEM, para o prog. de teste PT<sup>4</sup>.

Equações	Opção	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	NTOT	NZM	CPU/NTOT (useg)
20x30	IS=1	79,69	18,23	$0,109 \times 10^{-4}$	164800	20	111
20x30	IS=2	79,69	20,76	$0,109 \times 10^{-4}$	164800	20	126
35x35	IS=1	270,36	85,4	$0,351 \times 10^{-5}$	1174250	35	73
35x35	IS=2	270,36	104,4	$0,351 \times 10^{-5}$	1174250	35	89

Tab. (V-11): resultados da subrotina MASPI, com as opções IS=1 e IS=2, para o programa de teste PT3.

Equações	Opção	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	NTOT	NZM	CPU/NTOT (useg)
10x10	IS=1	54,88	1,91	$0,496 \times 10^{-5}$	47019	87	41
10x10	IS=2	43,75	1,14	$0,173 \times 10^{-3}$	21234	68	66
10x10	c/ORDEM	25,39	1,16	$0,391 \times 10^{-4}$	15676	44	74
10x15	IS=1	127,44	5,16	$0,190 \times 10^{-5}$	115734	137	45
10x15	IS=2	101,95	3,38	$0,426 \times 10^{-3}$	60689	108	56
10x15	c/ORDEM	71,19	2,69	$0,526 \times 10^{-4}$	42614	73	63

Tab. (V-12): resultados da sub.MASPl, empregando as opções IS=1, IS=2 e a sub. ORDEM, para o prog. de teste PT4.

Programa de teste	Equações	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	NZM	MJD	NTOT	NEC	KOE	CPU/NTOT (useg)
PT2	10x10	12,11	0,25	$0,148 \times 10^5$	18	71	1869	597	610	133
	5x10	2,54	0,09	$0,285 \times 10^{-3}$	6	12	278	92	124	323
PT3	10x10	10,93	0,34	$0,830 \times 10^4$	16	54	1573	418	452	216
	5x10	1,81	0,11	$0,597 \times 10^{-3}$	5	3	237	54	88	464
PT4	10x10	16,79	0,60	$0,834 \times 10^5$	26	89	2758	875	816	217
	5x10	4,00	0,14	0,130	11	38	405	171	195	346
	5x6	1,87	0,08	$0,975 \times 10^{-2}$	8	21	197	88	106	406

Tab. (V-13): resultados da sub.SPAM1 (IS=2) aplicando a ordenação ORDEM2, para os prog. de teste PT2, PT3 e PT4.

Caso	Subrotina	Opção	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	NZM	NTOT	NEC	KOE	CPU/NTOT (useg)
simples precisão	SPAM1	IS=1	98,41	23,54	$0,366 \times 10^{-6}$	60	257593	14268	9531	21
	SPAM1	IS=2	98,41	22,13	$0,189 \times 10^{-4}$	60	218859	14645	10817	101
	MASPL	IS=1	114,63	8,95	$0,254 \times 10^{-4}$	54	222547	-	-	40
	SIMQ modificada	-	281,61	39,14	$0,152 \times 10^{-6}$	-	1238515	-	-	32
dupla precisão	SPAM1	IS=1	163,31	24,42	$0,434 \times 10^{-7}$	60	246726	14008	9430	99
	SPAM1	IS=2	163,31	23,85	$0,451 \times 10^{-5}$	60	218859	14645	10817	109
	SPAM1 c/ORDEMI	IS=2	273,23	25,32	$0,663 \times 10^{-5}$	102	318995	12891	7115	79
	MASPL	IS=1	193,67	12,49	$0,673 \times 10^{-5}$	54	222547	-	-	56
	SIMQ	-	563,22	208	$0,146 \times 10^{-7}$	-	6380410	-	-	33
	SIMQ modificada	-	563,22	42,37	$0,162 \times 10^{-7}$	-	1216264	-	-	35

Tab. (V-14): resultados obtidos com as sub.SPAM1, MASPL e SIMQ, em simples e dupla precisão, para o prog. de teste PT5.

Equações	Subrotina	Opção	Memória (Kby)	CSU (seg)	Desvio Máximo Relativo	Número de ite- rações	NZM	NTOT	NEC	ROE	MUD	CPU/NTOT (useg)
100	SPAM1	IS=1	4,49	3,31	$0,58 \times 10^{-3}$	18	3	357	97	144	72	515
		IS=2	4,49	3,01	$0,18 \times 10^{-1}$	18	3	329	47	144	0	508
	MASPI	IS=1	5,86	3,14	$0,29 \times 10^{-3}$	18	2	240	-	-	-	727
200	SPAM1	IS=1	8,20	6,35	$0,69 \times 10^{-3}$	19	3	732	197	294	147	456
		IS=2	8,20	6,03	$0,41 \times 10^{-1}$	19	3	679	97	294	0	467
	MASPI	IS=1	10,55	6,71	$0,44 \times 10^{-3}$	19	2	490	-	-	-	721
300	SPAM1	IS=1	13,48	10,01	$0,75 \times 10^{-3}$	19	3	1107	297	444	222	476
		IS=2	13,48	8,42	0,23	19	3	1029	147	444	0	431
	MASPI	IS=1	15,82	10,32	$0,64 \times 10^{-3}$	19	2	740	-	-	-	734

Tab.(V-15): resultados das sub.SPAM1, IS=1 e IS=2, e MASPI, IS=1, para o prog. de teste PT6

Programa de teste	Equações	Memória (kby)	CPU (seg)	Desvio Máximo Relativo	NTOT	CPU/NTOT (useg)
PT1	100	39,45	10,24	$0,49 \times 10^{-5}$	328350	31,2
	200	157,03	61,32	$0,49 \times 10^{-5}$	2646700	30,7
PT2	10x10	39,45	10,11	$0,84 \times 10^{-4}$	328350	30,8
	10x20	157,03	60,52	$0,53 \times 10^{-4}$	2646700	30,4
PT3	10x10	39,45	10,13	$0,62 \times 10^{-4}$	328350	30,8
	10x20	157,03	78,34	$0,95 \times 10^{-5}$	2646700	29,6
PT4	10x10	39,45	10,40	$0,15 \times 10^{-3}$	328350	31,6
	10x20	157,03	80,36	$0,51 \times 10^{-3}$	2646700	30,4

Tab. (V-16): resultado dasub. SIMQ nos prog. de teste PT1, PT2, PT3 e PT4.

REFERÊNCIAS BIBLIOGRÁFICAS

- /1/ - ALWAY, G.G. & MARTIN, D.W. An algorithm for reducing the bandwidth of a matrix of symmetric configuration. Computer J., 8: 264-72, 1965.
- /2/ - BAPTISTA Fº, B.D. Modelo numérico para solução termohidráulica de um trocador de calor de carcaça e tubos 'U' com chicanas - segmentais. São Paulo, 1979. (Dissertação de mestrado IPEN)
- /3/ - BERRY, R.D. An optimal ordering of electronic circuit equations for a sparse matrix solution. IEEE Trans. Circuit Theory, CT-18: 139-45, 1971.
- /4/ - BRANDON Jr., D.M. The implementation and use of sparse matrix techniques in general simulation programs. Computer J., 17: 165-70, 1974.
- /5/ - BUNCH, J.R. Block methods for solving sparse linear systems. - In: BUNCH, J.R. Sparse Matrix Computations. New York, Academic, 1976. p.39-58.
- /6/ - CURTIS, A.R. & REID, J.K. Fortran subroutines for the solution of sparse sets of linear equations. London, England, 1971. - (AERE-R.6844).
- /7/ - CURTIS, A.R. & REID, J.K. The solution of large sparse unsymmetric systems of linear equations. J. Inst. Math. Appl., 8: 344-53, 1971.
- /8/ - DUFF, I.S. A survey of sparse matrix research. Proceedings of the IEEE, 65(4): 500-37, 1977.
- /9/ - LARCOMBE, M.H.E. A list processing approach to the solutions of large sparse sets of matrix equations and the factorisation of the overall matrix. In: REID, J.K. Large sparse sets of linear equations. London, Academic, 1971. p.25-40.

- /10/ - ECKMANN, B. & DOLD, A. Sparse matrix techniques. Berlin, Springer-Verlag, edited by V.A. Barker, 1977.
- /11/ - GEORGE, J.A. Block elimination on finite element systems of equations. Proc. conf. at IBM Research Center. New York, Sept/1971. p.101-14.
- /12/ - GEORGE, J.A. On block elimination for sparse linear systems. SIAM J. Numer. Anal., 11: 585-603, 1974.
- /13/ - GOMES, A.V. Dinâmica e controle de um gerador de vapor do tipo passo único para uma central nuclear reprodutora MSBR. São Paulo, 1979. (Dissertação de mestrado IPEN)
- /14/ - HSIEH, H.Y. & GHAUSI, M.S. A probabilistic approach to optimal pivoting and prediction of fill-in for random sparse matrices. IEEE Trans. Circuit Theory, CT-19: 329-36, 1972.
- /15/ - HSIEH, H.Y. & GHAUSI, M.S. On optimal pivoting algorithms in sparse matrices. IEEE Trans. Circuit Theory, CT-19: 93-96, 1972.
- /16/ - HSIEH, H.Y. Pivoting-order computation method for large random sparse systems. IEEE Trans. Circuits and Syst., CAS-21: 225 - 30, 1974.
- /17/ - IBM Application Program GH 20-0205-4. Scientific Subroutine - Package, 1968, p. 137-40.
- /18/ - MARKOWITZ, H.M. The elimination form of the inverse and its applications to linear programming. Management Sci., 3: 265-69, 1957.
- /19/ - NAKHLA, M. et al. An optimal pivoting order for the solution of sparse systems of equations. IEEE Trans. Circuits and Systems, CAS-21: 222-25, 1974.
- /20/ - ORCHARD-HAYS, W. Advanced Linear Programming computing techniques. New York, McGraw-Hill, 1968.

- /21/ - RALSTON, A. A first course in Numerical Analysis. Tokyo, Tosho Printing Co., 1975.
- /22/ - REID, J.K. A note on the stability of Gaussian elimination. - J. Inst. Math. Appl., 8: 374-75, 1971.
- /23/ - REID, J.K. Large sparse sets of linear equations. London, Academic, 1976.
- /24/ - ROSE, J.D. & BUNCH, J.R. Sparse matrix computation. New York, Academic, 1976.
- /25/ - ROSE, J.D. & WILLOUGHBY, R.A. Sparse matrices and their applications. London, Plenum, 1972.
- /26/ - SANTOS, VITORIANO RUAS DE BARROS. Curso de Cálculo Numérico. Rio de Janeiro, 3ª edição, Livros Técnicos e Científicos Editora S.A., 1977.
- /27/ - TEWARSON, R.P. Sparse Matrices. New York, Academic, 1973.
- /28/ - TEWARSON, R.P. The Crout reduction for sparse matrices. Comp. J., 12, 1969.
- /29/ - TARJAN, R. On the product form of inverses of sparse matrices and graph theory. SIAM Rev., 9: 91-99, 1967.
- /30/ - TARJAN, R. Solution of a system of simultaneous linear equations with a sparse coefficient matrix by elimination methods. BIT, 7: 226-39, 1967.
- /31/ - WELTER, A.N.V. Modelo numérico de isolamento térmica interna tipo po fibras em dutos de gás quente. São Paulo, 1979. (Dissertação de mestrado IPEN)
- /32/ - WILKINSON, J.H. The algébric eigenvalue problem. Oxford, Oxford Univ. Press, 1965.

APÊNDICE I

listagem das subrotinas:

- SPAMA1
- SPAMA2

```

C SUBROTINA SPAMAI
C OBJETIVO
C SOLUCAO DE UM SISTEMA DE EQ. ALGEBRICAS AX=B, ATRA-
C VES DO METODO DE ELIM. DE GAUSS, APLICANDO TECNICAS
C PARA MATRIZES ESPARSAS
C
C USO
C CALL SPAMAI(N, EPS, IS, NZM, IF)
C DIMENSIONAMENTO
C INTEGER*2 IC, INZ, IX, IY, IZ
C COMMON/G1/ A(N,NZM), B(N), X(N), IC(N,NZM), INZ(N), IX(N),
C IY(N), IZ(N)
C DESCRICAO DOS PARAMETROS
C N - NUM. DE EQUACOES DO SISTEMA
C EPS - CONST. DE ENTRADA USADA NO CALC. DA TOLERAN-
C CIA RELATIVA(TOL) P/ TESTE DE PERDA DE SIGNIFICANCIA
C IS - OPCOES DE USO DA SUBR. C/ CONDENSACAO PIVOTAL,
C IS=1, OU COM TOLERANCIA RELATIVA, IS=2
C A SAIDA IS=-1 SIGNIFICA TER SIDO UTILIZADO UM PIVO
C CUJO VALOR E MENOR QUE TOL
C NZM - NUM MAX DE ELEMENTOS FORMADOS EM LINHAS DE A,
C E IC, USADO PARA DEFINIR A SEGUNDA DIMENSAO DESSES
C ARRANJOS. INICIALMENTE DEVE SER ASSUMIDO UM VALOR
C PARA ESSA DIMENSAO
C QDO. OS COEF. INICIAIS DA MATRIZ A NAO ESTAO EM CR-
C DEM CRESCENTE P/ COL., DA-SE COMO ENTRADA NZM=0
C IF - PARA MATRIZES COM ESTRUTURA DE FAIXA, IF=LARGURA
C DA FAIXA INFERIOR. PARA MATRIZES GERAIS, OU PARA USO
C CGM A SUBROTINA ORDEM, IF=N
C ARRANJOS UTILIZADOS
C A(N,NZM) - ARMAZENA OS VAL. NAO NULOS DE A
C B(N) - ARMAZENA O LADO DIREITO DO SISTEMA
C X(N) - SAIDA CGM A SOLUCAO DAS EQUACOES
C IC(N,NZM) - ARMAZENA A COLUNA DOS ELEM. NAO NULOS DE A
C INZ(N) - DA O NUM DE ELEM EM CADA LINHA DE A(N,NZM)
C IX(N), IY(N), IZ(N) - VETORES AUXILIARES
C SUBROTINAS REQUERIDAS
C CALL ORDEM(N) - QDO. SE DESEJA UM ARRANJO INICIAL
C DAS LINHAS E COL. DE A P/ DIMINUIR A MEMORIA REQUE-
C RIDA. SEU USO E OPCIONAL. DEVE SER DEFINIDO COMO EN-
C TRADA NZM=0
C USO EM DUPLA PRECISAO
C DOUBLE PRECISION A,B,X,AUX,TOL,AM
C
C *****
C SUBROUTINE SPAMAI(N, EPS, IS, NZM, IF)
C INTEGER*2 IC, INZ, IX, IY, IZ
C COMMON /G1/A(600,21), B(600), X(600), IC(600,21), INZ(600),
C IX(600), IY(600), IZ(600)
C DO 1 I=1, N
C IY(I)=I
C IZ(I)=I
C CALL ORDEM(N)
C IF(NZM.EQ.1)GO TO 205
C ARRANJO DOS ELEM. DE A EM ORDEM CRESCENTE
C DO 200 I=1, N
C J1=1
C IK=1
C NZ=INZ(I)
C IF(NZ.EQ.1)GO TO 200
C 204 DO 201 J=J1, N
C DO 202 K1=IK, NZ
C IF(IC(I, K1).EQ.J)GO TO 203
C 202 CONTINUE

```

```

201 CCNTINUE
203 IAUX=IC(I,IK)
   IC(I,IK)=IC(I,K1)
   IC(I,K1)=IAUX
   AUX=A(I,IK)
   A(I,IK)=A(I,K1)
   A(I,K1)=AUX
   IK=IK+1
   IF(IK.GE.NZ)GO TO 200
   J1=J+1
   GO TO 204
200 CONTINUE
   NZM=1

```

C DET. CA TCL DO SISTEMA

```

205 AM=A(1,1)
   DO 206 I=1,N
   NZ=INZ(I)
   DO 206 K=1,NZ
   IF(ABS(AM).GE.ABS(A(I,K)))GO TO 206
   AM=A(I,K)
206 CCNTINUE
   TOL=EPS*ABS(AM)

```

C

```

IF(IS.FQ.2)GO TO 2
N1=N-1
DO 10 I2=1,N1
I=I2(I2)
L=1
IX(1)=0
AM=0.
N2=I2+IF
IF(N2.GT.N)N2=N
DO 11 J1=I2,N2
J=I2(J1)
IF(IC(J,1).NE.I2)GO TO 11
L=L+1
IX(1)=IX(1)+1
IX(L)=J
IF(ABS(A(J,1)).LE.ABS(AM))GO TO 11
AM=A(J,1)
NJ=J

```

```

11 CCNTINUE
IF(ABS(AM).GT.TOL)GO TO 12
GO TO 101

```

```

2 N1=N-1
DO 20 I2=1,N1
I=I2(I2)
L=1
IX(1)=0
N2=I+IF
IF(N2.GT.N)N2=N
DO 21 J1=I2,N2
J=I2(J1)
IF(IC(J,1).NE.I2)GO TO 21
L=L+1
IX(1)=IX(1)+1
IX(L)=J

```

```

21 CONTINUE
NJ=IX(2)
IF(ABS(A(NJ,1)).GT.TOL)GO TO 12

```

```

22 NZ=IX(1)+1
DO 23 K=2,NZ
NJ=IX(K)

```

```

      IF( ABS(A(NJ,1)) -GT. TOL) GO TO 15
23  CCNTINUE
      NJ=IX(2)
101  IS=-1
      IF(IX(1).EQ.0) GO TO 150
      12  IF(NJ.EQ.1) GO TO 14
C  MUDANCA DE LINHAS
15  CCNTINUE
      MUD=MUD+1
      DO 16 K=12,N
      IF(IZ(K).EQ.NJ) GO TO 17
16  CONTINUE
17  IZ(K)=I
      IZ(I2)=NJ
      J=IX(1)+1
      DO 18 K=2, J
      IF(IX(K).EQ.NJ) GO TO 19
18  CCNTINUE
19  IX(K)=IX(2)
      IX(2)=NJ
      I=NJ
14  A1=A(I,1)
      A(I,1)=0.
      B(I)=B(I)/A1
      NZ=INZ(I)
      DO 24 K=1,NZ
24  A(I,K)=A(I,K)/A1
C  OPERACOES DE ELIMINACAO
C
      NM=IX(1)+1
      IF(NM.GE.3) GO TO 25
      IF(IS.EQ.2) GO TO 20
      GO TO 10
25  DO 40 L=3,NM
      I1=IX(L)
      A2=A(I1,1)
      A(I1,1)=0
      KCE=KOE+1
      NZ=INZ(I)
      IF(NZ.GT.1) GO TO 57
C  DESLOCAMENTO DE ELEMENTOS NUMA LINHA
      IF(INZ(I1).EQ.1) GO TO 56
      K3=INZ(I1)-1
      DO 303 K4=1,K3
      IC(I1,K4)=IC(I1,K4+1)
      A(I1,K4)=A(I1,K4+1)
303  CCNTINUE
      INZ(I1)=INZ(I1)-1
      GO TO 56
57  NN=IC(I,NZ)
      NZ1=INZ(I1)
      IF(IC(I1,NZ1).GT.NN) NN=IC(I1,NZ1)
      DO 50 K=12,NN
50  X(K)=0.
      IF(NZ1.EQ.1) GO TO 55
      DO 51 K1=2,NZ1
      C=IC(I1,K1)
      X(K)=A(I1,K1)
51  CCNTINUE
55  DO 52 K1=2,NZ
      K=IC(I1,K1)
      X(K)=X(K)-A2*A(I,K1)
52  CONTINUE

```

```

K1=0
INZ(I1)=INZ(I1)-1
DO 53 K=I2,NN
IF(X(K).EQ.0)GO TO 53
K1=K1+1
A(I1,K1)=X(K)
IC(I1,K1)=K
53 CCNTINUE
KOC=KOC+(K1-INZ(I1))
KCN=KCN+NZ-(K1-INZ(I1))
INZ(I1)=K1
56 B(I1)=B(I1)-A2*B(I)
IF(INZ(I1).GT.NZM)NZM=INZ(I1)
40 CONTINUE
IF(I5.EQ.1)GO TO 10
20 CONTINUE
GO TO 33
10 CONTINUE

```

C  
C SUBSTITUICAO REGRESSIVA

```

33 J=IZ(N)
NZ=INZ(J)
DO 30 K=1,NZ
IF(IC(J,K).NE.C)GO TO 32
30 CCNTINUE
GO TO 101
32 B(J)=B(J)/A(J,K)
DO 90 J=2,N
IB=N-J+1
J1=IZ(IB)
NZ=INZ(J1)
IF(IC(J1,NZ).EQ.C)GO TO 90
DO 90 K=1,NZ
IA=IC(J1,K)
I1=IZ(IA)
B(J1)=B(J1)-A(J1,K)*B(I1)
90 CONTINUE
DO 94 I=1,N
J=IY(I)
K=IZ(J)
94 X(I)=B(K)
150 RETURN
END

```

```

C
C SUBROTINA SPAMAZ
C OBJETIVO
C SOLUCAO DE SISTEMAS DE EQUACOES ALGEBRICAS AX=B, -
C COM A FIXO E B VARIAVEL, ATRAVES DO METODO DE ELIMI-
C NACAO DE GAUSS, APLICANDO TECNICAS P/ MATRIZES ES-
C PARSAS
C
C USO
C CALL SPAMAZ(N, EPS, NZM, IS, IF, LB)
C
C DIMENSIONAMENTO
C INTEGER*2 IC, INZ, IBO, IBM, IB, IX, IY, IZ
C COMMON /G1/ A(N, NZM), B(N), X(N), IC(N, NZM), INZ(N), IX(N),
C IY(N), IZ(N)
C COMMON /G2/ AB(LB), IBO(N), IBM(LB), IB(N)
C
C DESCRICAO DOS PARAMETROS
C N - NUM DE EQUACOES DO SISTEMA
C EPS - CONSTANTE DE ENTRADA USADA NO CALCULO DA TOLE-
C RANCIA RELATIVA(TOL) P/ TESTES DE PERDA DE SIGNIFI-
C CANCIA
C NZM - NUM MAX DE ELEMENTOS FORMADOS EM LINHAS DE A,
C E IC, USADO P/ DEFINIR A SEGUNDA DIMENSAO DESSES -
C ARRANJOS. INICIALMENTE DEVE SER ASSUMIDO UM VALOR
C PARA ESSAS DIMENSÕES
C QUANDO OS COEF. INICIAIS DE A NAO ESTIVEREM EM OR-
C DEM CRESCENTE P/ COLUNA, DA-SE COMO ENTRADA NZM=0
C IF - P/ MATRIZES COM ESTRUTURA DE FAIXA, IF=LARGU-
C RA DA FAIXA INFERIOR. PARA MATRIZES GERAIS, OU P/
C USO C/ A SUB. ORDEM, IF=N
C IS - OPCOES DE USO DA SUBROTINA C/ CONDENSACAO PIVO-
C TAL, IS=1, OU COM TOL. RELATIVA, IS=2
C P/ IS=3, E USADA A SEGUNDA PARTE DA SUBROTINA
C A SAIDA IS=-1 SIGNIFICA TER SIDO UTILIZADO UM PIVO
C CUJO VALOR E MENOR QUE TOL
C LB - COMO SAIDA, INDICA A DIMENSÃO NECESSARIA P/
C UTILIZACAO DOS ARRANJOS AB E IBM
C
C ARRANJOS UTILIZADOS
C A(N, NZM) - ARMAZENA OS VALORES NAO NULOS DE A
C B(N) - ARMAZENA O LADO DIREITO DO SISTEMA
C X(N) - SAIDA C/ A SOLUCAO DAS EQUACOES
C IC(N, NZM) - ARMAZENA A COLUNA DOS ELEM NAO NULOS DE A
C INZ(N) - DA O NUM MAX DE ELEMENTOS EM CADA LINHA DE
C A(N, NZM)
C AB(LB) - ARMAZENA OS PIVOS E ELEM/ ELIMINADOS DU-
C RANTE A TRIANGULARIZACAO EFETUADA NA PRIM PARTE DA
C SUBROTINA
C IBM(LB) - ARMAZENA AS POSICOES NAS COL DE A DOS ELE-
C MENTOS CORRESPONDENTES EM AB
C IB(N), IBO(N) - VETORES USADOS NO CONTROLE DAS OPERA-
C COES NA SEGUNDA PARTE DA SUBROTINA
C IX(N), IY(N), IZ(N) - VETORES AUXILIARES
C
C SUBROTINAS REQUERIDAS
C CALL ORDEM(N) - QUANDO SE DESEJA UM ARRANJO INICI-
C AL DAS LINHAS E COLUNAS DE A P/ DIMINUIR A MEMORIA
C REQUERIDA. SEU USO E OPCIONAL. DEVE SER DEFINIDO CO-
C MO ENTRADA NZM=0. E USADA SOM/ COM A PRIMEIRA PAR-
C TE DA SUBROTINA
C
C USO EM DUPLA PRECISAO
C DOUBLE PRECISION A, B, X, AB, AUX, AM, TOL, A1, A2
C *****
C SUBROUTINE SPAMAZ(N, EPS, NZM, IS, IF, LB)
C INTEGER*2 IC, INZ, IBO, IBM, IB, IX, IY, IZ
C COMMON /G1/ A(2000, 2), B(2000), X(2000), IC(200, 3), INZ(2000),
C IX(2000), IY(2000), IZ(2000)

```

```

CCMMCN/G2/ AB(5000),IB(2000),IBM(5000),IB(2000)
IF(15.EQ.3)GO TO 1000
DO 1 I=1,N
  IB(I)=1
  IBC(I)=0
  IY(I)=1
1  IZ(I)=1
  LB=0
  CALL CRDEM(N)
  IF(N/M.EQ.1)GO TO 205
C ARRANJO DOS ELEM. DE A EM ORDEM CRESCENTE
DO 200 I=1,N
  J1=1
  IK=1
  NZ=INZ(I)
  IF(NZ.EQ.1)GO TO 200
204 DO 201 J=J1,N
  DO 202 K1=IK,NZ
  IF(IC(I,K1).EQ.J)GO TO 203
  202 CCNTINUE
  201 CCNTINUE
  203 IAUX=IC(I,IK)
  IC(I,IK)=IC(I,K1)
  IC(I,K1)=IAUX
  AUX=A(I,IK)
  A(I,IK)=A(I,K1)
  A(I,K1)=AUX
  IK=IK+1
  IF(IK-GE.NZ)GO TO 200
  J1=J+1
  GO TO 204
200 CCNTINUE
  NZM=1
C DET. DA TCL DO SISTEMA
205 AM=A(1,1)
DO 206 I=1,N
  NZ=INZ(I)
DO 206 K=1,NZ
  IF(ABS(AM).GE.ABS(A(I,K)))GO TO 206
  AM=A(I,K)
206 CCNTINUE
  TOL=EPS*ABS(AM)
C
  IF(15.EQ.2)GO TO 2
  N1=N-1
  DO 10 I2=1,N1
  I=IZ(I2)
  L=1
  IX(I)=0
  AM=0.
  N2=I+1
  IF(N2.GT.N)N2=N
  DO 11 J1=I2,N2
  J=I2(J1)
  IF(IC(J,1).NE.I2)GO TO 11
  L=L+1
  IX(I)=IX(I)+1
  IX(L)=J
  IF(ABS(A(J,1)).LE.ABS(AM))GO TO 11
  AM=A(J,1)
  NJ=J
  11 CCNTINUE
  IF(ABS(AM).GT.TOL)GO TO 12

```

```

GO TO 101
2 N1=N-1
CO 20 I2=1,N1
I=I2(I2)
L=1
IX(1)=0
N2=1+IF
IF(N2.GT.N)N2=N
DO 21 J1=I2,N2
J=I2(J1)
IF(IC(J,1).NE.I2)GO TO 21
L=L+1
IX(1)=IX(1)+1
IX(L)=J
21 CONTINUE
NJ=IX(2)
IF(ABS(A(NJ,1)).GT.TOL)GO TO 12
NZ=IX(1)+1
CO 23 K=2,NZ
NJ=IX(K)
IF(ABS(A(NJ,1)).GT.TOL)GO TO 15
23 CONTINUE
NJ=IX(2)
101 IS=-1
IF(IX(1).EQ.0)GO TO 150
12 IF(NJ.EQ.1)GO TO 14
C MUDANCA DE LINHAS
15 NZ=INZ(I)
IF(INZ(NJ).GT.NZ)NZ=INZ(NJ)
DO 13 K=1,NZ
AUX=A(I,K)
A(I,K)=A(NJ,K)
A(NJ,K)=AUX
IAUX=IC(I,K)
IC(I,K)=IC(NJ,K)
IC(NJ,K)=IAUX
13 CONTINUE
IAUX=INZ(I)
INZ(I)=INZ(NJ)
INZ(NJ)=IAUX
IB(I)=NJ
AUX=B(I)
B(I)=B(NJ)
B(NJ)=AUX
14 A1=A(I,1)
A(I,1)=0.
LB=LB+1
AB(LB)=A1
IBQ(LB)=IBQ(I)+1
IPM(LB)=I
B(I)=B(I)/A1
NZ=INZ(I)
DO 24 K=1,NZ
24 A(I,K)=A(I,K)/A1
C OPERACOES DE ELIMINACAO
C
AM=IX(1)+1
IF(AM.GE.3)GO TO 25
IF(IS.EQ.2)GO TO 20
GO TO 10
25 DO 40 L=3,AM
I1=IX(L)
A2=A(I1,1)

```

```

A(I1,1)=0.
NZ=INZ(I)
IF(INZ.GT.1)GO TO 57
C DESLOCAMENTO DE ELEMENTOS NUMA LINHA
IF(INZ(I1).EQ.1)GO TO 56
K3=INZ(I1)-1
DO 303 K4=1,K3
IC(I1,K4)=IC(I1,K4+1)
A(I1,K4)=A(I1,K4+1)
303 CCNTINUE
INZ(I1)=INZ(I1)-1
GO TO 56
57 NN=IC(I,NZ)
NZ1=INZ(I1)
IF(IC(I1,NZ1).GT.NN)NN=IC(I1,NZ1)
DO 50 K=12,NN
50 X(K)=0.
IF(NZ1.EQ.1)GO TO 55
DO 51 K1=2,NZ1
K=IC(I1,K1)
X(K)=A(I1,K1)
51 CCNTINUE
55 DO 52 K1=2,NZ
K=IC(I1,K1)
X(K)=X(K)-A2*A(I,K1)
52 CCNTINUE
K1=0
INZ(I1)=INZ(I1)-1
DO 53 K=12,NN
IF(X(K).EQ.0)GO TO 53
K1=K1+1
A(I1,K1)=X(K)
IC(I1,K1)=K
53 CCNTINUE
INZ(I1)=K1
56 B(I1)=B(I1)-A2*B(I)
IF(INZ(I1).GT.NZM)NZM=INZ(I1)
LB=LB+1
AB(LB)=A2
IBO(I)=IBO(I)+1
IBM(LB)=I1
40 CONTINUE
IF(15.EQ.1)GO TO 10
20 CCNTINUE
GO TO 33
10 CONTINUE
33 IF(15.NE.3)GO TO 1001
C SEG. PARTE
C OPERACOES ATRAVES DO ARMAZENAMENTO
1000 LC1=1
N1=N-1
DO 100 I=1,N1
LC2=LC1+IBO(I)-1
J1=IBM(LC1)
J=IB(J1)
AUX=B(I)
B(I)=B(J)
B(J)=AUX
B(I)=B(I)/AB(LC1)
110 LC1=LC1+1
IF(LC1.GT.LC2)GO TO 100
I1=IBM(LC1)
A2=AB(LC1)

```

```
      B(I1)=B(I1)-A2*B(I)
      GO TO 110
100  CCNTINUE
C
C  SUBSTITUICAO REGRESSIVA
1001 J=IZ(N)
      NZ=INZ(J)
      DO 30 K=1,NZ
      IF(IC(J,K).NE.C)GO TO32
30   CONTINUE
      GO TO 150
32   B(J)=B(J)/A(J,K)
      DO 90 J=2,N
      IB=N-J+1
      I1=IZ(IB)
      NZ=INZ(I1)
      IF(IC(I1,NZ).EQ.C)GO TO 90
      DO 90 K=1,NZ
      IA=IC(I1,K)
      J1=IZ(IA)
      B(I1)=B(I1)-A(I1,K)*B(J1)
90   CCNTINUE
      DO 94 I=1,N
      J=IY(I)
      K=IZ(J)
94   X(I)=B(K)
150  RETURN
      END
```

APÊNDICE II

Lista das subrotinas:

- CASPI

- CASPE

```

C
C SUBROTINA MASPL
C OBJETIVO
C SOLUCAO DE UM SISTEMA DE EQ. ALGEBRICAS AX=B, ATRA-
C VES DO METODO DA FATORIZACAO, APLICANDO TECNICAS
C PARA MATRIZES ESPARSAS
C
C USO
C CALL MASPL(N, EPS, IS, NZM)
C DIMENSIONAMENTO
C INTEGER*2 IC, ICX, INZ, IRZ, IX, IY, IZ
C COMMON /G1/A(N,K), B(N), X(N), IC(N,K), INZ(N), IX(N),
C IY(N), IZ(N)
C DIMENSION XU(N, NZM), XL(N), ICX(N, NZM), IRZ(N)
C DESCRICAO DOS PARAMETROS
C N - NUM DE EQUACOES DO SISTEMA
C EPS - CONST. DE ENTRADA USADA NO CALC. DA TOLERANCIA
C RELATIVA(TOL) P/ TESTE DE PERDA DE SIGNIFICANCIA
C IS - OPCOES DE LSO DA SUBROTINA. SEM ORDENAMENTO INI-
C CIAL DE LINHAS P/ IS=1, E COM ORDENAMENTO P/ IS=2.
C A SAICA IS=-1 SIGNIFICA TER SIDO UTILIZADO UM PIVO
C CUJO VALOR E MENOR QUE TOL
C NZM - NUM MAX. DE ELEMENTOS FORMADOS EM LINHAS DE XU,
C E ICX, USADO PARA DEFINIR A SEGUNDA DIMENSAO DESSES
C ARRANJOS. INICIALMENTE DEVE SER ASSUMIDO UM VALOR
C PARA ESSA DIMENSAO
C NZM=0 COMO VALOR DE ENTRADA, ACARRETA NO ORDENAMENTO
C CRESCENTE POR COLUNA DOS ELEMENTOS DE A
C ARRANJOS UTILIZADOS
C A(N,K) - ARMAZENA OS VALORES NAO NULOS DE A, C/ K
C IGUAL AC NUM MAX DE ELEM NUMA LINHA DESSA MATRIZ
C B(N) - ARMAZENA O LADO DIREITO DO SISTEMA
C X(N) - SAICA COM A SOLUCAO DAS EQUACOES
C IC(N,K) - ARMAZENA AS COLUNAS DOS ELEMENTOS NAO NULOS
C DE A
C INZ(N) - DA O NUM DE ELEM EM CADA LINHA DE A(N,K)
C XU(N, NZM) - ARMAZENA LINHA APOS LINHA OS VALORES NAO
C NULOS DE U, DA EXPRESSAO A=LU
C XL(N) - ARMAZENA PROVISORIAMENTE CADA LINHA DE L, DA
C EXPRESSAO A=LU
C ICX(N, NZM) - ARMAZENA AS COLUNAS QUE PERTENCEM CA-
C DA ELEMENTO DE XU EM U
C IRZ(N) - DA O NUM DE ELEM EM CADA LINHA DE XU
C IX(N), IY(N), IZ(N) - VETORES AUXILIARES
C SUBROTINAS REQUERIDAS
C CALL ORDEM(N) - QUANDO SE DESEJA UM ARRANJO INICI-
C AL DAS LINHAS E COLUNAS DE A P/ DIMINUIR A MEM. -
C REQUERIDA. SEU USO E OPCIONAL. DEVE SER DEFINIDO -
C COMO ENTRADA NZM=C
C USO EM DUPLA PRECISAO
C DOUBLE PRECISION A, B, X, XU, XL, AUX, TOL, AM, SOM
C
C *****
C
C SUBROUTINE MASPL(N, EPS, IS, NZM)
C INTEGER*2 IC, ICX, INZ, IRZ, IX, IY, IZ
C COMMON /G1/A(100,5), B(100), X(100), IC(100,5), INZ(100),
C *IX(100), IY(100), IZ(100)
C DIMENSION XU(100,44), XL(100), ICX(100,44), IRZ(100)
C DO 600 I=1, N
C IY(I)=I
600 IZ(I)=I
C CALL ORDEM(N)
C IF(NZM.EQ.1)GO TO 205

```

## C ARRANJO DOS ELEM. DE A EM ORDEM CRESCENTE

```

DO 200 I=1,N
  J1=1
  IK=1
  NZ=INZ(I)
  IF(NZ.EQ.1)GO TO 200
204 DO 201 J=J1,N
  DO 202 K1=IK,NZ
  IF(IC(I,K1).EQ.J)GO TO 203
202 CCNTINUE
201 CCNTINUE
203 IAUX=IC(I,IK)
  IC(I,IK)=IC(I,K1)
  IC(I,K1)=IAUX
  AUX=A(I,IK)
  A(I,IK)=A(I,K1)
  A(I,K1)=AUX
  IK=IK+1
  IF(IK.GE.NZ)GO TO 200
  J1=J+1
  GO TO 204
200 CONTINUE
  NZM=1
205 DO 51 I=1,N
51 IRZ(I)=0

```

## C DETERMINACAO DA TOLERANCIA DO SISTEMA

```

AM=A(1,1)
DO 206 I=1,N
  NZ=INZ(I)
  DO 206 K=1,NZ
  IF(ABS(AM).GE.ABS(A(I,K)))GO TO 206
  AM=A(I,K)
206 CCNTINUE
  TOL=EPS*ABS(AM)

```

## C ORDENACAO DA MATRIZ A

```

DO 10 I=1,N
10 IX(I)=IZ(I)
  IF(IS.EQ.1)GO TO 7
  J=1
  IAUX=1
  I=N+1
1 I=I-1
  K=IX(I)
  NZ1=INZ(IAUX)
  NZ=INZ(K)
  IF(NZ.GE.NZ1)GO TO 6
  DO 2 K1=1,NZ
  IF(IC(K,K1).EQ.J)GO TO 3
2 CONTINUE
  GO TO 6
3 DO 4 K1=1,NZ1
  IF(IC(IAUX,K1).EQ.1)GO TO 5
4 CCNTINUE
  GO TO 6
5 IX(I)=IAUX
  IX(J)=K
  IF(J.GE.(N-1))GO TO 7
  J=J+1
  I=N+1
  IAUX=IX(J)
  GO TO 1
6 IF(I.GT.J)GO TO 1
  IF(J.GE.(N-1))GO TO 7

```

```

J=J+1
I=N+1
IAUX=IX(J)
GO TO 1
7 DO 8 I=1,N
  K=IX(I)
8 X(I)=B(K)
  DO 9 I=1,N
9 B(I)=X(I)
C
C  OBTENCAO DA PRIM.COL. DE U E PRIM.LINHA DE L
15 II=IX(1)
  NZ=INZ(II)
  KM=IC(II,NZ)
  B(1)=B(1)/A(II,1)
  IF(NZ.EQ.1)GO TO 20
  DO 14 K=2,NZ
  I=IC(II,K)
  XU(I,1)=A(II,K)/A(II,1)
  IRZ(1)=1
  ICX(I,1)=1
14 CCNTINUE
C
C  OBTENCAO DAS MATRIZES U E L
20 DO 100 I=2,N
  II=IX(I)
  K1=IC(II,1)
  NZ=INZ(II)
  K2=IC(II,NZ)
  IF(K2.GT.KM)KM=K2
  DO 21 J=K1,KM
21 X(J)=0.
  DO 22 J=1,I
22 XL(J)=0.
  NZ=INZ(II)
  DO 23 K=1,NZ
  K1=IC(II,K)
23 X(K1)=A(II,K)
C  OBTENCAO DE UMA LINHA DE L
  K2=IC(II,1)
  XL(K2)=X(K2)
  IF(K2.EQ.1)GO TO 34
  J=K2+1
  DO 26 I1=J,I
  NZ=IRZ(I1)
  SCM=0.
  IF(NZ.EQ.0)GO TO 25
  DO 24 K=1,NZ
  K1=ICX(I1,K)
  IF(XL(K1).EQ.0.)GO TO 24
  SOM=XL(K1)*XU(I1,K)+SOM
24 CCNTINUE
25 XL(I1)=X(I1)-SOM
26 CCNTINUE
C  OBTENCAO DE UMA COLUNA DE U
34 IF(I1.EQ.N)GO TO 30
  I2=I+1
  IF(I2.GT.KM)GO TO 30
  DO 29 I1=I2,KM
  SCM=0.
  NZ=IRZ(I1)
  IF(NZ.EQ.0)GO TO 28
  DO 27 K=1,NZ

```

```

K1=ICX(I1,K)
IF(XL(K1).EQ.0.)GO TO 27
SOM=XL(K1)*XU(I1,K)+SOM
27 CONTINUE
28 AUX=X(I1)-SOM
AUX=AUX/XL(I1)
IF(AUX.EQ.0)GO TO 29
IRZ(I1)=IRZ(I1)+1
K=IRZ(I1)
XU(I1,K)=AUX
ICX(I1,K)=I
29 CCNTINUE
NZ=IRZ(I1)
IF(NZ.GT.NZM)NZM=NZ
C MODIFICAO DE UM ELEMENTO DO VETOR B
30 I1=I-1
AUX=0.
K2=IC(I1,1)
IF(K2.EQ.I)GO TO 35
DO 31 J=K2,I1
IF(XL(J).EQ.0.)GO TO 31
AUX=XL(J)*B(J)+AUX
31 CCNTINUE
35 IF(ABS(XL(I1)).LE.TOL)IS=-1
B(I)=(B(I)-AUX)/XL(I)
100 CCNTINUE
C
C DETERMINAO DA MATRIZ TRANSPOSTA DE U
DO 32 I=2,N
NZ=IRZ(I)
IF(NZ.EQ.0)GO TO 32
IRZ(I)=0
DO 32 K=1,NZ
J=ICX(I,K)
IRZ(J)=IRZ(J)+1
K1=IRZ(J)
ICX(J,K1)=I
XU(J,K1)=XU(I,K)
32 CCNTINUE
DO 33 I=2,N
NZ=IRZ(I)
IF(NZ.GT.NZM)NZM=NZ
33 CONTINUE
C
C SUBSTITUICAO REGRESSIVA
DO 40 J=2,N
I1=N-J+1
NZ=IRZ(I1)
IF(NZ.EQ.0)GO TO 40
DO 40 K=1,NZ
K1=ICX(I1,K)
B(I1)=B(I1)-XU(I1,K)*B(K1)
40 CCNTINUE
DO 41 I=1,N
J=IX(I)
K=IY(J)
41 X(I)=B(K)
RETURN
END

```

```

C  SUBROTINA MASP2
C  OBJETIVO
C      SOLUCAO DE UM SISTEMA DE EQ. ALGEBRICAS AX=B, COM
C      A FIXO E B VARIAVEL, ATRAVES DO METODO DA FATORIZA-
C      CAO, APLICANDO TECNICAS PARA MATRIZES ESPARSAS.
C  LSC
C      CALL MASP2(IN, EPS, IS, NZM, L)
C  DIMENSIONAMENTO
C      INTEGER*2 IC, ICX, INZ, IRZ, IZL, IX, IY, IZ
C      COMMON /G1/A(N,K), B(N), X(N), XU(N, NZM), IC(N,K), ICX(N,
C      NZM), INZ(N), IRZ(N), IX(N), IY(N), IZ(N)
C      COMMON /G2/XML(L), JB(L), IZL(N)
C      DIMENSION XL(N)
C  DESCRICAO DOS PARAMETROS
C      N - NUM DE EQUACOES DO SISTEMA
C      EPS - CONST. DE ENTRADA USADA NO CALC. DA TOLERANCIA
C      RELATIVA(TOL) P/ TESTE DE PERDA DE SIGNIFICANCIA
C      IS - OPCOES DE LSO DA SUBROTINA SEM ORDENACAO INI-
C      CIAL DE LINHAS P/ IS=1, E COM ORDENACAO P/IS=2
C      A SAICA IS=-1 SIGNIFICA TER SIDO UTILIZADO UM PIVO
C      CUJO VALOR E MENOR QUE TOL
C      PARA IS=3, E USADA A SEGUNDA PARTE DA SUBROTINA
C      NZM - NUM MAX DE ELEMENTOS FORMADOS EM LINHAS DE XU,
C      E ICX, USADO PARA DEFINIR A SEGUNDA DIMENSAO DESSES
C      ARRANJOS. INICIALMENTE DEVE SER ASSUMIDO UM VALOR
C      PARA ESSA DIMENSAO
C      NZM=0 COMO VALOR DE ENTRADA, ACARRETA NA ORDENACAO
C      CRESCENTE POR COLUNA DOS ELEMENTOS DE A
C      L - COMO SAIDA, INDICA A DIMENSAO NECESSARIA P/ A
C      UTILIZACAO DOS ARRANJOS AB E IBM.
C  ARRANJOS UTILIZADOS
C      A(N,K) - ARMAZENA OS VALORES NAO NULOS DE A, C/ K
C      IGUAL AO NUM MAX DE ELEM NUMA LINHA DESSA MATRIZ
C      B(N) - ARMAZENA O LADO DIREITO DO SISTEMA
C      X(N) - SAIDA COM A SOLUCAO DAS EQUACOES
C      IC(N,K) - ARMAZENA AS COLUNAS DOS ELEMENTOS NAO NULOS
C      DE A
C      INZ(N) - DA O NUM DE ELEM EM CADA LINHA DE A(N,K)
C      XU(N, NZM) - ARMAZENA LINHA APOS LINHA OS VALORES NAO
C      NULOS DE U, DA EXPRESSAO A=LU
C      XL(N) - ARMAZENA PROVISORIAMENTE CADA LINHA DE L, DA
C      EXPRESSAO A=LU
C      ICX(N, NZM) - ARMAZENA AS COLUNAS QUE PERTENCEM CA-
C      DA ELEMENTO DE XU EM U
C      IRZ(N) - DA O NUM DE ELEM EM CADA LINHA DE XU
C      IX(N), IY(N), IZ(N) - VETORES AUXILIARES
C      XML(L) - ARMAZENA OS VALORES DOS ELEMENTOS DA MATRIZ
C      L OBTIDA NA PRIM. PARTE DA SUBROTINA.
C      JB(L) - ARMAZENA AS POSICOES NAS COLUNAS DE L DOS
C      ELEMENTOS CORRESPONDENTES EM XML.
C      IZL(N) - VETOR QUE INDICA O NUM. DE ELEMENTOS LOCA-
C      LIZADOS EM CADA LINHA DE L.
C  SUBROTINAS REQUERIDAS
C      CALL ORDEM(N) - QUANDO SE DESEJA UM ARRANJO INICI-
C      AL DAS LINHAS E COLUNAS DE A P/ DIMINUIR A MEM. -
C      REQUERIDA. SEU LSO E OPCIONAL. DEVE SER DEFINIDO -
C      COMO ENTRADA NZM=0
C  LSC EM DUPLA PRECISAO
C      DOUBLE PRECISION A, B, X, XU, XL, XML, AUX, TOL, AM, SCM
C      *****
C
C      SUBROUTINE MASP2(IN, TOL, NZM, KN, IS, L)
C      INTEGER*2 IC, ICX, INZ, IRZ, IX, IZL

```

```

COMMON /G1/A(1225,5),B(1225),X(1225),XU(1225,35),IC(1225,5),
*ICX(1225,35),INZ(1225),IRZ(1225),IX(1225),IY(1225),IZ(1225)
COMMON /G2/XML(4000),JB(4000),IZL(4000)
DIMENSION XL(1225)
IF(15.EQ.3)GO TO 110
L=0
DO 51 I=1,N
  IRZ(I)=0
  IY(I)=1
51  IZ(I)=I
  CALL ORDEN(N)
  IF(NZM.EQ.1)GO TO 205
C  ARRANJO DOS ELEM. DE A EM ORDEM CRESCENTE
DO 200 I=1,N
  J1=1
  IK=1
  NZ=INZ(I)
  IF(NZ.EQ.1)GO TO 200
204  DO 201 J=J1,N
      DO 202 K1=IK,NZ
          IF(IC(I,K1).EQ.J)GO TO 203
202  CONTINUE
201  CONTINUE
203  IAUX=IC(I,IK)
      IC(I,IK)=IC(I,K1)
      IC(I,K1)=IAUX
      AUX=A(I,IK)
      A(I,IK)=A(I,K1)
      A(I,K1)=AUX
      IK=IK+1
      IF(IK.GE.NZ)GO TO 200
      J1=J+1
      GO TO 204
200  CONTINUE
      NZM=1
C  DETERMINACAO CA TOL DO SISTEMA
      AM=A(1,1)
      DO 206 I=1,N
          NZ=INZ(I)
          DO 206 K=1,NZ
              IF(ABS(AM).GE.ABS(A(I,K)))GO TO 206
              AM=A(I,K)
206  CONTINUE
          TCL=EPS*ABS(AM)
          L=0
          DO 51 I=1,N
              IRZ(I)=0
C  ORDENACAO DA MATRIZ A
          DO 10 I=1,N
              IX(I)=1
              IF(15.EQ.1)GO TO 15
              J=1
              IAUX=1
              I=N+1
              1  I=I-1
                  K=IX(I)
                  NZ1=INZ(IAUX)
                  NZ=INZ(K)
                  IF(NZ.GT.NZ1)GO TO 6
                  DO 2 K1=1,NZ
                      IF(IC(K,K1).EQ.J)GO TO 3
              2  CONTINUE
                  GO TO 6

```

```

3 DO 4 K1=1,NZ1
  IF(IC(IAUX,K1).EQ.1)GO TO 5
4 CONTINUE
  GO TO 6
5 IX(I)=IAUX
  IX(J)=K
  IF(J.GE.(N-1))GO TO 7
  J=J+1
  I=N+1
  IAUX=IX(J)
  GO TO 1
6 IF(I.GT.J)GO TO 1
  IF(J.GE.(N-1))GO TO 7
  J=J+1
  I=N+1
  IAUX=IX(J)
  GO TO 1
7 DO 8 I=1,N
  K=IX(I)
8 X(I)=B(K)
  DO 9 I=1,N
9 B(I)=X(I)
C
C OBTENCAO DA PRIM.COL. DE U E PRIM.LINHA DE L
15 II=IX(1)
  NZ=INZ(II)
  KM=IC(II,NZ)
  B(1)=B(1)/A(II,1)
  IF(NZ.EQ.1)GO TO 20
  DO 14 K=2,NZ
  I=IC(II,K)
  XU(1,1)=A(II,K)/A(II,1)
  IRZ(I)=1
  ICX(I,1)=1
14 CCNTINUE
C
C OBTENCAO DAS MATRIZES U E L
20 DO 100 I=2,N
  II=IX(I)
  K1=IC(II,1)
  NZ=INZ(II)
  K2=IC(II,NZ)
  IF(K2.GT.KM)KM=K2
  DO 21 J=K1,KM
21 X(J)=0.
  DO 22 J=1,I
22 XL(J)=0.
  NZ=INZ(II)
  DO 23 K=1,NZ
  K1=IC(II,K)
23 X(K1)=A(II,K)
C OBTENCAO DE UMA LINHA DE L
  K2=IC(II,1)
  XL(K2)=X(K2)
  IF(K2.EQ.1)GO TO 34
  J=K2+1
  DO 26 I1=J,I
  NZ=IRZ(I1)
  SOM=0.
  IF(NZ.EQ.0)GO TO 25
  DO 24 K=1,NZ
  K1=IC(I1,K)
  IF(XL(K1).EQ.0.)GO TO 24

```

```

SCM=XL(K1)*XU(I1,K1)+SOM
24 CONTINUE
25 XL(I1)=X(I1)-SOM
26 CONTINUE
C OBTENCAO DE UMA COLUNA DE U
34 IF(I.EQ.N)GO TO-30-
    I2=I+1
    IF(I2.GT.KM)GO TO 30
    DO 29 I1=I2,KM
    SOM=0.
    NZ=IRZ(I1)
    IF(NZ.EQ.0)GO TO 28
    DO 27 K=1,NZ
    K1=ICX(I1,K)
    IF(XL(K1).EQ.0.)GO TO 27
    SCM=XL(K1)*XU(I1,K)+SOM
27 CONTINUE
28 AUX=X(I1)-SOM
    AUX=AUX/XL(I)
    IF(AUX.EQ.0)GO TO 29
    IRZ(I1)=IRZ(I1)+1
    K=IRZ(I1)
    XU(I1,K)=ALX
    ICX(I1,K)=I
29 CCNTINUE
    NZ=IRZ(I)
    IF(NZ.GT.NZM)NZM=NZ
C SUBSTITUICAO DIRETA DE L SOBRE B
30 I1=I-1
    IZL(I)=0
    AUX=0.
    K2=IC(I1,1)
    IF(K2.EQ.1)GO TO 35
    DO 31 J=K2,I1
    IF(XL(J).EQ.0.)GO TO 31
    L=L+1
    IZL(I)=IZL(I)+1
    JB(L)=J
    XML(L)=XL(J)
    AUX=XL(J)*B(J)+AUX
31 CONTINUE
    IF(ABS(XL(I)).LE.TGL)IS=-1
35 L=L+1
    IZL(I)=IZL(I)+1
    JB(L)=I
    XML(L)=XL(I)
    B(I)=(B(I)-AUX)/XL(I)
100 CCNTINUE
C
C DETERMINACAO DA MATRIZ TRANSPOSTA DE U
DO 32 I=2,N
    NZ=IRZ(I)
    IF(NZ.EQ.0)GO TO 32
    IRZ(I)=0
    DO 32 K=1,NZ
    J=ICX(I,K)
    IRZ(J)=IRZ(J)+1
    K1=IRZ(J)
    ICX(J,K1)=I
    XU(J,K1)=XU(I,K)
32 CCNTINUE
    DO 33 I=2,N
    NZ=IRZ(I)

```

```

      IF(NZ.GT.NZM)NZM=NZ
33  CCNTINUE
      IF(IS.NE.2)GO TO 120
C
C  PARTE2  CALCULO ATRAVES DE OPERACOES ARMAZENADAS
C  SUBSTITUICAO CIRETA DE L SOBRE B
110  DO 111 I=1,N
      K=IX(I)
111  X(I)=B(K)
      DO 112 I=1,N
112  B(I)=X(I)
      II=IX(I)
      K1=1
      B(I)=B(I)/A(II,1)
      DO 115 I=2,N
      K2=IZL(I)+K1-2
      AUX=0.
      IF(K2.LT.K1)GO TO 114
      DO 113 K=K1,K2
      J=JB(K)
113  AUX=AUX+XML(K)*B(J)
114  K2=K2+1
      B(I)=(B(I)-AUX)/XML(K2)
      K1=K2+1
115  CONTINUE
C
C  SUBSTITUICAO REGRESSIVA
120  DO 40 J=2,N
      I1=N-J+1
      NZ=IRZ(I1)
      IF(NZ.EQ.0)GO TO 40.
      DO 40 K=1,NZ
      K1=ICX(I1,K)
      B(I1)=B(I1)-XU(I1,K)*B(K1)
40  CCNTINUE
      DO 41 I=1,N
      J=IX(I)
      K=IY(J)
41  X(I)=B(K)
      RETURN
      END

```

AVÉROICE III

listagem da subrotina:

- ORDEN

```

C  SUBROTINA ORDEM
C  OBJETIVO
C      ARRANJO DAS LINHAS E COLUNAS DE UMA MATRIZ ESPARSA
C      A, DE UM SISTEMA AX=B, EM ORDEM CRESCENTE DE NUME-
C      RO DE ELEMENTOS
C
C      CALL ORDEM(N)
C  DIMENSIONAMENTO
C      INTEGER*2 IC, INZ, IX, IY, IZ
C      COMMON/G1/ A(N,NZM),B(N),X(N),IC(N,NZM),INZ(N),
C      IX(N),IY(N),IZ(N)
C      N - NUM. DE EQUACOES DO SISTEMA
C  OBSERVACOES
C      - ESSA SUBROTINA E CHAMADA PELAS SUB. SPAMA1,SPAMA2
C      MASP1 OU MASP2. UTILIZA O MESMO BLOCO DE COMMON DAS
C      REFERIDAS SUBROTINAS, POREM OPERA SOMENTE COM IC,
C      INZ,IY E IZ
C      - A MENSAGEM 'MATRIZ MAL DEFINIDA' ALERTA A POS-
C      SIBILIDADE DA NAO EXISTENCIA DE UM PIVO NAO NULO
C      DURANTE O PROCESSAMENTO
C
C *****
C      SUBROUTINE ORDEM(N)
C      INTEGER*2 IC, INZ, IX, IY, IZ
C      COMMON /G1/A(600,100),B(600),X(600),IC(600,100),INZ(600),
C      *IX(600),IY(600),IZ(600)
C      DO 201 I=1,N
201  IY(I)=0
C  DETERMINACAO DO NUM DE ELEM. P/CADA COLUNA
C      DO 202 I=1,N
C      NZ=INZ(I)
C      DO 202 K=1,NZ
C      I1=IC(I,K)
C      IY(I1)=IY(I1)+1
202  CONTINUE
C      K2=0
C      K1=N
C      DO 203 I=1,N
C      IF(IY(I).GT.K2)K2=IY(I)
C      IF(IY(I).LT.K1)K1=IY(I)
203  CONTINUE
C  ARRANJO DE IY EM ORDEM CRESCENTE
C      J=1
C      DO 205 K=K1,K2
C      I1=J
C      DO 204 I=I1,N
C      IF(IY(I).NE.K)GO TO 204
C      IY(I)=IY(J)
C      IY(J)=IZ(I)
C      IZ(I)=IZ(J)
C      IZ(J)=IY(J)
C      J=J+1
204  CONTINUE
205  CONTINUE
C      DO 218 I=1,N
C      K=IY(I)
C      IZ(K)=I
218  CONTINUE
C      DO 219 I=1,N
219  IY(I)=IZ(I)
C  REARRANJO DA MATRIZ A
C      DO 206 I=1,N
C      NZ=INZ(I)

```

```
DO 206 K=1,NZ
K1=IC(I,K)
IC(I,K)=IZ(K1)
206 CCNTINUE
C ORCENAMENTO DE A POR LINHAS
DO 207 I=1,N
207 IZ(I)=I
I1=1
208 K1=IZ(I1)
K2=I1
NZ1=INZ(K1)
DO 209 K=1,NZ1
IF(IC(K1,K).EQ.I1)GO TO 210
209 CONTINUE
NZ1=N+1
210 DO 212 I=I1,N
J=IZ(I)
NZ=INZ(J)
IF(NZ.GE.NZ1)GO TO 212
DO 211 K=1,NZ
IF(IC(J,K).NE.I1)GO TO 211
NZ1=NZ
K2=I
GO TO 212
211 CONTINUE
212 CCNTINUE
IF(NZ1.LT.(N+1))GO TO 215
NZ=INZ(K2)
DO 220 K=1,NZ
IF(IC(K2,K).LE.K2)GO TO 215
220 CCNTINUE
WRITE(6,350)
350 FORMAT(/,5X,'MATRIZ MAL DEFINIDA',/)
215 IZ(I1)=IZ(K2)
IZ(K2)=K1
I1=I1+1
IF(I1.LE.N)GO TO 208
RETURN
END
```

APÊNDICE IV

listagem dos programas de teste:

- PT1 aplicado às subrotinas SPAMAT e MASP
- PT1 aplicado à subrotina GELB
- PT2 aplicado às subrotinas SPAMAT e MASP
- PT2 aplicado à subrotina GELB
- PT3
- PT4

C PROGRAMA DE TESTE NUM. 1 - PT1  
 C DISTRIBUICAO DE TEMP. NUMA BARRA UNIDIMENSIONAL  
 C

```

    INTEGER*2 IC, INZ, IX, IY, IZ
    COMMON /G1/ A(1200,3), B(1200), X(1200), IC(1200,3), INZ(1200),
    *          IX(1200), IY(1200), IZ(1200)
    N=1200
    XL=2.
    DX=XL/(N-1)
    TX1=200.
    TX2=50.
    XHX1=1.
    XHX2=1.
    XK=-1
    IK=1
    A(IK,1)=XHX1+XK/DX
    IC(IK,1)=1
    A(IK,2)=-XK/DX
    IC(IK,2)=2
    INZ(IK)=2
    B(IK)=XHX1*TX1
    I11=N-1
    DO 11 I=2, I11
    IK=IK+1
    A(IK,1)=DX**2
    IC(IK,1)=IK-1
    A(IK,2)=-2.*(DX**2)
    IC(IK,2)=IK
    A(IK,3)=DX**2
    IC(IK,3)=IK+1
    B(IK)=0.
    INZ(IK)=3
11  CCNTINUE
    IK=IK+1
    A(IK,1)=-XK/DX
    IC(IK,1)=N-1
    A(IK,2)=XHX2+XK/DX
    IC(IK,2)=N
    B(IK)=XHX2*TX2
    INZ(IK)=2
    IK=1
    B(IK)=XHX1*TX1
    I11=N-1
    DO 15 I=2, I11
    IK=IK+1
    B(IK)=0.
15  CCNTINUE
    IK=IK+1
    B(IK)=XHX2*TX2
    EPS=10.**(-7)
    EPS=10.**(-12)
    NZM=0
    NZM=1
    IS=2
    IS=1
    IF=3
    CALL SPAM1(N, EPS, IS, NZM, IF)
    WRITE(6, 111) NZM
111 FORMAT(/, 10X, 'NUM MAX DE ELEM=', I5)
    WRITE(6, 112) IS
112 FORMAT(/, 10X, 'IS=', I2)
    K=0
106 L=0
  
```

```
WRITE(6,101)
101 FORMAT('1',22X,'TEMPERATURE DISTRIBUTION',///)
105 IF(L.EQ.51)GO TO 106
K=K+1
L=L+1
K2=4*K
K1=K2-3
IF(K2.GT.N)K2=N
WRITE(6,102)(I,X(I),I=K1,K2)
102 FORMAT(10X,4('T(',I3,')=',F6.1,3X))
IF(K2.LT.N)GO TO 105
STOP
END
```

C P.PRINCIPAL SUB. GELB  
 C PROGRAMA DE TESTE NUM. 1 - PT1  
 C DISTRIBUICAO DE TEMP. NUMA BARRA UNIDIMENSIONAL  
 C

```

DIMENSION A(8000),B(2000)
N=1
II=1
M=2000
JJ1=M-1
XL=2.
DX=XL/(M-1)
TX1=200.
TX2=50.
XHX1=1.
XHX2=1.
XK=.1
MUD=II
MLD=II
ME=M*(1+MUD+MLD)-II*(1+II)
DO 50 I=1,ME
50  A(I)=0.
   A(I)=XHX1+XK/DX
   A(2)=-XK/DX
   B(1)=XHX1*TX1
   DO 31 K=1,K1
   J1=K*M+1
31  B(J1)=B(1)
   L=2
   DO 2 J=2,JJ1
   L=L+1
   A(L)=DX**2
   L=L+1
   A(L)=-2.*DX**2
   L=L+1
   A(L)=DX**2
   B(J)=0
   DO 32 K=1,K1
   J1=K*M+J
32  B(J1)=B(J)
2   CONTINUE
   L=L+1
   A(L)=-XK/DX
   L=L+1
   A(L)=XHX2+XK/DX
   B(M)=XHX2*TX2
   DO 33 K=1,K1
   J1=K*M+M
33  B(J1)=B(M)
   WRITE(6,103)M
103 FORMAT('1',20X,'NUM.DE EQ.=' ,I5)
   CALL GELB(B,A,M,N,MUD,MLD,EPS,IER)
   WRITE(6,100)IER
100 FORMAT(20X,' IER=' ,I3)
   K3=K1+1
   DO 35 MN=1,K3
   K=0
   M1=M+M*(MN-1)
105  L=0
   WRITE(6,101)
101  FORMAT('1',22X,'TEMPERATURE DISTRIBUTION',///)
105  IF(L.EQ.99)GO TO 106
   K=K+1
   L=L+1

```

```
K2=4*K+M*(MN-1)
K1=K2-3
IF(K2.GT.M1)K2=M1
WRITE(6,102)((I,B(I),I=K1,K2)
102  FORMAT(10X,4('I(',14,')=' ,F6.1,3X))
IF(K2.LT.M1)GO TO 105
35  CCNTINUE
306 STOP
END
```

C PROGRAMA DE TESTE PT2  
 C DISTRIBUICAO DE TEMPERATURAS NUMA PLACA BIDIMENSIONAL  
 C

```

    INTEGER*2 ICP,INP,IC,INZ,IX,IY,IZ
    COMMON /G1/A(600,21),B(600),X(600),IC(600,21),INZ(600),
    *      IX(600),IY(600),IZ(600)
    II=20
    JJ=30
10  N=II*JJ
    XL1=1.
    XL2=2.
    DX=XL1/(II-1)
    DY=XL2/(JJ-1)
    TX1=200.
    TX2=100.
    TY1=50.
    TY2=250.
    XHX1=1.
    XHX2=1.
    XHY1=1.
    XHY2=1.
    XK=.1
    IK=0
    DO 11 I=1, II
    IK=IK+1
    A(IK,1)=XHY1+XK/DY
    IC(IK,1)=IK
    A(IK,2)=-XK/DY
    IC(IK,2)=IK+II
    B(IK)=XHY1+TY1
    INZ(IK)=2
11  CONTINUE
    JJ1=JJ-1
    DO 12 J=2, JJ1
    IK=IK+1
    A(IK,1)=XHX1+XK/DX
    IC(IK,1)=IK
    A(IK,2)=-XK/DX
    IC(IK,2)=IK+1
    INZ(IK)=2
    B(IK)=XHX1*TX1
    II1=II-1
    DO 13 I=2, II1
    IK=IK+1
    A(IK,1)=DX**2
    IC(IK,1)=IK-II
    A(IK,2)=DY**2
    IC(IK,2)=IK-1
    A(IK,3)=-2.*(DX**2+DY**2)
    IC(IK,3)=IK
    A(IK,4)=DY**2
    IC(IK,4)=IK+1
    A(IK,5)=DX**2
    IC(IK,5)=IK+II
    B(IK)=0
    INZ(IK)=5
13  CONTINUE
    IK=IK+1
    A(IK,1)=-XK/DX
    IC(IK,1)=IK-1
    A(IK,2)=XHX2+XK/DX
    IC(IK,2)=IK
    B(IK)=XHX2*TX2
  
```

```

      INZ(IK)=2
12  CONTINUE
      DO 14 I=1,II
      IK=IK+1
      A(IK,1)=-XK/DY
      IC(IK,1)=IK-11
      A(IK,2)=(XHY2+XK/DY)
      IC(IK,2)=IK
      B(IK)=XHY2*TY2
      INZ(IK)=2
14  CONTINUE
      IS=1
      IS=2
      EPS=10.**(-7)
      NZM=1
      IF=20
      CALL SPAMA1(N,EPS,IS,NZM,IF)
      WRITE(6,111)NZM
111  FORMAT(/,10X,'NZM=',I5)
      K=0
106  L=0
      WRITE(6,101)
101  FORMAT('1',22X,'TEMPERATURE DISTRIBUTION',///)
105  IF(L.EQ.51)GO TO 106
      K=K+1
      L=L+1
      K2=4*K
      K1=K2-3
      IF(K2.GT.N)K2=N
      WRITE(6,102) (I,X(I),I=K1,K2)
102  FORMAT(10X,4('I1',I4,'')=',F6.1,3X))
      IF(K2.LT.N)GO TO 105
5   STOP
      END

```

C P.PRINCIPAL SUB. GELB  
 C PROGRAMA DE TESTE NUM. 2 - PT2  
 C DISTRIBUICAO DE TEMP. NUMA PLACA BIDIMENSIONAL  
 C

```

DIMENSION A(85715),B(1225)
II=35
JJ=35
15 III=II-1
JJ1=JJ-1
XL1=1.
XL2=2.
DX=XL1/(II-1)
DY=XL2/(JJ-1)
TX1=200.
TX2=100.
TY1=50.
TY2=250.
XHX1=1.
XHX2=1.
XHY1=1.
XHY2=1.
XK=.1
M=II*JJ
N=2
N=1
K1=1
MUC=II
MLD=II
EPS=10.**(-9)
ME=M*(1+MUC+MLD)-II*(1+II)
DO 50 I=1,ME
50 A(I)=0.
L=0
NFU=-II
DO 1 I=1,II
L=L+1
NFU=NFU+II+L-1
K=NFU+L
A(K)=XHY1+XK/DY
K=NFU+L+II
A(K)=-XK/DY
B(L)=XHY1*TY1
1 CONTINUE
DO 2 J=2,JJ1
L=L+1
IF(J-2)10,10,11
10 NFU=NFU+II+L-1
GO TO 12
11 NFU=NFU+2*II+1
12 K=NFU+II+1
A(K)=XHX1+XK/DX
K=NFU+II+2
A(K)=-XK/DX
B(L)=XHX1*TX1
DO 3 I=2,III
L=L+1
NFU=NFU+2*II+1
K=NFU+II+1
A(K)=-2.*(DX**2+DY**2)
K=NFU+II
A(K)=DY**2
K=NFU+II+2
A(K)=DY**2

```

```

K=NFU+1
A(K)=DX**2
K=NFU+2*II+1
A(K)=DX**2
B(L)=0.
3 CONTINUE
L=L+1
NFU=NFU+2*II+1
K=NFU+II+1
A(K)=XHX2+XK/DX
K=NFU+II
A(K)=-XK/DX
B(L)=XHX2+TX2
2 CONTINUE
NFU=NFU+2*II+1
L=L+1
K=NFU+II+1
A(K)=XHY2+XK/DY
K=NFU+1
A(K)=-XK/DY
B(L)=XHY2+TY2
DO 4 I=1,III
L=L+1
NFU=NFU+2*II+1-I
K=NFU+II+1
A(K)=XHY2+XK/DY
K=NFU+1
A(K)=-XK/DY
B(L)=XHY2+TY2
4 CONTINUE
WRITE(6,103)L
103 FORMAT(20X,'L= ',I4,'**')
WRITE(6,600)II,JJ
600 FORMAT(/,10X,'II=',I4,20X,'JJ=',I4)
CALL GELB(B,A,M,N,MUC,MLD,EPS,IER)
WRITE(6,100)IER
100 FORMAT(10X,'IER=',I5)
K3=K1+1
DO 35 MN=1,K3
K=0
M1=M+M*(MN-1)
106 L=0
WRITE(6,101)
101 FORMAT('1',22X,'TEMPERATURE DISTRIBUTION',///)
105 IF(L.EQ.55)GO TO 106
K=K+1
L=L+1
K2=4*K+M*(MN-1)
K1=K2-3
IF(K2.GT.M1)K2=M1
WRITE(6,102)(I,B(I),I=K1,K2)
102 FORMAT(10X,4('I(',I4,')=',F6.1,3X))
IF(K2.LT.M1)GO TO 105
35 CONTINUE
9 STOP
END

```

C PROGRAMA DE TESTE NUM 3 - PT3  
 C DISTRIBUICAO DE TEMPERATURAS NUMA PLACA BIDIMENSIONAL  
 C

INTEGER\*2 IC,INZ,IX,IY,IZ  
 COMMON /G1/A(900,5),B(900),X(900),IC(900,5),INZ(900)  
 \* IX(900),IY(900),IZ(900)

II=30  
 JJ=30  
 N=II\*JJ  
 XL1=1.  
 XL2=2.  
 DX=XL1/(II-1)  
 CY=XL2/(JJ-1)  
 TX1=200.  
 TX2=100.  
 TY1=50.  
 TY2=250.  
 XHX1=1.  
 XHX2=1.  
 XHY1=1.  
 XHY2=1.  
 XK=.1  
 IK=0  
 III=II-1  
 DO 11 I=1,III  
 IK=IK+1  
 A(IK,1)=XHY1\*XK/DY  
 IC(IK,1)=IK  
 A(IK,2)=-XK/DY  
 IC(IK,2)=2\*(II-1)+2\*(JJ-1)+IK-1  
 INZ(IK)=2  
 B(IK)=XHY1\*TY1  
 11 CCNTINUE  
 IK=IK+1  
 A(IK,1)=XHY1\*XK/DY  
 IC(IK,1)=IK  
 A(IK,2)=-XK/DY  
 IC(IK,2)=IK+1  
 INZ(IK)=2  
 B(IK)=XHY1\*TY1  
 JJ1=JJ-1  
 DO 12 I=2,JJ1  
 IK=IK+1  
 A(IK,1)=XHX2\*XK/DX  
 A(IK,2)=-XK/DX  
 IC(IK,1)=IK  
 J1=2\*(II-1)+2\*(JJ-1)  
 IC(IK,2)=J1+(I-2)\*(II-2)+(II-2)  
 INZ(IK)=2  
 B(IK)=XHX2\*TX2  
 12 CCNTINUE  
 IK=IK+1  
 A(IK,1)=-XK/DY  
 IC(IK,1)=IK-1  
 A(IK,2)=(XHY2\*XK/DY)  
 IC(IK,2)=IK  
 INZ(IK)=2  
 B(IK)=XHY2\*TY2  
 J2=J1+(JJ-2)\*(II-2)+1  
 III=II-1  
 DO 13 I=2,III  
 IK=IK+1  
 A(IK,1)=(XHY2\*XK/DY)

```

A(IK,2)=-XK/DY
IC(IK,1)=IK
IC(IK,2)=J2-I+1
INZ(IK)=2
B(IK)=XHY2*TY2
13 CONTINUE
IK=IK+1
A(IK,1)=(XHY2+XK/DY)
A(IK,2)=-XK/DY
IC(IK,1)=IK
IC(IK,2)=IK+1
INZ(IK)=2
B(IK)=XHX2*TY2
DO 14 I=2, JJ1
IK=IK+1
J2=J1+(JJ1-I)*(II-2)+1
A(IK,1)=XHX1+XK/DX
A(IK,2)=-XK/DX
IC(IK,1)=IK
IC(IK,2)=J2
INZ(IK)=2
B(IK)=XHX1*TX1
14 CONTINUE
II2=II-2
DO 15 I=2, II2
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=I
A(IK,2)=DY*DY
IC(IK,2)=IK-1
A(IK,3)=-2.*(DX*DX+DY*DY)
IC(IK,3)=IK
A(IK,4)=DY*DY
IC(IK,4)=IK+1
A(IK,5)=DX*DX
IC(IK,5)=IK+II-2
INZ(IK)=5
B(IK)=0.
15 CONTINUE
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=II-1
A(IK,2)=DY*DY
IC(IK,2)=II+1
A(IK,3)=DY*DY
IC(IK,3)=IK-1
A(IK,4)=-2.*(DX*DX+DY*DY)
IC(IK,4)=IK
A(IK,5)=DX*DX
IC(IK,5)=IK+II-2
INZ(IK)=5
B(IK)=0.
JJ2=JJ-2
DO 16 J=3, JJ2
IK=IK+1
A(IK,1)=DY*DY
IC(IK,1)=J1-J+2
A(IK,2)=DX*DX
IC(IK,2)=IK-(II-2)
A(IK,3)=-2.*(DX*DX+DY*DY)
IC(IK,3)=IK
A(IK,4)=DY*DY
IC(IK,4)=IK+1

```

```

A(IK,5)=DX*DX
IC(IK,5)=IK+(II-2)
INZ(IK)=5
B(IK)=0.
II2=II-2
DO 17 I=3, II2
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=IK-(II-2)
A(IK,2)=DY*DY
IC(IK,2)=IK-1
A(IK,3)=-2.*(CX*DX+CY*DY)
IC(IK,3)=IK
A(IK,4)=DY*DY
IC(IK,4)=IK+1
A(IK,5)=DX*DX
IC(IK,5)=IK+II-2
INZ(IK)=5
B(IK)=0.
17 CONTINUE
IK=IK+1
A(IK,1)=DY*DY
IC(IK,1)=II+J-1
A(IK,2)=DX*DX
IC(IK,2)=IK-(II-2)
A(IK,3)=DY*DY
IC(IK,3)=IK-1
A(IK,4)=-2.*(DX*DX+DY*DY)
IC(IK,4)=IK
A(IK,5)=DX*DX
IC(IK,5)=IK+(II-2)
INZ(IK)=5
B(IK)=0.
16 CONTINUE
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=2*(II-1)+JJ-1
A(IK,2)=CY*DY
IC(IK,2)=2*(II-1)+JJ+1
A(IK,3)=DX*DX
IC(IK,3)=IK-(II-2)
A(IK,4)=-2.*(CX*DX+DY*DY)
IC(IK,4)=IK
A(IK,5)=CY*DY
IC(IK,5)=IK+1
INZ(IK)=5
B(IK)=0.
II2=II-2
DO 18 I=3, II2
IK=IK+1
A(IK,1)=CX*DX
IC(IK,1)=2*II+JJ-1-I
A(IK,2)=DX*DX
IC(IK,2)=IK-(II-2)
A(IK,3)=DY*DY
IC(IK,3)=IK-1
A(IK,4)=-2.*(CX*DX+DY*DY)
IC(IK,4)=IK
A(IK,5)=DY*DY
IC(IK,5)=IK+1
INZ(IK)=5
B(IK)=0.
18 CONTINUE

```

```
IK=IK+1
A(IK,1)=DY*DY
IC(IK,1)=II+JJ-2
A(IK,2)=DX*DX
IC(IK,2)=II+JJ
A(IK,2)=DX*DX
IC(IK,3)=IK-(II-2)
A(IK,4)=DY*DY
IC(IK,4)=IK-1
A(IK,5)=-2.*(DX*DX+DY*DY)
IC(IK,5)=IK
INZ(IK)=5
B(IK)=0.
IS=1
IS=2
NZM=0
NZM=1
EPS=10.*(-7)
CALL MASP1(N, EPS, IS, NZM)
WRITE(6,131)IS
131 FORMAT(/,10X,'IS=',I5)
WRITE(6,132)NZM
132 FORMAT(/,10X,'NZM=',I5)
K=0
106 L=0
WRITE(6,101)
101 FORMAT('1',22X,'TEMPERATURE DISTRIBUTION',/)
105 IF(L.EQ.51)GO TO 106
K=K+1
L=L+1
K2=4*K
K1=K2-3
IF(K2.GT.N)K2=N
WRITE(6,102)(I,X(I),I=K1,K2)
102 FORMAT(10X,4('T(',I4,')=',F6.1,3X))
IF(K2.LT.N)GO TO 105
STOP
END
```

```

C   PROGRAMA DE TESTE NUM 4 - PT4
C   DISTRIBUICAO DE TEMPERATURAS NUMA PLACA BIDIMENSIONAL
      INTEGER*2 IC, INZ, IX, IY, IZ
      COMMON /G1/A(100,30), B(100), X(100), IC(100,30), INZ(100),
      *      IX(100), IY(100), IZ(100)
      II=10
      JJ=10
      N=II*JJ
      XL1=1.
      XL2=2.
      DX=XL1/(II-1)
      DY=XL2/(JJ-1)
      TX1=200.
      TX2=100.
      TY1=50.
      XHX1=1.
      XHX2=1.
      TY2=250.
      XHY1=1.
      XHY2=1.
      XK=.1
      II2=II-2
      J1=2*(II-1)+2*(JJ-1)
      IK=1
      A(IK,1)=-2.*(DX*DX+DY*DY)
      IC(IK,1)=IK
      A(IK,2)=DY*DY
      IC(IK,2)=IK+1
      A(IK,3)=DX*DX
      IC(IK,3)=IK+II2
      A(IK,4)=DX*DX
      J2=N-J1
      IC(IK,4)=J2+2
      A(IK,5)=DY*DY
      IC(IK,5)=N
      INZ(IK)=5
      B(IK)=0.
      DO 11 I=3, II2
      IK=IK+1
      A(IK,1)=DY*DY
      IC(IK,1)=IK-1
      A(IK,2)=-2.*(DX*DX+DY*DY)
      IC(IK,2)=IK
      A(IK,3)=DY*DY
      IC(IK,3)=IK+1
      A(IK,4)=DX*DX
      IC(IK,4)=IK+II2
      A(IK,5)=DX*DX
      IC(IK,5)=N-J1+I
      INZ(IK)=5
      B(IK)=0.
11  CONTINUE
      IK=IK+1
      A(IK,1)=DY*DY
      IC(IK,1)=IK-1
      A(IK,2)=-2.*(DX*DX+DY*DY)
      IC(IK,2)=IK
      A(IK,3)=DX*DX
      IC(IK,3)=IK+II2
      A(IK,4)=DX*DX
      IC(IK,4)=J2+II-1
      A(IK,5)=DY*DY
      IC(IK,5)=J2+II+1

```

```

INZ(IK)=5
E(IK)=0.
JJ2=JJ-2
DO 13 J=3, JJ2
IK=IK+1
A(IK,1)=DX*DX-----
IC(IK,1)=IK-II2
A(IK,2)=-2.*(DX*DX+DY*DY)
IC(IK,2)=IK
A(IK,3)=DY*DY
IC(IK,3)=IK+1
A(IK,4)=DX*DX
IC(IK,4)=IK+II2
A(IK,5)=DY*DY
IC(IK,5)=N-J+2
INZ(IK)=5
E(IK)=0.
DO 12 I=3, II2
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=IK-II2
A(IK,2)=DY*DY
IC(IK,2)=IK-1
A(IK,3)=-2.*(DX*DX+DY*DY)
IC(IK,3)=IK
A(IK,4)=DY*DY
IC(IK,4)=IK+1
A(IK,5)=DX*DX
IC(IK,5)=IK+II2
INZ(IK)=5
B(IK)=0.
12 CONTINUE
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=IK-II2
A(IK,2)=DY*DY
IC(IK,2)=IK-1
A(IK,3)=-2.*(DX*DX+DY*DY)
IC(IK,3)=IK
A(IK,4)=DX*DX
IC(IK,4)=IK+II2
A(IK,5)=DY*DY
IC(IK,5)=J2+II+J-1
INZ(IK)=5
B(IK)=0.
13 CONTINUE
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=IK-II2
A(IK,2)=-2.*(DX*DX+DY*DY)
IC(IK,2)=IK
A(IK,3)=DY*DY
IC(IK,3)=IK+1
A(IK,4)=DX*DX
IC(IK,4)=N-JJ+1
A(IK,5)=DY*DY
IC(IK,5)=N-JJ+3
INZ(IK)=5
B(IK)=0.
DO 14 I=3, II2
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=IK-II2

```

```

A(IK,2)=DY*CY
IC(IK,2)=IK-1
A(IK,3)=-2.*(DX*DX+DY*DY)
IC(IK,3)=IK
A(IK,4)=DY*DY
IC(IK,4)=IK+1
A(IK,5)=DX*DX
IC(IK,5)=N-JJ+I-3
INZ(IK)=5
B(IK)=0.
14 CONTINUE
IK=IK+1
A(IK,1)=DX*DX
IC(IK,1)=IK-II2
A(IK,2)=DY*DY
IC(IK,2)=IK-1
A(IK,3)=-2.*(DX*DX+DY*DY)
IC(IK,3)=IK
A(IK,4)=DY*DY
IC(IK,4)=J2+II+JJ-2
A(IK,5)=CX*DX
IC(IK,5)=J2+II+JJ
INZ(IK)=5
B(IK)=0.
IK=IK+1
A(IK,1)=XHY1+XK/CY
IC(IK,1)=IK
A(IK,2)=-XK/CY
IC(IK,2)=N
INZ(IK)=2
B(IK)=XHY1*TY1
III=II-1
DO 15 I=2,III
IK=IK+1
A(IK,1)=-XK/DY
IC(IK,1)=I-1
A(IK,2)=XHY1+XK/DY
IC(IK,2)=IK
INZ(IK)=2
B(IK)=XHY1*TY1
15 CONTINUE
IK=IK+1
A(IK,1)=XHY1+XK/CY
IC(IK,1)=IK
A(IK,2)=-XK/DY
IC(IK,2)=IK+1
INZ(IK)=2
B(IK)=XHY1*TY1
JJ1=JJ-1
DO 16 J=2,JJ1
IK=IK+1
A(IK,1)=-XK/DX
IC(IK,1)=(J-1)*II2
A(IK,2)=XHX2+XK/DX
IC(IK,2)=IK
INZ(IK)=2
B(IK)=XHX2*TX2
16 CONTINUE
IK=IK+1
A(IK,1)=-XK/DY
IC(IK,1)=IK-1
A(IK,2)=XHY2+XK/DY
IC(IK,2)=IK

```

```

INZ(IK)=2
B(IK)=XHY2*TY2
DO 17 I=2, I11
IK=IK+1
A(IK,1)=-XK/DY
IC(IK,1)=J2-I+2
A(IK,2)=XHY2+XK/DY
IC(IK,2)=IK
INZ(IK)=2
B(IK)=XHY2*TY2
17 CONTINUE
IK=IK+1
A(IK,1)=XHY2+XK/DY
IC(IK,1)=IK
A(IK,2)=-XK/DY
IC(IK,2)=IK+1
INZ(IK)=2
B(IK)=XHY2*TY2
I=J2-I12+1
DO 18 J=2, J11
IK=IK+1
A(IK,1)=-XK/DX
IC(IK,1)=I-I12*(J-2)
A(IK,2)=XHX1+XK/DX
IC(IK,2)=IK
INZ(IK)=2
B(IK)=XHX1*TX1
18 CONTINUE
EPS=10.**(-7)
IS=2
IS=1
IF=N
NZM=0
NZM=1
CALL SPAMA1(N, EPS, IS, NZM, IF)
WRITE(6, 303) NZM
303 FORMAT(/, 10X, 'NUM MAX DE ELEMENTOS =', I4)
K=0
106 L=0
WRITE(6, 101)
101 FORMAT('1', 22X, 'TEMPERATURE DISTRIBUTION', ///)
105 IF(L.EQ.51) GO TO 106
K=K+1
L=L+1
K2=4*K
K1=K2-3
IF(K2.GT.N) K2=N
WRITE(6, 102) (I, X(I), I=K1, K2)
102 FORMAT(10X, 4('T(', I4, ')=' , F6.1, 3X))
IF(K2.LT.N) GO TO 105
STCP
END

```

ERRATA

<u>PAG.</u>	<u>LINHA</u>	<u>ONDE SE LÊ</u>	<u>LEIA-SE</u>
5	9	eq. (II-2)	exp. (II-2)
12	26	difícil	difícil
14	20	seu não utilização	sua não utilização
19	11	da posição J2	da posição J2
20	3	a qual	às quais
22	16	em linha	na linha
24	7	a qual	às quais
52	8	difícil	difícil