



Escola Politécnica - EPBC



31200053415

BT/PEF-8921

PEFMAT - RELATÓRIOS DE SUBROTINAS
MATEMÁTICAS (III)

(*) Paulo de Mattos Pimenta

(**) Priscila Goldenberg

(*) Professor Associado

(**) Professora Doutora

(recebido em 05/12/89)

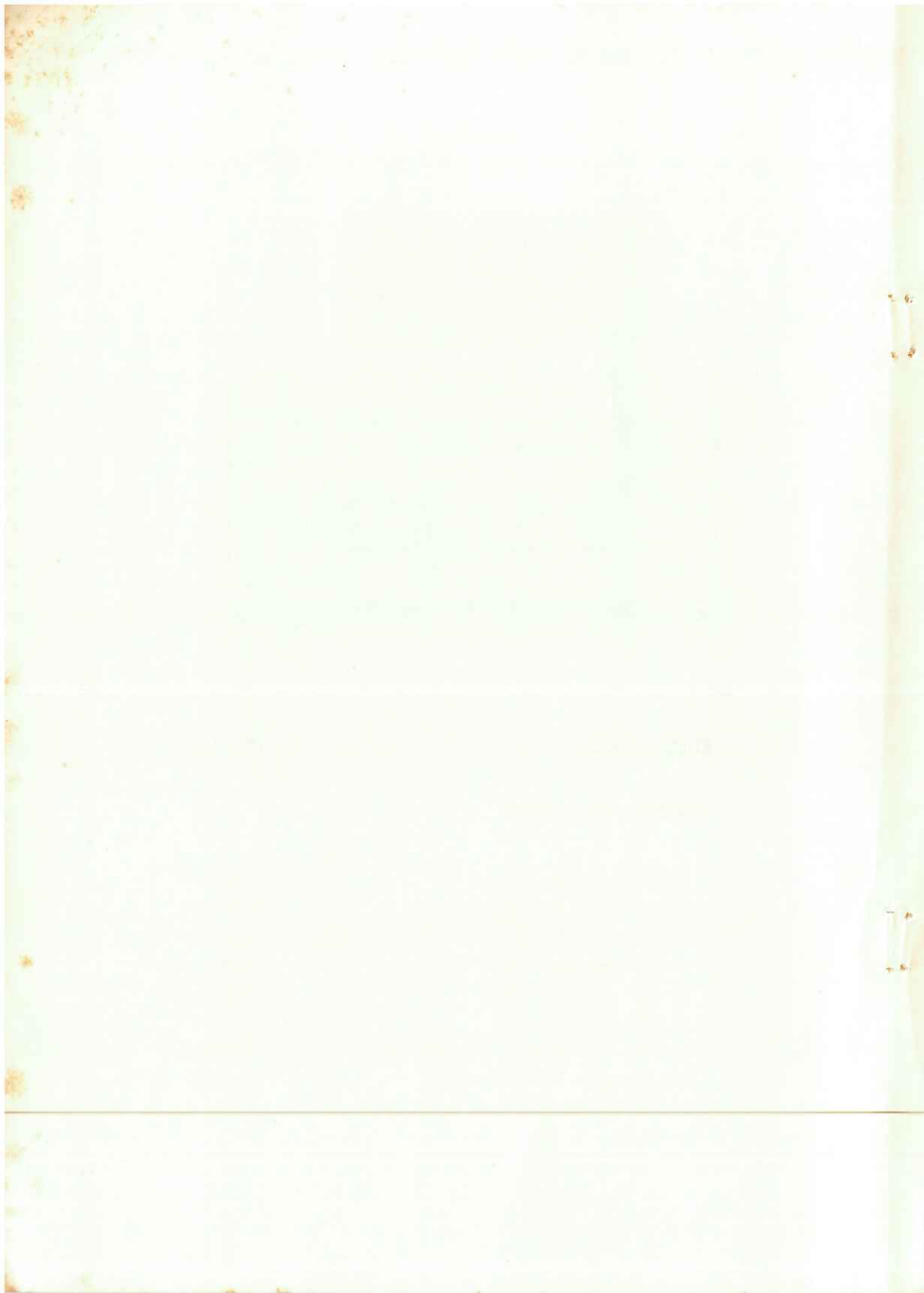
EDITOR CHEFE

P.M.Pimenta

COMISSÃO EDITORIAL

- Engenharia de Solos	W.Hachich
- Estruturas de Concreto	P.B.Fusco
- Estruturas Metálicas e de Madeira	P.B.Fusco
- Interação Solo-Estrutura	C.E.M.Maffei
- Mecânica Aplicada	P.M.Pimenta
- Métodos Numéricos	J.C.André
- Pontes e Grandes Estruturas	D.Zagottis
- Teoria das Estruturas	V.M.Souza Lima

852862



PEFMAT - RELATÓRIOS DE SUBROTINAS MATEMÁTICAS (III)

Priscila Goldenberg
Prof^a Doutora

Paulo de Mattos Pimenta
Prof. Associado

RESUMO

Um novo conjunto de subrotinas matemáticas do PEFMAT é apresentado.

ABSTRACT

A set of mathematical subroutines for the pack PEFMAT is presented.

1. Comentários

Este boletim apresenta mais um conjunto de subrotinas matemáticas do pacote PEFMAT em desenvolvimento no LMC do Departamento de Engenharia de Estruturas e Fundações.

Estão incluídas subrotinas de operações matriciais como também de solução de sistemas de equações lineares. As subrotinas deste boletim manipulam matrizes retangulares, diagonais e, principalmente, as simétricas, armazenadas em diversas formas, conforme as normas do boletim BT/PEF-8917.



2. Índice

<u>Nome</u>	<u>Parâmetros</u>	<u>Págs.</u>
SPMP	(AP, N, BP, C, LDC)	3
SBMB	(AB, N, BB, C, LDC)	4
SRMRT	(A, LDA, M, N, B, LDB, L, C, LDC)	5
SRTMR	(A, LDA, M, N, B, LDB, L, C, LDC)	6
SRCTD	(A, LDA, M, N, B, R, LDR)	7
SRIGE2	(A, LDA, N, R, LDR, IERR)	8
SPIGE	(A, N, B, R, IERR)	9
SBCH	(A, N, IB, EPS, U, IERR)	10
SBCHSL	(U, N, IB, Y, X)	11
SRQLE	(A, NM, N, PREC, D, E, E2)	12
SYGE	(Y, IY, NEQ, VKGS, VKGD, KLD, IERR)	13
SYGESL	(Y, IY, VFG, NEQ, VKGS, VKGD, KLD)	14
SRYGES	(NEQ, VKGS, VKGD, VKGI, VEG, KLD, IERR)	15
IROOT	(ROOT, XADJ, ADJNCY, MASK, NLVL, XLS, LS)	16
IYGED	(ROOT, XADJ, ADJNCY, MASK, DEG, CCSIZE, LS)	17
IYOTI	(NEQNS, XADJ, ADJNCY, PERM, MASK, XLS)	18
IYRCM	(ROOT, XADJ, ADJNCY, MASK, PERM, CCSIZE, DEG)	19
SRHS	(A, LDA, N, INT, Z)	20
SRNOR1	(A, LDA, N, M, V, SRNOR)	21
SRNORI	(A, LDA, N, M, V, SRNOR)	22
SVNOR1	(V, N, SASUM)	23
SVNOR2	(V, N, SNOR)	24
SRNORS	(A, LDA, N, PREC, D, E, E2, SRNOR, IERR)	25



Subrotina SPMP

Objetivo: Multiplicar duas matrizes packed simétricas $C = AP * BP$.

Utilização: CAL SPMP (AP, N, BP, C, LDC).

Argumentos: AP = vetor real de entrada, contendo os elementos da matriz AP.

N = constante inteira de entrada, contendo a ordem das matrizes AP, BP e CP.

BP = vetor real de entrada, contendo os elementos da matriz BP.

C = matriz real de saída, de dimensão NXN, contendo a operação produto.

LDC = constante inteira de entrada, contendo o número de linhas de C no dimensionamento do programa que chama SPMP. Deve-se tomar cuidado que $LDC \geq N$.

Subrotinas requeridas: IPADR.



Subrotina SBMB

Objetivo: Multiplicar duas matrizes banda simétricas AB e BB;
 $C = AB * BB.$

Utilização: CALL SBMB (AB, N, BB, C, LDC).

Argumentos: AB = vetor real de entrada, contendo os elementos da matriz AB.

N = constante inteira de entrada, contendo a ordem da matriz AB.

IB = constante inteira de entrada, contendo a semi-largura da banda.

BB = vetor real de entrada, contendo os elementos da matriz BB.

C = matriz real de saída, de dimensão NXN, contendo a operação produto.

LDC = constante inteira de entrada, contendo o número de linhas de C no dimensionamento do programa que chama SBMB. Deve-se tomar cuidado que $LDC \geq N$.

Subrotinas requeridas: IBADR.



Subrotina SRMRT

Objetivo: Efetuar a operação matricial $C = A * B^T$

Utilização: CALL SRMRT (A, LDA, M, N, B, LDB, L, C, LDC)

Argumentos: A = matriz real de entrada com M linhas e N colu
nas

LDA = constante inteira de entrada, contendo o nú
mero de linhas de A declarado no dimensiona
mento do programa que chama SRMRT. Deve-se
tomar cuidado que $LDA \geq M$

M = constante inteira de entrada, contendo o nú
mero de linhas de A e C

N = constante inteira de entrada, contendo o nú
mero de colunas de A (que é igual ao número
de linhas de B^T)

B = matriz real de entrada com L linhas, e N co
lunas

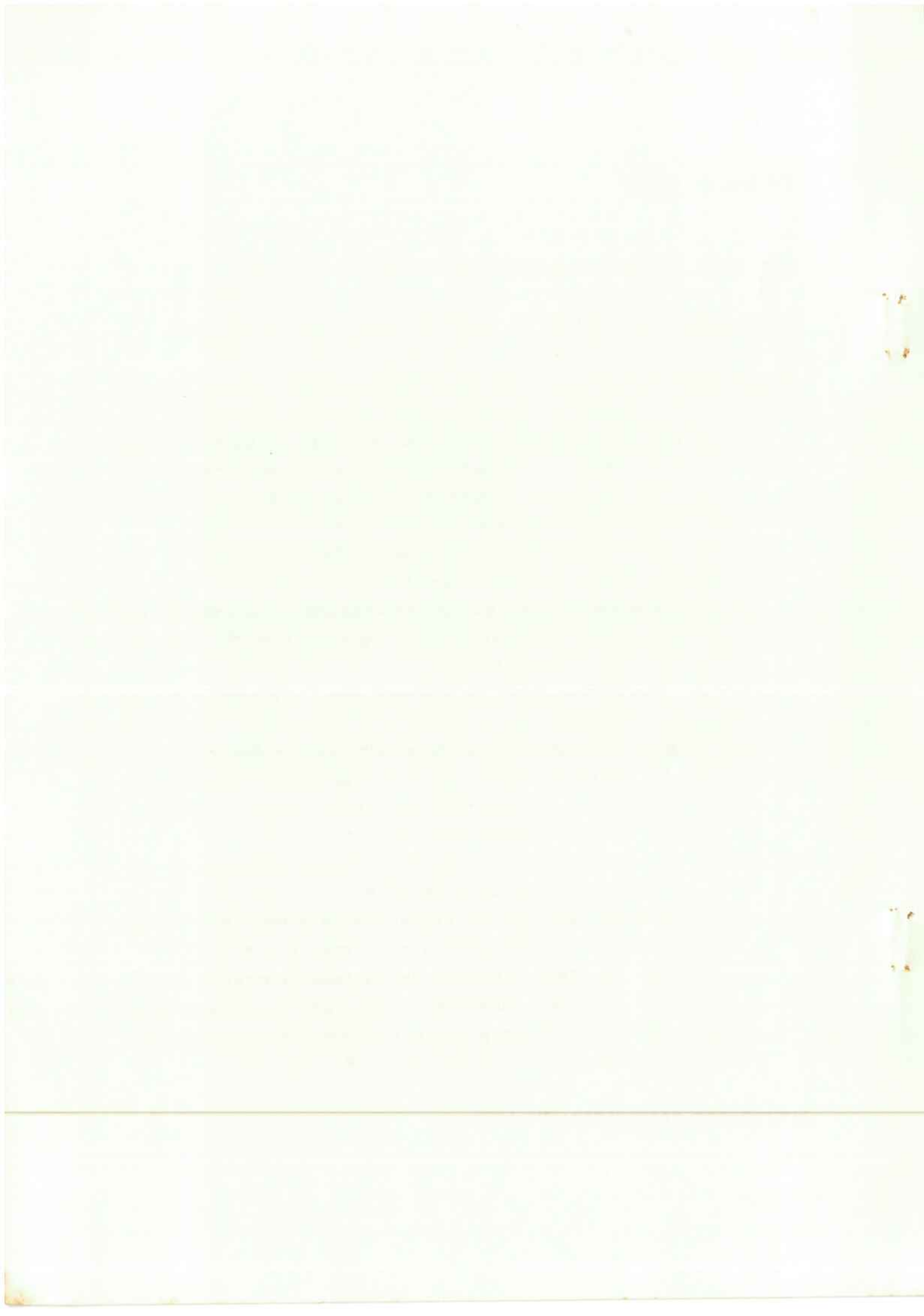
LDB = constante inteira de entrada, contendo o nú
mero de linhas de B declarado no dimensiona
mento do programa que chama SRMRT. Deve-se
tomar cuidado que $LDB \geq N$

L = constante inteira de entrada, contendo o nú
mero de colunas de B^T e C

C = matriz real de saída, com M linhas e L colu
nas, contendo a matriz produto $A * B$.

LDC = constante inteira de entrada, contendo o nú
mero de linhas de C declarado no dimensiona
mento do programa que chama SRMRT. Deve-se
tomar cuidado que $LDC \geq M$

Subrotinas requeridas: Nenhuma



Subrotina SRTMR

Objetivo: Efetuar a operação matricial $C = A^T * B$.

Utilização: CALL SRTMR (A, LDA, M, N, B, LDB, L, C, LDC).

Argumentos: A = matriz real de entrada com M linhas e N colunas

LDA = constante inteira de entrada, contendo o número de linhas de A declarado no dimensionamento do programa que chama SRTMR. Deve-se tomar cuidado que $LDA \geq M$

M = constante inteira de entrada, contendo o número de linhas de A e B

N = constante inteira de entrada, contendo o número de colunas de A e de linhas de C

B = matriz real de entrada com M linhas e L colunas

LDB = constante inteira de entrada, contendo o número de linhas de B declarado no dimensionamento do programa que chama SRTMR. Deve-se tomar cuidado que $LDB \geq M$

C = matriz real de saída, com M linhas, N colunas, contendo a operação efetuada.

LDC = constante inteira de entrada, contendo o número de linhas de C declarado no dimensionamento do programa que chama SRTMR. Deve-se tomar cuidado que $LDC \geq M$.

Subrotinas requeridas: Nenhuma.



Subrotina SRCTD

Objetivo: Efetuar a transformação consequente $R = A^T * B * A$ onde A é uma matriz retangular com M linhas e N colunas e B é diagonal

Utilização: CALL SRCTD (A, LDA, M, N, B, R, LDR).

Argumentos: A = matriz real de entrada com M linhas e N colunas

LDA = constante inteira de entrada, contendo o número de linhas de A declarado no dimensionamento do programa que chama SRCTD. Deve-se tomar cuidado que $LDA \geq M$

M = constante inteira de entrada, contendo o número de linhas de A

N = constante inteira de entrada, contendo o número de colunas de A

B = vetor real de entrada, contendo os elementos da mat^{ria} diagonal

R = matriz real de saída com N linhas e N colunas contendo a matriz $A^T * B * A$

LDR = constante inteira de entrada, contendo o número de linhas de R declarado no dimensionamento do programa que chama SRCTD. Deve-se tomar cuidado que $LDR \geq N$.

Subrotinas requeridas: Nenhuma

Subrotina SRIGE2

Objetivo: Efetuar a inversão de uma matriz real não simétrica pelo método de eliminação de Gauss, sem pivotação.

Utilização: CALL SRIGE2 (A, LDA, N, R, LDR, IERR).

Argumentos: A = matriz real de entrada com N linhas e N colunas

LDA = constante inteira de entrada, contendo o número de linhas de A declarado no dimensionamento do programa que chama SRIGE2. Deve-se tomar cuidado que $LDA \geq N$

N = constante inteira de entrada, contendo a ordem da matriz A

R = matriz real de saída, contendo a matriz inversa

LDR = constante inteira de entrada, contendo o número de linhas de RR declarado no dimensionamento do programa que chama SRIGE2. Deve-se tomar cuidado que $LDR \geq N$

IERR = constante inteira de saída, se a matriz é não singular, IERR = 0. Caso contrário IERR = K, onde K indica o elemento da diagonal que se anulou. Neste caso o programa é interrompido.

Subrotinas requeridas: Nenhuma

Subrotina SPIGE

Objetivo: Inverte uma matriz real simétrica armazenada na forma packed simétrica utilizando o método de eliminação de Gauss (sem pivotação): $R = A^{-1}$

Utilização: CALL SPIGE (A, N, B, R, IERR)

Argumentos: A = vetor real de entrada contendo os elementos da matriz A

N = constante inteira de entrada contendo a ordem da matriz

B = matriz real de dimensão N por N, utilizada como área de trabalho

R = vetor real de saída contendo a matriz inversa.

IERR = constante inteira de saída. Se IERR = 0 a inversão foi efetuada; caso contrário, se IERR = k ≠ 0, a matriz é singular e o k-ésimo elemento da diagonal se anulou. Neste caso a subrotina é interrompida.

Subrotinas Utilizadas: SRTOP, SPTOR, SRIGE2.

Subrotina SBCH

Objetivo: Efetuar a decomposição de Cholesky $A = U^T U$ de uma matriz de banda simétrica definida positiva.

Utilização: CALL SBCH (A, N, IB, EPS, U, IERR).

Argumentos: A = vetor real de entrada contendo a matriz de banda A, de dimensão N e com semi-largura da banda igual a IB.

N = constante inteira de entrada, contendo a ordem da matriz

IB = constante inteira de entrada, contendo a semi-largura da banda

EPS = constante real de entrada, contendo a precisão a ser adotada.

U = vetor real de saída contendo a matriz triangular superior obtida na decomposição de Cholesky

IERR = constante inteira de saída

= 0 valor normal

= k se $A(K, K) < \epsilon$ ou se a matriz não é definida positiva. Esta é uma condição de erro na subrotina. Quando isto ocorre a subrotina é imediatamente interrompida.

Observações: 1) U pode sobrepor A.

Subrotinas requeridas: IBADR.

Subrotina SBCHSL

Objetivo: Resolução de um sistema linear simétrico definido positivo $AX = Y$, armazenado na forma de banda, utilizando os fatores obtidos em SBCH, $A = U^T U$ (decomposição de Cholesky).

Utilização: CALL SBCHSL (U, N, IB, Y, X).

Argumentos: U = vetor real de entrada, contendo a matriz triangular superior obtida na decomposição de Cholesky (SBCH).

N = constante inteira de entrada, contendo a ordem da matriz.

IB = constante inteira de entrada, contendo a semi-largura da banda.

Y = vetor real de entrada, de ordem N, contendo o 2º membro do sistema linear.

X = vetor real de saída, de ordem N, contendo a solução do sistema linear.

Subrotinas requeridas: IBADR.

Subrotina SRQLE

Objetivo: Determinar todos os autovalores de uma matriz real simétrica utilizando o método QL.

Utilização: CALL SRQLE (A, NM, N, PREC, D, E, E2).

Argumentos: A = matriz real de entrada, de dimensão N por N ,
cujos autovalores deseja-se determinar

NM = constante inteira de entrada, contendo o número de linhas de A declarado no dimensionamento programa que chama SRQLE

N = constante inteira de entrada, contendo a ordem da matriz simétrica A. Deve-se tomar cuidado que $NM \geq N$.

PREC = constante real de entrada, contendo a precisão a ser considerada.

D = vetor real de saída, contendo os autovalores de A em ordem crescente

E = vetor real de dimensão pelo menos N, utilizado como área de trabalho.

E2 = vetor real de dimensão pelo menos N, utilizado como área de trabalho.

Subrotina requerida: Nenhuma.

Subrotina SYGE

Objetivo: Triangularizar uma matriz simétrica pelo método de eliminação de Gauss, quando a matriz está armazenada na forma skyline.

Utilização: CALL SYGE (Y, IY, NEQ, VKGS, VKGD, KLD, IERR).

Argumentos: Y = vetor real de entrada contendo os elementos da matriz, armazenados na forma skyline (coluna por coluna, de cima para baixo, do primeiro elemento não-nulo até a diagonal inclusive). Na saída contém os elementos da matriz triangularizada.

IY = vetor inteiro de entrada, contendo os endereços dos elementos da diagonal da matriz. Na saída, contém os endereços dos elementos da diagonal da matriz triangularizada

NEQ = constante inteira de entrada, contendo a ordem da matriz

VKGS = vetor real de dimensão igual a Y, utilizado como área de trabalho

VKGD = vetor real de dimensão NEQ, utilizado como área de trabalho

KLD = vetor real de dimensão NEQ + 1, utilizado como área de trabalho

IERR = constante inteira de saída. Se IERR = 0 a triangularização ocorreu normalmente. Se IERR = K $\neq$ 0, então o programa foi interrompido pois houve um elemento nulo na diagonal K.

Subrotinas requeridas: SVTMV.



Subrotina SYGESL

Objetivo: Resolver um sistema linear simétrico $Ax = b$, onde A está triangularizada pelo método de Gauss (SYGE), e armazenada na forma Skyline.

Utilização: CALL SYGESL (Y, IY, VFG, NEQ, VKGS, VKGD, KLD)

Argumentos: Y = vetor real de entrada contendo os elementos da matriz triangularizada, armazenados na forma skyline (matriz obtida em SYGE)

IY = vetor inteiro de entrada, contendo os endereços dos elementos da diagonal da matriz triangularizada

VFG = vetor real de entrada, de dimensão NEQ, o 2º membro do sistema. Na saída, contém a solução do sistema

NEQ = constante inteira de entrada, contendo a ordem da matriz

VRGS = vetor real de dimensão igual a Y, utilizado como área de trabalho

VKGD = vetor real de dimensão NEQ, utilizado como área de trabalho

KLD = vetor real de dimensão NEQ + 1, utilizado como área de trabalho.

Subrotinas requeridas: SVTMV.



Subrotina SRYGES

Objetivo: Resolução de um sistema linear $Ax = b$, não simétrico pelo método de eliminação de Gauss. A matriz A está armazenada na forma skyline.

Utilização: CALL SRYGES (NEQ, VKGS, VKGD, VKGI, VFG, KLD, IERR)

Argumentos: NEQ = constante inteira de entrada, contendo a ordem do sistema

VKGS = vetor real de entrada, contendo a parte superior de A, armazenados de cima para baixo

VKGD = vetor real de entrada, contendo os elementos da diagonal de A

VKGI = vetor real de entrada, contendo a parte inferior de A, armazenada da esquerda para a direita

VFG = vetor real de entrada, contendo o 2º membro do sistema. Na saída, contém a solução do sistema

KLD = vetor real de entrada, contendo o endereço da posição mais alta não nula em cada coluna

IERR = constante inteira de saída. Se IERR = 0, o programa terminou normalmente. Se IERR = K $\neq$ 0, o programa foi interrompido, pois o elemento $a(k, k) = 0$.

Observações: A teoria relativa a este método pode ser encontrada em Dhatt, G. and Touzot G. - "The finite element Method Displayed", John Wiley e Sons, 1984.

Subrotinas requeridas: SVTMV.



Subrotina IROOT

Objetivo: Determinar um nó inicial, a ser utilizado a seguir no algoritmo de Cuthull McKee reverso (ver observações).

Utilização: CALL IROOT (ROOT, XADJ, ADJNCY, MASK, NLVL, XLS , LS).

Argumentos: ROOT = constante inteira que na entrada, define um nó arbitrário e na saída, contém o nó inicial a ser utilizado

XADJ = vetor inteiro de entrada contendo o endereço do início de cada coluna na lista de incidências ADJNCY. A dimensão de XADJ deve ser NEQNS + 1 de modo que em XADJ(NEQNS + 1) - XADJ(NEQNS) contereá indicação de quantos elementos possui a última coluna da matriz de incidência

ADJNCY = vetor inteiro de entrada contendo em lista sequencial a matriz de incidência. A dimensão de ADJNCY deve ser igual ao tamanho do envelope

MASK = vetor inteiro de dimensão NEQNS, utilizado como área de trabalho

NLVL = constante inteira de saída contendo o número de estruturas geradas

XLS = vetor inteiro de saída de dimensão NEQNS , contendo as estruturas geradas

LS = vetor inteiro de saída, contendo o número de elementos das estruturas geradas.

Observações: A teoria sobre o algoritmo de Cuthull McKee reverso pode ser encontrada em: George, A. & Liu, J.W.H. - "Computer Solution of Large Sparse Positive Definite Systems". Prentice Hall, 1981.

Subrotinas requeridas: Nenhuma.



Subrotina IYDEG

Objetivo: Determinar o grau dos nós de uma componente grafo.

A teoria sobre o algoritmo utilizado encontra-se em (1) nas observações.

Utilização: CALL IYDEG (ROOT, XADJ, ADJNCY, MASK, DEG, CCSIZE, LS).

Argumentos: ROOT = constante inteira de entrada indicando o nó inicial para a ordenação

XADJ = vetor inteiro de entrada contendo o endereço de início de cada coluna na lista de incidências ADJNCY. A dimensão de XADJ deve ser $NEQNS + 1$ (onde NEQNS é a ordem da matriz de incidência)

ADJNCY = vetor inteiro de entrada contendo em lista sequencial a matriz de incidência. A dimensão de ADJNCY deve ser igual do tamanho do envelope

MASK = vetor inteiro de entrada de dimensão NEQNS especificando os nós do grafo que pertencem à componente a ser considerada. Se $MASK(K) = 0$, o nó k do grafo não será considerado.

DEG = vetor inteiro de saída, contendo os graus dos nós

CCSIZE = constante inteira de saída, contendo o tamanho da componente do grafo percorrido.

LS = vetor inteiro utilizado como área de trabalho.

Observações: (1) A teoria sobre a determinação dos graus, pode ser encontrada em: George, A. & Liu, J.W.H. - "Computer solution of Large Sparse Positive Definite Systems", Prentice Hall, 1981.

Subrotinas requeridas: Nenhuma.



Subrotina IYOTI

Objetivo: Efetuar uma reordenação dos nós de um grafo de modo a diminuir o tamanho do envelope de armazenamento da matriz de incidência associada ao grafo. Consideraremos o caso em que esta matriz é simétrica e o algoritmo utilizado é o de Cuthull-McKee reverso (ver referência nas observações).

Utilização: CALL IYOTI (NEQNS, XADJ, ADJNCY, PERM, MASK, XLS)

Argumentos: NEQNS = constante inteira de entrada, contendo o número de nós do grafo.

XADJ = vetor inteiro de entrada contendo o endereço do início de cada coluna na lista de incidências ADJNCY. A dimensão de XADJ deve ser NEQNS + 1 de modo que em XADJ (NEQNS + 1) - XADJ (NEQNS) contera indicação de quantos elementos possui a última coluna da matriz de incidência.

ADJNCY = vetor inteiro de entrada contendo em lista sequencial a matriz de incidência. A dimensão de ADJNCY deve ser igual ao tamanho do envelope.

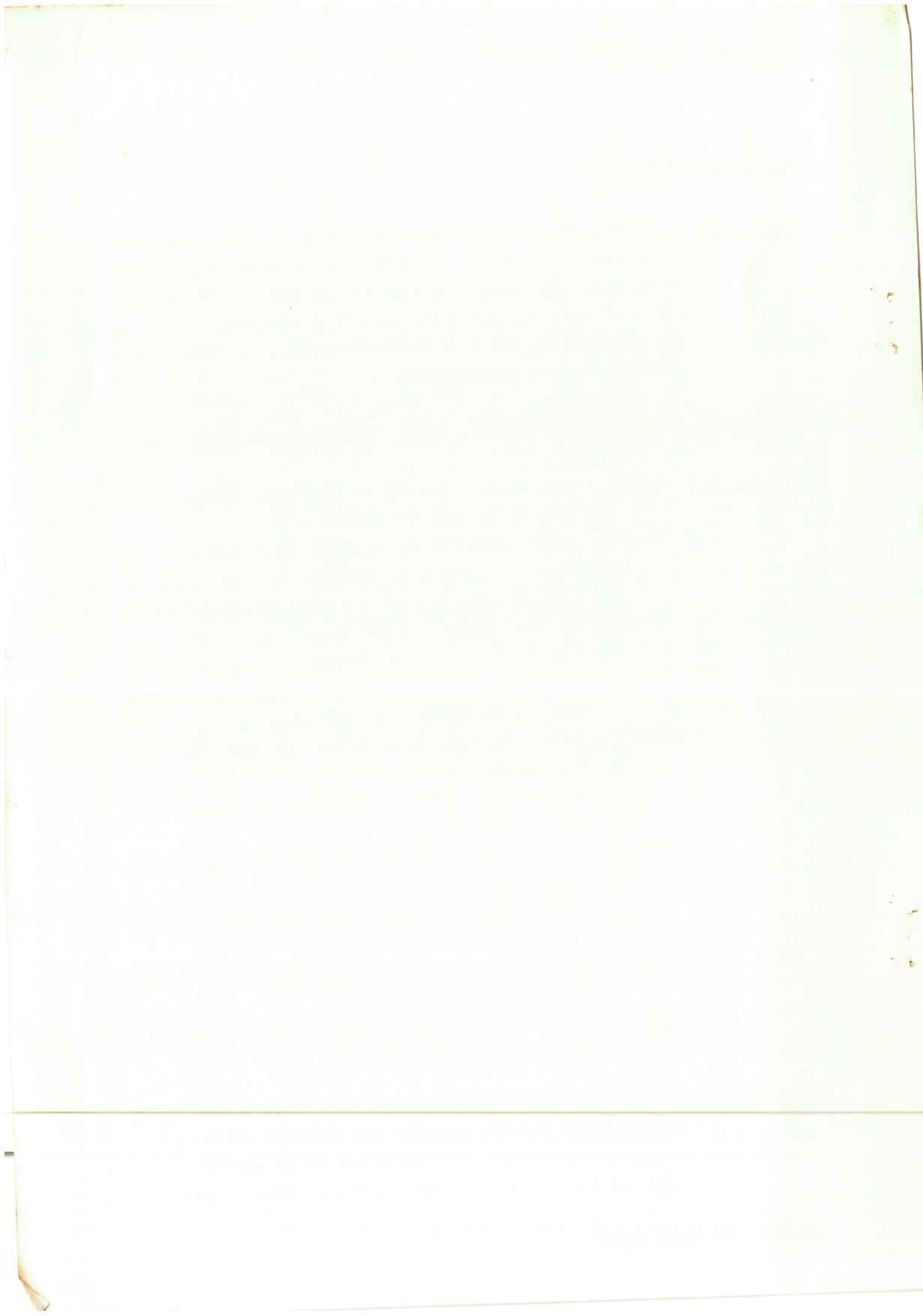
PERM = vetor inteiro de saída, contendo a reordenação dos nós, isto é, se $perm(i) = j$ o nó original j será o nó i na nova ordenação.

MASK = vetor inteiro de dimensão NEQNS, utilizado como área de trabalho.

XLS = vetor inteiro de dimensão NEQNS, utilizado como área de trabalho.

Observações: A teoria sobre o algoritmo de Cuthull McKee reverso pode ser encontrada em: George, A. & Liu, J.W.H. - "Computer Solution of Large Sparse Positive Definite Systems". Prentice Hall, 1981.

Subrotinas requeridas: IROOT, IYRCM.



Subrotina IYRCM

Objetivo: Efetuar uma reordenação dos nós de um grafo a partir do nó inicial ROOT, de modo a medir o tamanho do envelope de armazenamento da matriz de incidência associada ao grafo. Consideraremos o caso em que esta matriz é simétrica e o algoritmo utilizado é o Cuthull-McKee Reverso (ver referência nas observações).

Utilização: CALL IYRCM (ROOT, XADJ, ADJNCY, MASK, PERM, CCSIZE, DEG).

Argumentos: ROOT = constante inteira de entrada indicando o nó inicial para a ordenação.

XADJ = vetor inteiro de entrada contendo o endereço de início de cada coluna na lista de incidências ADJNCY. A dimensão de XADJ deve ser $NEQNS + 1$ (onde NEQNS é a ordem da matriz de incidência).

ADJNCY = vetor inteiro de entrada contendo em lista sequencial a matriz de incidência. A dimensão de ADJNCY deve ser igual ao tamanho do envelope.

MASK = vetor inteiro de dimensão NEQNS, utilizado como área de trabalho.

PERM = vetor inteiro de saída, contendo a reordenação dos nós, isto é, se $perm(i) = j$, o nó original j será o nó i na nova reordenação.

CCSIZE = variável inteira de saída, contendo o tamanho do grafo numerado por IYRCM.

DEG = vetor inteiro de dimensão NEQNS, utilizado como área de trabalho.

Observações: A teoria sobre o algoritmo de Cuthull McKee reverso pode ser encontrada em: George, A. & Liu, J. W.H. - "Computer solution of Large Sparse Positive Definite Systems", Prentice Hall, 1981.

Subrotinas requeridas: IYDEG



Subrotina SRHS

Objetivo: Reduzir uma matriz real geral à forma de Hessenberg superior, utilizando transformações elementares. As transformações são também acumuladas.

Utilização: CALL SRHS (A, LDA, N, INT, Z).

Argumentos: A = matriz real de entrada de ordem N, ser reduzida à forma de Hessenberg superior. Na saída, A contém a matriz superior de Hessenberg e na parte inferior, contém os multiplicadores usados na redução.

LDA = constante inteira de entrada, contendo o número de linhas de A e Z declarado no dimensionamento do programa que chama SRHS. Deve-se tomar cuidado que $LDA \geq N$.

N = constante inteira de entrada, contendo a ordem das matrizes A e Z.

INT = vetor inteiro de saída de dimensão pelo menos N, identificando as permutações efetuadas nas linhas e colunas durante a redução.

Z = matriz real de saída de ordem N contendo a matriz transformação produzida durante a redução, isto é, se $Z^{-1}AZ$ possui a forma de Hessenberg.

Subrotinas requeridas: Nenhuma.



Subrotina SRNOR1

Objetivo: Calcular a norma infinito de uma matriz A

$$\|A\|_1 = \max_j \left(\sum_{i=1}^N |a_{ij}| \right)$$

Utilização: CALL SRNOR1 (A, LDA, N, M, V, SRNOR)

Argumentos: A = matriz real de entrada com N linhas e M colunas

LDA = constante inteira de entrada contendo o número de linhas declarado no dimensionamento da matriz A no programa que chama SRNOR1. Deve-se cuidar para que $LDA \geq N$.

N = constante inteira de entrada contendo o número de linhas de A

M = constante inteira de entrada contendo o número de colunas de A

V = vetor real de dimensão N utilizando como área de trabalho

SRNOR = constante inteira de saída, contendo o valor da norma.

Subrotinas requeridas: SVIMAX.

Subrotina SRNORI

Objetivo: Calcular a norma infinito de uma matriz A $\|A\|_{\infty} =$

$$= \max_i \left(\sum_{j=1}^M |a_{ij}| \right)$$

Utilização: CALL SRNORI (A, LDA, N, M, V, SRNOR)

Argumentos: A = matriz real de entrada com N linhas e M colunas

LDA = constante inteira de entrada contendo o número de linhas declarado no dimensionamento da matriz A no programa que chama SRNORI. Deve-se tomar cuidado para que $LDA \geq N$.

N = constante inteira de entrada contendo o número de linhas de A

M = constante inteira de entrada contendo o número de colunas de A

V = vetor real de dimensão N utilizado como área de trabalho

SRNOR = constante inteira de saída, contendo o valor da norma.

Subrotinas requeridas: SVIMAX.

Subrotina SVNOR1

Objetivo: Calcular a norma 1 de um vetor (soma dos valores ab solutos).

Utilização: CALL SVNOR1 (V, N, SASUM).

Argumentos: V = vetor real de entrada, contendo os elementos do vetor V.

N = constante inteira de entrada, contendo a di mensão do vetor V.

SASUM = constante real de saída, contendo a norma 1.

Subrotinas requeridas: Nenhuma.



Subrotina SVNOR2

Objetivo: Calcular a norma 2 de um vetor (raiz quadrada positiva do produto escalar).

Utilização: CALL SVNOR2 (V, N, SNOR).

Argumentos: V = vetor real de entrada, contendo os elementos do vetor V.

N = constante inteira de entrada, contendo a di mensão do vetor V.

SNOR = constante real de saída, contendo a norma 2.

Subrotinas requeridas: SVTMV.



Subrotina SRNORS

Objetivo: Determinar a norma espectral de uma matriz A real simétrica.

$$\|A\|_S = \left| \frac{\lambda_{\max}}{\lambda_{\min}} \right|,$$

onde max e min são respectivamente os autovalores máximos e mínimos de A.

Utilização: CALL SRNORS (A, LDA, N, PREC, D, E, E2, SRNOR , IERR)

Argumentos: A = matriz real de entrada, de dimensão N por N, cuja norma deseja-se determinar.

LDA = constante inteira de entrada, contendo o número de linhas de A declarado no dimensionamento do programa que chama SRQLE

N = constante inteira de entrada, contendo a ordem da matriz simétrica A. Deve-se tomar cuidado que $LDA \geq N$.

PREC = constante real de entrada, contendo a precisão a ser considerada

D = vetor real de saída, contendo os autovalores de A em ordem crescente

E = vetor real de dimensão pelo menos N, utilizado como área de trabalho

E₂ = vetor real de dimensão pelo menos N, utilizado como área de trabalho

SRNOR = constante real de saída, contendo o valor da norma espectral

IERR = constante inteira de saída. Se IERR = 0 o programa terminou normalmente. Se IERR = 1, o programa foi interrompido pois $\lambda_{\min} = 0$.

Subrotinas requeridas: SRQLE

