

**SISTEMA BUGRE PARA OTIMIZAÇÃO COM RESTRIÇÕES
LINEARES: MANUAL TÉCNICO**

Hermínio Simões Gomes

RELATÓRIO TÉCNICO Nº 43/87

RESUMO. BUGRE é um sistema para resolver problemas de Otimização com restrições lineares de grande porte. Usa menos memória que MINOS e parece ser mais estável numericamente. Essencialmente, usa uma técnica de condicionamento periódico das bases. Este relatório é o Manual do Usuário do sistema.

Universidade Estadual de Campinas
Instituto de Matemática, Estatística e Ciência da Computação
IMECC – UNICAMP
Caixa Postal 6065
13.081 - Campinas, SP
BRASIL

O conteúdo do presente Relatório Técnico é de única responsabilidade do autor.

Setembro – 1987

1. INTRODUÇÃO

Apresentamos neste manual uma descrição sobre o uso do
do sistema EASY e seus parâmetros.

INDICE

As explicações detalhadas sobre o funcionamento interno
do sistema de EASY são dadas no primeiro volume do livro.

A subrotina EASY permite-se para resolver o seguinte
problema:

Resolver $E(x)$, $E = A^T A$ e A linear ou não-linear

1. INTRODUÇÃO	1
2. DESCRIÇÃO DOS PARÂMETROS	3
2.1. Áreas em COMMON	8
3. ESTRUTURA ESPARSA POR LINHAS	11
4. EXEMPLOS	12
4.1. Exemplo 1	16
4.2. Exemplo 2	20
4.3. Exemplo 3	28
5. DIAGNÓSTICOS DE ERROS	32
6. LISTAGEM DO PROGRAMA FONTE	34

O usuário deverá fornecer a subrotina para o cálculo
da função e gradientes da seguinte forma:

SYNOPSIS: REAL *E, X, F, G, H, D

INTEGER N, M, L, I, J, K, L, M, N, P, Q, R, S, T, U, V, W, X, Y, Z

REAL *A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, S, T, U, V, W, X, Y, Z

1. INTRODUÇÃO

Apresentamos neste manual uma descrição sobre o uso da sub-rotina BUGRE e seus propósitos.

As explicações detalhadas sobre o funcionamento interno das rotinas da BUGRE são dadas no primeiro volume do texto.

A sub-rotina BUGRE presta-se para resolver o seguinte problema :

Minimizar $f(x)$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ linear ou não-linear
 f diferenciável e contínua

Sujeita a

$$Ax \begin{cases} = \\ \leq \\ \geq \end{cases} b$$

$$l \leq x \leq u$$

$$x, l, u \in \mathbb{R}^n$$

$$b \in \mathbb{R}^m$$

$$A(m \times n)$$

Ela usa decomposição ortogonal esparsa e faz preconditionamento periódico dos sistemas lineares. Trabalha em precisão dupla, real * 8, e consta, além de suas sub-rotinas internas, de duas sub-rotinas externas, que são as rotinas DATAE e SAIDA. A sub-rotina DATAE elabora um arquivo com os dados de entrada do problema, prestando-se, portanto, para fazer uma verificação dos dados. A sub-rotina SAIDA faz um bonito relatório de saída de resultados.

O usuário deverá fornecer a sub-rotina para o cálculo de função e gradiente da seguinte forma :

```
SUBROUTINE FUN (N, X, F, G, MODO)
IMPLICIT REAL * 8 (A-H, O-Z)
REAL * 8 X(N), G(N), F
```

INTEGER MODO

cálculo de F e G

RETURN

END

A variável F deverá sair com o valor da função no ponto X, G deverá sair com as componentes do gradiente de F em X. A variável MODO serve para o controle da primeira chamada da sub-rotina FUN, é uma variável de entrada e se for igual a 1 é a primeira chamada, e zero caso contrário.

Esta variável serve para controle da leitura de um arquivo, passagem de uma variável ou atribuição ao vetor G.

No caso de o problema ser linear, o vetor G é inalterado durante todo o processo de resolução do problema.

Nas seções posteriores veremos a descrição dos parâmetros, diversos exemplos resolvidos e um exemplo de como fazer a estrutura esparsa por linhas da matriz tecnológica A.

A rotina BUGRE foi totalmente escrita em VAX-FORTRAN. Não foi feito uso de estrutura em blocos, podendo-se, portanto, adaptá-lo com relativa facilidade a outros computadores.

2. DESCRIÇÃO DOS PARÂMETROS

A sub-rotina BUGRE possui 47 parâmetros e pode ser chamada da seguinte forma :

CALL BUGRE (M, N, A, COM, QUEM, AS, COMAS, QUEMAS, R, COMR, QUEMR, B, IQ, IV, NDIM, NDIMV, X, Y, L, U, D, DTIL, G, GFI, F, ICONJ, IRATIV, INDT, STATUS, ICNA, NA, NCNA, ITMAX, ITMG, ITOT, EPS, ICHAVE, NMAX, ISGN, AUX, AUX1, AUX2, AUX3, AUX4, AUX5, V, IER)

onde

- M - número de restrições do problema não incluindo as canalizações. (e)
- N - número de variáveis do problema. (e)
- A - vetor REAL * 8 com os elementos não-nulos da matriz de tecnologia. Os elementos devem estar dispostos por linha. (e)
- COM e QUEM - vetor de inteiros com a estrutura esparsa por linha da matriz de tecnologia. COM contém os índices dos i-ésimos elementos não-nulos que começam cada linha da matriz de tecnologia. QUEM contém as colunas correspondentes aos elementos não-nulos da referida matriz.
COM deve ter dimensão M+1. A última posição é ocupada com o valor do índice do último elemento de A mais uma unidade. Assim o vetor A conterá exatamente COM(M+1) - 1 elementos.
O vetor QUEM deve ter a dimensão do vetor A, ou seja NE = COM(M+1) - 1.
- AS - vetor auxiliar com as mesmas características de A. Deve apenas ser dimensionado.

- COMAS e QUEMAS - vetores de inteiros com as mesmas características de COM e QUEM. Devem apenas ser dimensionados.
- R - vetor REAL * 8 auxiliar que será usado para armazenar o fator esparsos R da decomposição ortogonal $QA = R$, usado no condicionamento do sistema linear. Deve ser dimensionado fazendo-se uma estimativa do tamanho de uma base e na esparsidade da matriz de tecnologia. Uma estimativa razoável pode ser a mesma dimensão de A. A variável NDIM deve conter a dimensão de R.
- COMR e QUEMR - vetores de inteiros auxiliares para a rotina preparar a estrutura esparsa por linhas para R. A sub-rotina ESTR que faz a estrutura de R avisará se a dimensão de QUEMR for insuficiente. A dimensão de COMR deve ser igual ao de uma base da matriz de tecnologia, portanto $N+1$ será suficiente.
- B - vetor REAL * 8 com o termo independente das restrições (RHS). Deve ter dimensão no mínimo M. (e)
- IQ e IV - vetores de inteiros auxiliares de dimensão NDIMV. Dimensionando-se N será suficiente.
- NDIMV - dimensão de IQ e IV. (e)
- NDIM - dimensão de R. (e)
- X - vetor REAL * 8 que deve conter, na entrada, uma solução factível (não obrigatoriamente básica) do problema. Esta solução inicial poderá ser, inclusive, um ponto dentro do convexo. Deve ser N-dimensional.

Y - vetor auxiliar REAL * 8 que conterá a transformação linear induzida pela matriz das restrições ativas sobre X. Deve ser N-dimensional.

L e U - vetores REAL * 8 que deverão conter os limites inferiores e superiores das variáveis. Considera-se que todas as variáveis são canalizadas. Se alguma variável não for canalizada pode-se atribuir, por exemplo, $L = -1 \cdot D + 6$ e $U = +1 \cdot D + 6$ para esta variável. Ambos os vetores L e U devem ter dimensão N.

D - vetor auxiliar REAL * 8 de dimensão N.

DTIL - mesmas especificações de D.

G - vetor REAL * 8 que na saída conterá o vetor gradiente na solução. Deve ter dimensão N. (s)

GFI - vetor auxiliar REAL * 8 que na saída conterá o gradiente da função transformada (função em Y). As componentes de GFI são os multiplicadores de Lagrange na solução. Deve ter dimensão N. (s)

F - variável REAL * 8 que na saída conterá o valor da função objetivo na variável X.

ICONJ - vetor auxiliar de inteiros de dimensão N.

IRATIV - ~~mesmas especificações de ICONJ. Dê-se M+N~~
Vetor auxiliar de inteiros de dimensão M+N

INDT - vetor auxiliar de inteiros que na saída conterá os índices das restrições de trabalho na última iteração. Obedece a seguinte partição :

1	NA	
	Restrições gerais	Canalizações
	NA + 1	N

STATUS - vetor de inteiros, INTEGER*2, que na saída conterá o estado de cada componente da solução com relação a sua correspondente canalização.

$$\text{STATUS}(i) = \begin{cases} -1 & \text{está no limite inferior} \\ 0 & \text{está livre} \\ 1 & \text{está no limite superior} \end{cases}$$

Deve ter dimensão N. (s)

ICNA - vetor auxiliar de inteiros que na saída conterá os índices das canalizações não ativas na solução. Deve ter dimensão N.

NA - variável inteira que na saída conterá o número de restrições gerais (não incluindo canalizações) ativas. (s)

NCNA - variável inteira que na saída conterá o número de canalizações não-ativas na solução. (s)

ITMAX - variável inteira, de entrada, que deve conter o máximo permitido de chamadas a sub-rotina MRGESP. (e)

ITMG - variável inteira que na saída conterá o número de chamadas a sub-rotina MRGESP.

ITOT - variável inteira que na saída conterá o número de atualizações feitas no vetor X (não incluindo atualizações parciais de busca unidimensional). (s)

EPS - variável REAL*8, de entrada, que deve conter uma tolerância para o cálculo de resíduos. (e)

ICHAVE - variável inteira, de entrada, que determina de que modo deve ser feito o condicionamento. Se ICHAVE = 0 o condicionamento será sempre

feito; caso contrário o precondicionamento será feito cada vez que o número de iterações nos sistemas lineares atingirem um teto de NMAX iterações. (e)

NMAX - veja ICHAVE. Se NMAX for setado em zero, o programa assume NMAX = N. (e)

ISGN - vetor de inteiros, INTEGER * 2, que na entrada deve conter os sinais referentes as restrições do problema

$$ISGN(i) = \begin{cases} -1 & \text{se a restrição } i \text{ for de menor ou igual} \\ 0 & \text{se a restrição } i \text{ for de igual} \\ 1 & \text{se a restrição } i \text{ for de maior ou igual} \end{cases}$$

Deve ter dimensão M. (e)

AUX, AUX1, AUX2, AUX3, AUX4, AUX5, V - são vetores auxiliares, REAL * 8, de dimensão N, usados como área de trabalho. Devem apenas ser dimensionados.

IER - código de erro.

Se IER = 0 - tudo correu normalmente e o ótimo foi encontrado.

IER = 1 - a variável IPARE, vinda de um COMMON no programa principal foi setada em 1 e a melhora na função de uma iteração para outra, dentro da sub-rotina MRGESP, foi insuficiente.

IER = 2 - foi atingido o máximo de ITMAX chamadas permitidas a sub-rotina MRGESP.

2.1. Áreas em COMMON

O Programa de chamada deverá conter as seguintes declarações de COMMON :

```
COMMON /BASE/ NAMED
COMMON /NCFUN/ NCFG
COMMON /OPCAO/ IOP, IPARE, IPRI      (obrigatório)
COMMON /FAIXA/ NF                     (obrigatório)
COMMON /TEMPO/ TEMPO
COMMON /MEMO/ MEM
COMMON /DEV/ IN, IOUT                 (obrigatório)
COMMON /RELAT/ SAI                    (obrigatório)
COMMON /FASE/ IFASE, NVAR
```

NAMED - conterá na saída o tamanho médio da base usada no problema em questão.

NCFG - conterá o número total de chamadas a sub-rotina FUN.

IOP - variável de entrada

IOP = 1 - função objetivo LINEAR

IOP = 2 - função objetivo NÃO-LINEAR

IPARE - variável de entrada

IPARE = 0 - a sub-rotina MRGESP seguirá seu curso normalmente mesmo que a diferença na função objetivo de uma iteração para outra seja pequena.

IPARE = 1 - determina a saída da sub-rotina MRGESP quando a diferença na função

objetivo for pequena demais de uma iteração para outra.

Normalmente optamos por $IPARE = 0$, a opção $IPARE = 1$ determina a saída em uma região próxima a uma região de estacionariedade se não houverem restrições ativas.

- IPRI - Se $IPRI = 1$ a função F e o vetor X serão listados no terminal a cada iteração. Se $IPRI = 0$ não serão listados.
- NF - variável de entrada, atribui-se o número de faixas que devem ser usados no condicionamento. NF deve ser no mínimo 2. Se atribuirmos $NF < 2$ será assumido $NF = 2$. O número de faixas depende da estrutura esparsa da matriz de tecnologia. Se NF for maior que uma das bases do problema, a correção será automática, portanto, pode-se atribuir a NF um valor grande.
- TEMPO - serve para o caso de se usar contagem do tempo de CPU e poder passar este dado para a rotina de saída dos resultados, SAÍDA.
- MEM - como no caso do TEMPO, a variável MEM conterá o total de memória usada pelas variáveis dimensionadas do programa.
- IN, IOU - números dos arquivos de leitura de dados e saída de dados respectivamente.
- SAI - variável lógica, se $SAI = .TRUE.$ então o relatório de saída será abundante de dados, se $SAI = .FALSE.$, apenas alguns dados irão para o arquivo de saída.

IFASE, NVAR → usadas para se passar a fase do problema, 1 ou 2, para a sub-rotina FUN. A variável NVAR indica o número de variáveis do problema em questão.

Realiza-se operações por IFASE = 0, a opção IFASE = 1 determina a saída em uma região próxima a uma região de características se não houverem restrições ativas.

Se IFASE = 1 a função F e o vetor X serão lidos, que no terminal a cada iteração. Se IFASE = 0 não serão lidos.

variável de controle, atribui-se o número de iterações que devem ser feitas no procedimento. Se deve ser no mínimo 1. Se atribuímos NF = 1 será assumido NF = 5. O número de iterações depende da estrutura esparsa da matriz de tecnologia. Se NF for maior que um das passadas do problema, a contagem será automática, portanto, pode-se atribuir a NF um valor grande.

serve para o caso de se usar contador de tempo de CPU a poder passar este dado para a rotina de saída dos resultados, SAIDA.

como no caso do TEMPO, a variável MEM controla o total de memória usada pelas variáveis dimensionadas no programa.

números dos arquivos de leitura de dados e saída de dados respectivamente.

variável lógica, se SAI = .TRUE. então é relatório de saída será abundante de dados, se SAI = .FALSE., apenas alguns dados irão para o arquivo de saída.

3. ESTRUTURA ESPARSA POR LINHAS

A matriz tecnológica do problema a ser resolvido deve ser fornecida em uma estrutura esparsa por linhas, da seguinte maneira :

- A - vetor com os elementos não-nulos da matriz tecnológica dispostos por linha.
- COM - vetor de inteiros que deve conter os índices dos i-ésimos elementos que começam cada linha.
- QUEM - vetor de inteiros que deve conter os índices das colunas dos elementos de A.

Por exemplo, seja a seguinte matriz :

$$\begin{bmatrix} 0 & 0 & 7.3 & 0 & 2.8 \\ 1.3 & 0 & 0 & 2.4 & 5.3 \\ 0 & 0 & 5.8 & 0 & 0 \\ 0 & 4.3 & 0 & 2.9 & 0 \end{bmatrix}$$

então

$$A = (7.3, 2.8, 1.3, 2.4, 5.3, 5.8, 4.3, 2.9)$$

$$COM = (1, 3, 6, 7, 9^*)$$

$$QUEM = (3, 5, 1, 4, 5, 3, 2, 4)$$

O elemento 9 é incluído em COM, para efeito de controle de laço no acesso a última linha. Este elemento de controle sempre é igual ao número de elementos em A mais uma unidade.

4. EXEMPLOS

Mostramos, a seguir, alguns exemplos para o uso do programa. Elaborou-se um único programa principal para rodar os exemplos, modificando-se apenas as sub-rotinas de cálculo de função e gradiente, FUN, e os arquivos de dados de entrada.

Utilizamos um arquivo de dados de nome DADOS.DAT para entrada. Este arquivo foi criado seguindo-se a seguinte ordem :

NOME1	
N, M, IOP	
COM	
QUEM	fase 1
A	
B	
L, U	
X	
NOME2	
NVAR	
COM	fase 2
QUEM	
A	
ISGN	

Onde NOME1 é o nome do problema na fase 1 e NOME2 é o nome na fase 2. NVAR é o número de variáveis na fase 2. Os primeiros COM e QUEM que aparecem, se referem a estrutura da matriz na fase 1, os últimos se referem a estrutura na fase 2. Durante a fase 1 o vetor de sinais ISGN é preenchido com zeros, indicando restrições de igualdade. Na fase 2 o vetor é lido com os sinais das restrições do problema original.

No caso de haver somente fase única podemos dispor o arquivo DADOS.DAT do seguinte modo :

NOME
 N, M, IOP
 COM
 QUEM
 A
 B
 L, U
 X
 ISGN

A seguir mostramos a listagem do programa principal usado para rodar os exemplos.

1:	C		60
2:	C	ARQUIVO PPEDBL.FOR	
3:	C		C
4:	C	PROGRAMA PRINCIPAL	C
5:	C		C
6:		IMPLICIT REAL*8 (A-H,O-Z)	
7:		REAL*8 NOME	C
8:		REAL*8 A(2000),AS(2000),R(2000)	C
9:		REAL*8 X(5000),Y(5000),L(5000),U(5000)	C
10:		REAL*8 B(5000),D(5000),DTIL(5000),G(5000),GFI(5000)	
11:		1 ,AUX(5000),AUX1(5000),AUX2(5000),AUX3(5000),	
12:		2 ,AUX4(5000),AUX5(5000),V(5000)	
13:		REAL*4 TEMPO	
14:		INTEGER*4 DCPU1,P1	
15:		INTEGER COM(1000),QUEM(2000)	
16:		INTEGER COMAS(1000),QUEMAS(2000)	20
17:		INTEGER COMR(1000),QUEMR(2000)	
18:		INTEGER ICONJ(5000),IRATIV(5000),INDT(5000),ICNA(5000)	40
19:		INTEGER IQ(2000),IV(2000)	
20:		INTEGER*2 ISGN(5000),STATUS(5000)	
21:		LOGICAL SAI	C
22:		COMMON/TEMPO/ TEMPO	C
23:		COMMON/DEV/IN, IOUT	C
24:		COMMON/OPCAD/ IOP,IPARE,IPRI	
25:		COMMON/FASE/IFASE,NVAR	C
26:		COMMON/FAIXA/ NF	C
27:		COMMON/RELAT/ SAI/MEMO/ MEM	C
28:		DATA NDIM,NDIMV/2000,2000/	C
29:	C		
30:	C	DADOS DE ENTRADA DO PROBLEMA	
31:	C		
32:		DATA IN,IOUT,IPARE,IPRI,EPS,GR,ITMAX,NF	
33:		1 /20,30,0,0,0,1.D-6,1.E+10,1000,10/	20
34:		SAI=.TRUE.	
35:		IENT= 5	
36:		ISAI= 6	
37:	C		
38:	C	LEITURA DE DADOS	40
39:	C		60
40:		OPEN (UNIT=IN,FILE='DADOS.DAT',STATUS='OLD')	
41:		READ (IN,20) NOME	
42:	20	FORMAT(A8)	C
43:		READ(IN,*) N,M,IOP	C
44:		IF(M.EQ.0) GOTO 40	C
45:		READ(IN,*) (COM(J),J=1,M+1)	
46:		NQ=COM(M+1)-1	C
47:		READ (IN,*) (QUEM(J),J=1,NQ)	C
48:		READ (IN,*) (A(J),J=1,NQ)	C
49:		READ (IN,*) (B(J),J=1,M)	C
50:	40	DO 60 J=1,N	
51:	60	READ(IN,*) L(J),U(J)	
52:		READ(IN,*) (X(J),J=1,N)	
53:	C		
54:		CALL DATAE(N,M,COM,QUEM,A,B,ISGN,L,U,X,NOME)	
55:		WRITE(ISAI,*) ' DIGITE 1 PARA FASE UNICA'	
56:		READ(IENT,*) IR	
57:		IF (IR.EQ.1) GOTO 140	20
58:	C		
59:	C		
60:		IF(M.EQ.0) GOTO 100	


```

61:      DO 80 J=1,M
62: 80      ISGN(J)=0
63: 100      IFASE=1
64:      WRITE(ISAI,*) 'ENTROU NA FASE 1'
65:      CALL LIB$INIT_TIMER(DCPU1)
66:      CALL BUGRE(M,N,A,COM,QUEM,AS,COMAS,QUEMAS,R,COMR,QUEMR,
67: 1      B,IQ,IV,NDIM,
68: &      NDIMV,X,Y,L,U,D,DTIL,G,GFI,F,ICONJ,IRATIV,INDT,STATUS,ICNA,NA,
69: &      NCNA,ITMAX,ITMG,ITOT,EPS,ICHAVE,NMAX,ISGN,AUX,AUX1,AUX2,AUX3,
70: &      AUX4,AUX5,V,IER)
71:      CALL LIB$STAT_TIMER(2,P1,DCPU1)
72:      TEMPO= P1/100.
73: C
74:      WRITE(ISAI,*) ' SAIDA DA FASE 1'
75:      WRITE(ISAI,*) ' VETOR   X  DA FASE 1'
76:      DO 120 J=1,N
77: 120      WRITE(ISAI,*) X(J)
78:      WRITE(ISAI,*)
79:      WRITE(ISAI,*) 'VALOR DA FUNCAO NA SAIDA DA FASE 1 = ',F
80:      CALL SAIDA (NOME,N,M,X,F,G,GFI,NA,STATUS,INDT,ICNA,NCNA,
81: 1      ITMG,ITOT,IER)
82: C
83: C      ENTRADA DE DADOS PARA FASE 2
84: C
85:      READ(IN,20) NOME
86:      READ(IN,*)  NVAR
87:      N=NVAR
88:      READ(IN,*) (COM(J),J=1,M+1)
89:      NQ=COM(M+1)-1
90:      READ(IN,*) (QUEM(J),J=1,NQ)
91:      READ (IN,*) (A(J),J=1,NQ)
92: 140      IF(M.EQ.0) GOTO 160
93:      READ (IN,*) (ISGN(J),J=1,M)
94:      CALL DATAE(N,M,COM,QUEM,A,B,ISGN,L,U,X,NOME)
95: C
96: C
97: 160      IFASE=2
98:      WRITE(ISAI,*) '      ENTROU NA FASE 2 '
99: C
100:      CALL LIB$INIT_TIMER(DCPU1)
101:      CALL BUGRE(M,N,A,COM,QUEM,AS,COMAS,QUEMAS,R,COMR,QUEMR,
102: 1      B,IQ,IV,NDIM,
103: &      NDIMV,X,Y,L,U,D,DTIL,G,GFI,F,ICONJ,IRATIV,INDT,STATUS,ICNA,NA,
104: &      NCNA,ITMAX,ITMG,ITOT,EPS,ICHAVE,NMAX,ISGN,AUX,AUX1,AUX2,AUX3,
105: &      AUX4,AUX5,V,IER)
106:      CALL LIB$STAT_TIMER(2,P1,DCPU1)
107:      TEMPO= P1/100.
108: C
109:      CALL SAIDA(NOME,N,M,X,F,G,GFI,NA,STATUS,INDT,ICNA,NCNA,
110: 1      ITMG,ITOT,IER)
111: C
112:      CLOSE(UNIT=IN)
113:      STOP
114:      END

```

4.1. Exemplo 1

$$\text{Minimizar } f(x) = x_1 \log x_2 + (x_3 - x_1)^2$$

$$\text{Sujeito a } 2x_1 + 5x_2 = 19$$

$$-4x_2 + 8x_3 \geq 20$$

$$7x_1 + 2x_2 + 6x_3 \leq 55$$

$$1 \leq x_1 \leq 3$$

$$2 \leq x_2 \leq 5$$

$$0 \leq x_3$$

Começando com o ponto inicial $x^0 = (2, 3, 5)$ observe-se que x^0 não é básico.

Os resultados obtidos são mostrados a seguir, bem como os arquivos de dados e arquivo para cálculo de função e gradiente.

C
C
C

ARQUIVO FUNLOG.FOR

```

SUBROUTINE FUN(N,X,F,G,MODE)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(N),G(N)
XLOG=LOG(X(2))
G(1)=XLOG - 2.*(X(3)-X(1))
G(2)=X(1)/X(2)
G(3)= 2.*(X(3)-X(1))
F=X(1)*XLOG + (X(3)-X(1))**2
RETURN
END

```

EXEMP1

```

3,3,2
1,3,5,8
1,2,2,3,1,2,3
2.,5.,-4.,8.,7.,2.,6.
19.,20.,55.
1.,3.
2.,5.
0.,1.05
2.,3.,5.
0,1,-1

```

***** RESULTADOS DO PROBLEMA EXEMP1 ***** PAGINA

INDICE	VECTOR X	STATUS	GRAD(X)	GRAD(Y)
1	0.300000D+01	SUPERIOR	-0.644489D+00	0.390769D+00
2	0.260000D+01	LIVRE	0.115385D+01	0.200000D+00
3	0.380000D+01	LIVRE	0.160000D+01	-0.142603D+01

VALOR DA FUNCAO OBJETIVO = 0.350653D+01

CONJUNTO DE INDICES DAS RESTRICOES DE TRABALHO NA SOLUCAO

1 2

CONJUNTO DE INDICES DAS CANALIZACOES NAO-ATIVAS NA SOLUCAO

2 3

NUMERO DE ATUALIZACOES NO VETOR X = 2

NUMERO DE CHAMADAS A SUB-ROTINA "MRGESP" = 3

NUMERO DE VARIAVEIS = 3

NUMERO DE RESTRICOES GERAIS = 3

CODIGO DE ERRO = 0

TEMPO DE CPU (SEGUNDOS) = 0.0900

TAMANHO MEDIO DA BASE = 1

TAMANHO RELATIVO DA BASE EM
RELACAO AO PROBLEMA (X) = 11.11111

4.2. Exemplo 2

Este exemplo, devido a Nelson Maculan [22], é resolvido nas suas duas fases.

Minimizar

$$f(x) = 100x_1 + 80x_2 + 50x_3 + 40x_4 + 80x_5 + 60x_6 + 100x_7 + 80x_8$$

Sujeito a

$$x_1 + 2.5x_3 + 3.5x_5 + 5x_7 \geq 70$$

$$x_2 + 2.5x_4 + 3.5x_6 + 5x_8 \geq 80$$

$$x_3 + x_4 \leq 10$$

$$x_5 + x_6 = 15$$

$$x_7 + x_8 \leq 8$$

$$x_i \geq 0 \quad i = 1, \dots, 8$$

Preparando a fase 1 para este problema, temos :

Minimizar

$$f_1(x) = x_{10} + x_{12} + x_{14} = \text{soma das variáveis artificiais}$$

Sujeito a

$$x_1 + 2.5x_3 + 3.5x_5 + 5x_7 - x_9 + x_{10} = 70$$

$$x_2 + 2.5x_4 + 3.5x_6 + 5x_8 - x_{11} + x_{12} = 80$$

$$x_3 + x_4 + x_{13} = 10$$

$$x_5 + x_6 + x_{14} = 15$$

$$x_7 + x_8 + x_{15} = 8$$

$$x_i \geq 0 \quad i = 1, \dots, 15$$

Temos um problema com 15 variáveis. A seguir mostramos os arquivos de dados de entrada, o arquivo para função e gradiente e ainda os resultados obtidos.

C
C
C

ARQUIVO FUNMACULAN.FOR

```

SUBROUTINE FUN(N,X,F,G,MOD0)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(1),G(1),C(8)
COMMON/FASE/IFASE,NVAR
DATA C/100.,80.,50.,40.,80.,60.,100.,80./
IF(IFASE.EQ. 1) GOTO 90
IF(MOD0.EQ.0) GOTO 10
DO 1 J=1,N
G(J)=C(J)

```

1
C
10

```

F=0.
DO 2 J=1,N
F=F+C(J)*X(J)
RETURN

```

2

C

C

PARTE REFERENTE A FASE 1

C
90

```

IF(MOD0.EQ.0) GOTO 20
DO 3 J=1,N

```

3

```

G(J)=0.
G(10)=1.
G(12)=1.
G(14)=1.

```

20

```

F=X(10)+X(12)+X(14)
RETURN
END

```

***** RESULTADOS DO PROBLEMA MACULF1 ***** PAGINA

INDICE	VECTOR X	STATUS	GRAD(X)	GRAD(Y)
1	0.358508D+02	LIVRE	0.000000D+00	0.000000D+00
2	0.183051D+00	LIVRE	0.000000D+00	0.000000D+00
3	0.708000D+01	LIVRE	0.000000D+00	0.000000D+00
4	0.200000D+01	LIVRE	0.000000D+00	0.000000D+00
5	0.487603D+01	LIVRE	0.000000D+00	0.000000D+00
6	0.101240D+02	LIVRE	0.000000D+00	0.000000D+00
7	-0.277556D-16	INFERIOR	0.000000D+00	0.100000D+01
8	0.800000D+01	LIVRE	0.000000D+00	0.100000D+01
9	0.616949D+00	LIVRE	0.000000D+00	0.100000D+01
10	0.000000D+00	INFERIOR	0.100000D+01	0.000000D+00
11	0.616949D+00	LIVRE	0.000000D+00	0.000000D+00
12	0.000000D+00	INFERIOR	0.100000D+01	0.000000D+00
13	0.920000D+00	LIVRE	0.000000D+00	0.000000D+00
14	0.000000D+00	INFERIOR	0.100000D+01	0.000000D+00
15	0.000000D+00	INFERIOR	0.000000D+00	0.000000D+00

VALOR DA FUNCAO OBJETIVO = 0.000000D+00

CONJUNTO DE INDICES DAS RESTRICOES DE TRABALHO NA SOLUCAO

1 2 3 4 5

CONJUNTO DE INDICES DAS CANALIZACOES NAO-ATIVAS NA SOLUCAO

5 8 3 6 1 4 2 9 11 13

NUMERO DE ATUALIZACOES NO VETOR X = 6

NUMERO DE CHAMADAS A SUB-ROTINA "MRGESP" = 7

NUMERO DE VARIAVEIS = 15

NUMERO DE RESTRICOES GERAIS = 5

CODIGO DE ERRO = 0

TEMPO DE CPU (SEGUNDOS) = 0.2400

TAMANHO MEDIO DA BASE = 5

TAMANHO RELATIVO DA BASE EM
RELACAO AO PROBLEMA (%) = 33.33334

***** RESULTADOS DO PROBLEMA MACULF2 ***** PAGINA

INDICE	VECTOR X	STATUS	GRAD(X)	GRAD(Y)
1	0.000000D+00	INFERIOR	0.100000D+03	-0.220000D+03
2	0.325000D+02	LIVRE	0.800000D+02	0.857143D+02
3	0.100000D+02	LIVRE	0.500000D+02	0.800000D+02
4	0.000000D+00	INFERIOR	0.400000D+02	-0.164286D+03
5	0.142857D+01	LIVRE	0.800000D+02	-0.328571D+03
6	0.135714D+02	LIVRE	0.600000D+02	0.142857D+02
7	0.000000D+01	LIVRE	0.100000D+03	0.428571D+01
8	0.000000D+00	INFERIOR	0.800000D+02	0.857143D+01

VALOR DA FUNCAO OBJETIVO = 0.482857D+04

CONJUNTO DE INDICES DAS RESTRICOES DE TRABALHO NA SOLUCAO

4 1 2 3 5

CONJUNTO DE INDICES DAS CANALIZACOES NAO-ATIVAS NA SOLUCAO

5 7 6 3 2

NUMERO DE ATUALIZACOES NO VETOR X = 9

NUMERO DE CHAMADAS A SUB-ROTINA "MRGESP" = 10

NUMERO DE VARIAVEIS = 8

NUMERO DE RESTRICOES GERAIS = 5

CODIGO DE ERRO = 0

TEMPO DE CPU (SEGUNDOS) = 0.3000

TAMANHO MEDIO DA BASE = 3

TAMANHO RELATIVO DA BASE EM
RELACAO AO PROBLEMA (%) = 22.50000

MACULF1

1,5,1

1,7,12,16,19,22

1,3,5,7,9,10

2,4,6,8,11,12

3,4,13

5,6,14

7,8,15

1.,2.5,3.5,5.,-1.,1.

1.,2.5,3.5,5.,-1.,1.

1.,1.,1.

1.,1.,1.

1.,1.,1.

70.,80.,10.,15.,9.

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,1.05

0.,0.,0.,0.,0.,0.,0.,0.,70.,0.,80.,10.,15.,8.

MACULF2

8

1,5,9,11,13,15

1,3,5,7,2,4,6,8,3,4,5,6,7,8

1.,2.5,3.5,5.,1.,2.5,3.5,5.,1.,1.,1.,1.,1.,1.

1,1,-1,0,-1

4.3. Exemplo 3

Resolvemos um problema não-linear sem restrições, que é a minimização da clássica função de Rosenbrock

$$\text{Minimizar } f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$$

com o ponto inicial $x^0 = (-1.2, 1)$

Neste exemplo temos $M = 0$ e as canalizações triviais.

A seguir mostramos o arquivo de dados, DADOS.DAT, a rotina do cálculo de F e G e resultados de saída.

```

C
C      ARCHIVO FUNE.FOR
C
C      SUBROUTINE FUN(N,X,F,G,MOD0)
C
C      FUNCION DE ROSENBROCK
C
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8  X(N),G(N)
C      T=X(1)*X(1)-X(2)
C      T1=X(1)-1.
C      F=100.*T*T+T1*T1
C      G(1)=400.*T*X(1)+2.*T1
C      G(2)=-200.*T
C      RETURN
C      END

```

```

ROSENBRO
2,0,2
-1.E5,1.E5
-1.E5,1.E5
-1.2,1.

```

***** RESULTADOS DO PROBLEMA ROSENBR ***** PÁGINA

INDICE	VECTOR X	STATUS	GRAD(X)	GRAD(Y)
1	0.999993D+00	LIVRE	-0.147961D-04	-0.147961D-04
2	0.999986D+00	LIVRE	0.196705D-06	0.196705D-06

VALOR DA FUNCAO OBJETIVO = 0.518598D-10

CONJUNTO DE INDICES DAS CANALIZACOES NAO-ATIVAS NA SOLUCAO

1 2

NUMERO DE ATUALIZACOES NO VETOR X = 27

NUMERO DE CHAMADAS A SUB-ROTINA "MRGESP" = 1

NUMERO DE VARIAVEIS = 2

NUMERO DE RESTRICOES GERAIS = 0

CODIGO DE ERRO = 0

TEMPO DE CPU (SEGUNDOS) = 0.3500

5. DIAGNÓSTICOS DE ERROS

Existe a possibilidade de ocorrência de erros dentro do programa BUGRE. Podem ser por impossibilidade de se conseguir um ótimo para o problema em questão, ou, o mais provável, os dados de entrada não estarem corretos.

O programa BUGRE contém uma sub-rotina para consistência dos dados de entrada, sub-rotina CONSYS. Ela examina alguns dados de entrada, faz também um cálculo numérico de gradiente e compara com o obtido analiticamente, verifica se a diferença não é tão acentuada e poderá aceitar ou não como certo. No caso de a diferença ser "grande" então a seguinte mensagem sairá no terminal :

"POSSÍVEL ERRO NA CODIFICAÇÃO DO GRADIENTE"

Deve-se, neste caso, rever a codificação da função e gradiente na sub-rotina FUN.

É ainda possível que os dados de entrada passem incólu-
mes, pela sub-rotina ECONS, e ainda hajam erros. Vejamos os ca-
sos mais comuns e suas soluções :

- 1) Erro na rotina de busca unidimensional - É quase certeza ha-
ver erro na sub-rotina FUN.
- 2) Erro de "FLOATING OVERFLOW" nas rotinas HSATA1 ou HSATA2 - O
número de faixas, NF, dado no programa principal foi insufi-
ciente para produzir uma matriz R, da decomposição ortogonal,
que seja singular. A matriz original supostamente era singu-
lar, a matriz R deveria também ser singular para poder-se de-
tectar a presença de uma base singular e fazer-se a reSeleção
das colunas para uma nova base. Neste caso, deve-se aumentar
o valor de NF.
- 3) Erro de "degeneração técnica" - É possível que a primeira es-
colha das colunas, para formar uma base, falhe. Se falhar en-
tramos na sub-rotina RESEL, que fará uma nova seleção das co-

lunas. É muito raro que as duas tentativas falhem. O modelo deve ser revisto, ou melhor, a estrutura COM e QUEM devem ser revisadas. Se as duas tentativas falharem, uma perturbação no vetor B é realizada (rotina PERTB) para tentar-se conseguir uma base.

Quatro tentativas são feitas, se não sair da degeneração técnica o programa para.

- 4) Convergência lenta demais ou divergência - Deve-se verificar a grandeza dos resíduos e o valor da variável EPS, dado de entrada. Não há ocorrência de "Zig zag" nas restrições.

... e não há necessidade de "tip test" nas testilhas.
... a primeira das testilhas é o valor da variável L25, dado 0-20
... 4) Comparação feita entre os dois testes - Deve-se verificar
... não o programa para.

LISTAGEM DO PROGRAMA FONTE

```
C
C
C      ARQUIVO BUGRE.FOR
C
C      SUBROUTINE BUGRE(M,N,A,COM,QUEM,AS,COMAS,QUEMAS,R,COMR,QUEMR,
1     B,IQ,IV,NDIM,NDIMV,X,Y,L,U,D,DTIL,G,GFI,F,ICONJ,IRATIV,
2     INDT,STATUS,ICNA
3     ,NA,NCNA,ITMAX,ITMG,ITOT,EPS,ICHAVE,NMAX,
4     ISGN,AUX,AUX1,AUX2,AUX3,AUX4,AUX5,V,IER)
C
C      PROPOSITO:   RESOLVER UM PROBLEMA DE PROGRAMACAO NAO-LINEAR
                    COM RESTRICOES LINEARES E VARIAVEIS CANALIZADAS.
C
C      MINIMIZAR    F(X)
C
C      SUJEITO A
C
C                  A*X =,<=,>= B
C
C                  L=< X <= U
C
C      O PROGRAMA UTILIZA DECOMPOSICAO ORTOGONAL ESPARSA PARA FAZER
C      PRECONDICIONAMENTO DOS SISTEMAS LINEARES. TRABALHA ,PORTANTO,
C      COM UMA FORMA BASTANTE ESTAVEL.
C
C      AUTOR: H. S. GOMES       07/1987
C      FEE- UNICAMP- CAMPINAS
C      PROJETADO PARA OPERAR COM O SISTEMA VAX/VMS
C
C      VERSAO 1.0
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 A(1),AS(1),B(1),X(N),L(N),U(N),D(N),DTIL(N),G(N),GFI(N)
C      REAL*8 Y(N),AUX(N)
C      REAL*8 AUX1(N),AUX2(N),AUX3(N),AUX4(N)
C      REAL*8 AUX5(N),V(N)
C      REAL*8 R(NDIM)
C      INTEGER MODO,ICHAVE
C      INTEGER IQ(NDIMV),IV(NDIMV)
C      INTEGER COM(1),QUEM(1),ICONJ(N),IRATIV(1),INDT(N),ICNA(N)
C      INTEGER COMR(1),QUEMR(NDIM),COMAS(1),QUEMAS(1)
C      INTEGER*2 ISGN(1),STATUS(1)
C      LOGICAL OTIMO,FAZ,DEG
C      LOGICAL ANORM
C      COMMON/BASE/ NAMED,NAMAX
C      COMMON/NAANT/ NAANT
C      COMMON/NFAT/ NFAT
C      COMMON/NCFUN/ NCFGT
C      ITELA= 6
C      NAMED=0
C      NFAT=0
C      FAZ=.TRUE.
C      DEG=.FALSE.
C      ANORM =.FALSE.
C      IPERT=0
C      ITMG=0
C      ITOT=0
C      NCFGT=0
C      IER=0
```

```

NA=0
NRIGUA=0
EPSR=EPS*1.D2
IF( NMAX.EQ.0) NMAX= N
MAX=N
CALL CONSIS(M,N,A,COM,QUEM,L,U,G,Y)

NORMALIZA "A" SE ANORM = .TRUE.

IF(ANORM) CALL ESCAL(M,N,A,COM,QUEM,B)

CALCULA FUNCAO E GRADIENTE NO PONTO INICIAL

MODO=0
SE A ROTINA DE CONSISTENCIA NAO FOR CHAMADA , DEVE-SE
COLOCAR MODO =1 NA DECLARACAO ACIMA

CALL FUN(N,X,F,G,MODO)
NCFGT=NCFGT+1
MODO=0
DO 20 J=1,N
D(J)=-G(J)
CONTINUE

COLOQUE PERMANENTEMENTE OS INDICES DAS RESTRICOES DE
IGUALDADE EM "INDT" E "IRATIV"
"NRIGUA" SERA' O NUMERO DE RESTRICOES DE IGUALDADE
ATIVAS EM "X"

NRAX=0
IF( M.EQ.0) GOTO 120
DO 60 I=1,M
IF(ISGN(I).NE.0) GOTO 60
CALL SELIND(M,N,A,COM,QUEM,B,X,EPSR,IRATIV,NRAX,I,ISGN)
INDT(NRAX)=I
CONTINUE
NRIGUA=NRAX

"NRAX" SERA' O NUMERO TOTAL DE RESTRICOES ATIVAS EM "X"

"NRGA" SERA' O NUMERO DE RESTRICOES GERAIS ATIVAS EM "X"

NRAX=NRIGUA
IF (M.EQ.0) GOTO 120

OBTENCAO DO CONJUNTO DE TRABALHO INDT
FAZENDO PRIMEIRO A SELECAO DOS INDICES DAS
RESTRICOES ATIVAS "IRATIV"

DO 100 I=1,M
IF(ISGN(I).EQ.0) GOTO 100
CALL SELIND(M,N,A,COM,QUEM,B,X,EPSR,IRATIV,NRAX,I,ISGN)
CONTINUE
NRGA=NRAX
NA=NRGA

```

```

C      NRCA" SERA" O NUMERO DE CANALIZACOES ATIVAS EM "X"
C      "NCNA" SERA" O NUMERO DE CANALIZACOES NAO-ATIVAS EM "X"
C      "NA" SERA" O NUMERO DE RESTRICOES GERAIS ATIVAS NAO
C      INCLUINDO CANALIZACOES ATIVAS
C
C      AGORA FAZENDO PARA AS CANALIZACOES
C
120    NRCA=0
      NCNA=0
      DO 160 I=1,N
      XI=X(I)
      XL=L(I)
      XU=U(I)
C
C      MAIS RIGOR NA TOLERANCIA DAS CANALIZACOES
C
      IF( XI .LT. (XL-EPS) .OR. XI.GT. (XU+EPS) ) GOTO 540
      IF( ABS(XI-XL) .LT.EPS .OR. ABS(XI-XU) .LT. EPS) GOTO 140
C
C      SE A CANALIZACAO E' NAO ATIVA ENTAO ATUALIZE O CONTADOR
C      DE CANALIZACOES NAO-ATIVAS E COLOQUE O INDICE NO CONJUNTO
C      "ICNA" DOS INDICES DAS RESTRICOES CANALIZADAS NAO-ATIVAS
C
      NCNA=NCNA+1
      ICNA(NCNA)=I
      GOTO 160
C
C      SE A CANALIZACAO E' ATIVA SEU INDICE SERA' COLOCADO
C      EM "IRATIV"
C
140    NRCA=NRCA+1
      NRAX=NRAX+1
      IRATIV(NRAX)=I
160    CONTINUE
C
C      OBTENCAO DE INDT
C
      IF(NA.EQ.0) GOTO 180
      IF( NRAX .EQ. NRIGUA) GOTO 280
      IF(NRAX.GT.N) GOTO 220
180    DO 200 J=NRIGUA+1,NRAX
200    IND(T(J))=IRATIV(J)
      IF(NA.EQ.0) GOTO 240
      GOTO 280
C
C
C
C      SE OCORREU DEGENERACAO ENTAO CHAME REFOR
220    CALL REFOR(IRATIV,INDT,ISGN,A,COM,QUEM,B,L,U,X,N,NRAX,NRGA,NA,EPS,
      & NAANT)
      IF(NAANT.EQ.0) GOTO 40
      GOTO 280
C
C
C
240    J1=1
      DO 260 J=(NRAX+1),N
      IND(T(J))=ICNA(J1)

```

```

260      J1=J1+1
C
C      GOTO 300
C
C      SE AS CANALIZACOES ATIVAS NAO FORAM SUFICIENTES
C      PARA COMPLETAR INDT ENTAO A SUB-ROTINA TRANS
C      SELECIONARA ENTRE AS CANALIZACOES NAO-ATIVAS
C      QUAIS AQUELAS QUE SERAO UTILIZADAS
280      CALL TRANS(M,N,A,COM,QUEM,INDT,NA,ICNA,NCNA,AS,COMAS,QUEMAS,NRAX,
&      FAZ,DEG,COMR,QUEMR,R,NDIM,V,IQ,IV,NDIMV,EPS,AUX)
      NAMED=NAMED+NA
      NAMAX=MAX0(NA,NAMAX)
      IF(.NOT. DEG ) GOTO 300
      CALL PERTB(M,N,B,INDT,ISGN,NA,EPS)
      IPERT=IPERT+1
      IF(IPERT.EQ.4) GOTO 520
      DEG=.FALSE.
      NAANT= NA
C
C      VA RECOMECA A ITERACAO
C
C      GOTO 80
C
C      ATRIBUA A Y AS SUAS COMPONENTES CORRESPONDENTES
300      DO 320 I=(NA+1),N
320      Y(I)=X(INDT(I))
      IF(NA.EQ.0) GOTO 360
      DO 340 I=1,NA
340      Y(I)=B(INDT(I))
C
C      CHAMA A ROTINA PRINCIPAL PARA RESOLVER
C      O PROBLEMA CANALIZADO
360      CALL MRGESP(M,N,A,COM,QUEM,B,NA,AS,COMAS,QUEMAS,R,NDIM,COMR,QUEMR,
&      G,GFI,D,DTIL,X,Y,L,U,EPS,MAX,V,AUX,AUX1,AUX2,AUX3,AUX4,AUX5,
&      INDT,ICNA,NCNA,ICONJ,IT,F,ISGN,OTIMO,KONT,IER)
      NAANT=NA
      ITOT=ITOT+IT
      WRITE(ITELA,*) 'ITOT = ',ITOT
      ITMG=ITMG+1
      IF(IER.EQ.1) GOTO 380
      IF( .NOT. OTIMO) GOTO 480
C
C      CALCULA O STATUS FINAL
380      NRAX=0
      EPSR=EPSR*1.D+1
      DO 400 I=1,M
      CALL SELIND (M,N,A,COM,QUEM,B,X,EPSR,IRATIV,NRAX,I,ISGN)
400      CONTINUE
C
      DO 460 I=1,N
      XI=X(I)
      XL=L(I)
      XU=U(I)
      IF( ABS(XI-XL).LT.EPS) GOTO 420
      IF( ABS(XI-XU).LT.EPS) GOTO 440

```



```

260      J1=J1+1
C
C
C      GOTO 300
C
C      SE AS CANALIZACOES ATIVAS NAO FORAM SUFICIENTES
C      PARA COMPLETAR INDT ENTAO A SUB-ROTINA TRANS
C      SELECIONARA ENTRE AS CANALIZACOES NAO-ATIVAS
C      QUAIS AQUELAS QUE SERAO UTILIZADAS
C
280      CALL TRANS(M,N,A,COM,QUEM,INDT,NA,ICNA,NCNA,AS,COMAS,QUEMAS,NRAX,
      & FAZ,DEG,COMR,QUEMR,R,NDIM,V,IQ,IV,NDIMV,EPS,AUX)
      NAMED=NAMED+NA
      NAMAX=MAX0(NA,NAMAX)
      IF(.NOT. DEG ) GOTO 300
      CALL PERTS(M,N,B,INDT,ISGN,NA,EPS)
      IPERT=IPERT+1
      IF(IPERT.EQ.4) GOTO 520
      DEG=.FALSE.
      NAANT= NA
C
C      VA RECOMECAR A ITERACAO
C
C      GOTO 80
C
C      ATRIBUA A Y AS SUAS COMPONENTES CORRESPONDENTES
C
300      DO 320 I=(NA+1),N
320      Y(I)=X(INDT(I))
      IF(NA.EQ.0) GOTO 360
      DO 340 I=1,NA
340      Y(I)=B(INDT(I))
C
C      CHAMA A ROTINA PRINCIPAL PARA RESOLVER
C      O PROBLEMA CANALIZADO
C
360      CALL MRGESP(M,N,A,COM,QUEM,B,NA,AS,COMAS,QUEMAS,R,NDIM,COMR,QUEMR,
      & G,GFI,D,DTIL,X,Y,L,U,EPS,MAX,V,AUX,AUX1,AUX2,AUX3,AUX4,AUX5,
      & INDT,ICNA,NCNA,ICONJ,IT,F,ISGN,OTIMO,KONT,IER)
      NAANT=NA
      ITOT=ITOT+IT
      WRITE(ITELA,*) 'ITOT = ',ITOT
      ITMG=ITMG+1
      IF(IER.EQ.1) GOTO 380
      IF(.NOT. OTIMO) GOTO 480
C
C      CALCULA O STATUS FINAL
C
380      NRAX=0
      EPSR=EPSR*1.D+1
      DO 400 I=1,M
      CALL SELIND (M,N,A,COM,QUEM,B,X,EPSR,IRATIV,NRAX,I,ISGN)
400      CONTINUE
C
      DO 460 I=1,N
      XI=X(I)
      XL=L(I)
      XU=U(I)
      IF( ABS(XI-XL).LT.EPS) GOTO 420
      IF( ABS(XI-XU).LT.EPS) GOTO 440

```



```

20 DO 20 J=COM(I),COM(I+1)-1
   S=DMAX1(DABS(A(J)),S)
   IF( S.GE.1.D+5) GOTO 80
   IS=IDINT(S)
   J=1
40 IF( IS.LE.J) GOTO 60
   J=J+J
   GOTO 40
60 S=DFLOAT(J)
C
80 IF( S.EQ.0.D0) GOTO 140
DO 100 J=COM(I),COM(I+1)-1
100 A(J)=A(J)/S
   B(I)=B(I)/S
120 CONTINUE
   RETURN
140 WRITE(ITELA,*) ' MATRIZ COM LINHA NULA '
   STOP
END

```

C
C
C

ARQUIVO MRGESPDBL.FOR

```

SUBROUTINE MRGESP(M,N,A,COM,QUEM,B,NA,AS,COMAS,QUEMAS,
1  R,NDIM,COMR,QUEMR,G,GFI,D,DTIL,X,Y,L,U,EPS,MAX,V,AUX,
2  AUX1,AUX2,AUX3,AUX4,AUX5,INDT,ICNA,NCNA,ICONJ,IT,
3  F,ISGN,OTIMO,KONT,IER)

```

C
C
C
C

PROPOSITO: RESOLVE O PROBLEMA CANALIZADO ASSOCIADO
AO PROBLEMA COM RESTRICAO GERAL.

```

IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(1),AS(1),R(NDIM),B(1)
REAL*8 G(N),GFI(N),D(N),DTIL(N),X(N),Y(N),L(N),U(N),V(N)
REAL*8 AUX(N),AUX1(N),AUX2(N),AUX3(N),AUX4(N),AUX5(N)
INTEGER COM(1),QUEM(1)
INTEGER COMAS(1),QUEMAS(1)
INTEGER COMR(1),QUEMR(NDIM)
INTEGER INDT(N),ICNA(N),ICONJ(N)
COMMON/OPCAO/IOP,IPARE,IPRI
COMMON/NCFUN/NCFGT

```

C
C
C

VETOR DE SINAL COM INTEGER*2

```

INTEGER*2 ISGN(1)
LOGICAL OTIMO

```

C
C
C

```

ITELA= 6
OTIMO= .FALSE.
KONT=0
TOL=1.D-8
EPSF=1.D-7
EPSA=EPS
IT=0
IER=0
FAUX=F

```

```

C
C      TESTE DE OTIMALIDADE APOS CALCULAR O GRADIENTE DA FUNCAO FI
C
20  CALL GRADFI(M,NA,AS,COMAS,QUEMAS,A,COM,QUEM,R,COMR,QUEMR,G,GFI,
    &  INDT,ICNA,N,AUX,V,Y,EPS,AUX1,AUX2,AUX3,AUX4,AUX5,KON,MAX)
C      A VARIÁVEL "KONT" CONTERÁ O MÁXIMO DE ITERAÇÕES
C      REALIZADAS PELAS ROTINAS DE SISTEMAS LINEARES
C      ( ROTINAS HSATA1 E HSATA2 )
C
      KONT=MAX0(KONT,KON)
40  KI=0
      KJ=N+1
      IF(NA.EQ.0) GOTO 180
      DO 160 J=1,NA
        IF (ISGN(INDT(J))) 80,60,140
60    KI=KI+1
        ICONJ(KI)=J
        GOTO 160
80    IF( GFI(J).LT.EPS) GOTO 120
100   KJ=KJ-1
        ICONJ(KJ)=J
        GOTO 160
120   KI=KI+1
        ICONJ(KI)=J
        GOTO 160
140   IF(GFI(J).GT.-EPS) GOTO 120
        GOTO 100
160   CONTINUE
180   IF(NA.EQ.N) GOTO 240
      DO 220 J=(NA+1),N
        JAUX=INDT(J)
        IF( (ABS(Y(J)-L(JAUX)) .LT. EPS .AND. GFI(J).GT.-EPS) .OR. (ABS(Y(J)-
&U(JAUX)) .LT. EPS .AND. GFI(J).LT.EPS) ) GOTO 200
        IF (ABS(GFI(J)).LT.EPS) GOTO 200
        KJ=KJ-1
        ICONJ(KJ)=J
        GOTO 220
200   KI=KI+1
        ICONJ(KI)=J
220   CONTINUE
240   IF(KI.NE.N) GOTO 280
260   OTIMO=.TRUE.
      WRITE(ITELA,*) 'SAIDA NORMAL'
      RETURN
C
C
280   IQ=0
      DO 300 J=1,N
300   D(J)=-GFI(J)
        IF(KI.EQ.0) GOTO 340
        DO 320 J=1,KI
          JAUX=ICONJ(J)
          D(JAUX)=0.
320   GFI(JAUX)=0.
340   G1=0.
        DO 360 J=1,N
360   G1=G1+GFI(J)*GFI(J)
        IF(G1.LT.EPSA) GOTO 260
C
C      CALCULO DE DTIL

```



```

DO 560 J=KJ,N
JAUX=ICONJ(J)
IF(ABS(GFI(JAUX)) .GT. EPS) GOTO 580
560 CONTINUE
580 IF(KI.EQ.0) GOTO 620
C
C PROJECAD DO GRADIENTE
C
DO 600 J=1,KI
JAUX=ICONJ(J)
600 GFI(JAUX)=0.
C
C CALCULO DA NOVA DIRECAO
C
620 G2=0.
DO 640 J=1,N
G2=G2+GFI(J)*GFI(J)
640 IF(G2.LT.EPSA*G1) GOTO 20
IQ=IQ+1
IF(IQ.EQ.N+1) GOTO 280
BT=G2/G1
G1=G2
DO 660 J=1,N
660 D(J)=-GFI(J)+BT*D(J)
GD=0.0
DO 680 J=1,N
680 GD=GD+GFI(J)*D(J)
IF(GD.LE.0.0) GOTO 380
DO 700 J=1,N
700 D(J)=-GFI(J)
GOTO 380
720 IER=1
740 CALL GRADFI(M,NA,AS,COMAS,QUEMAS,A,COM,QUEM,R,COMR,QUEMR,G,GFI,
& IND, ICNA,N,AUX,V,Y,EPS,AUX1,AUX2,AUX3,AUX4,AUX5,KON,MAX)
RETURN
END

SUBROUTINE NORM(V,VMAX,N)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 V(N)
TOL=1.D-5
VMAX=0.D0
DO 20 I=1,N
T=ABS(V(I))
IF(T.LT.VMAX) GOTO 20
VMAX=T
20 CONTINUE
IF (VMAX.EQ.0) GOTO 80
IF (VMAX.LT.TOL) GOTO 60
DO 40 I=1,N
40 V(I)=V(I)/VMAX
RETURN
60 VMAX=1.D0
RETURN
80 WRITE(ITELA,*) 'VETOR COM NORMA NULA'
STOP
END

C
C ARQUIVO PASSODBL.FOR

```



```

C      SUBROUTINE PASMAX(M,N,A,COM,QUEM,B,X,L,U,DTIL,AMAX,INDT,
1      NA,ISGN,EPSP)
C
C      CALCULA O PASSO MAXIMO PERMITIDO PARA A SUB-ROTINA
C      DE GRADIENTES CONJUGADOS
C
C      AMAX CONTERA O VALOR DO MULTIPLO ESCALAR MAXIMO
C      A SER DADO
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 A(1),B(1),X(N),L(N),U(N),DTIL(N)
C      INTEGER COM(1),QUEM(1),INDT(N)
C      INTEGER*2 ISGN(1)
C      AMAX=1.D+30
C      ACC=1.D-12
C      EPS=EPSP*1.D-1
C      IF( EPS.LT.ACC ) EPS=ACC
C      IF(M.EQ.0) GOTO 80
C      DO 60 I=1,M
C      IF(NA.EQ.0) GOTO 20
C      FAZ A BUSCA SEQUENCIAL PARA VER SE O INDICE " I "
C      ENCONTRA-SE NO VETOR INDT(.).. CASO AFIRMATIVO
C      PULE PARA O FINAL DO LACO
C
C      CALL ACHA(INDT,N,I,IPOS,1,NA)
C      IF( IPOS.NE.0) GOTO 60
C
C      20  CALL AIX(I,N,A,COM,QUEM,X,AX)
C      CALL AIX(I,N,A,COM,QUEM,DTIL,ADTIL)
C      BAX=B(I)-AX
C      IF( DABS(ADTIL).LT.EPS) GOTO 60
C      IF( DABS(ADTIL).LT.EPSP .AND. ISGN(I).EQ.0) GOTO 60
C
C      40  IF(ADTIL.LT.EPSP.AND.ISGN(I).EQ.-1) GOTO 60
C      IF(ADTIL.GT.-EPSP.AND.ISGN(I).EQ.1) GOTO 60
C      IF(DABS(BAX).LT.EPS) GOTO 1
C
C      T=BAX/ADTIL
C      IF (T.LT.ACC) GOTO 60
C      IF(T.GE.AMAX) GOTO 60
C      AMAX=T
C      60  CONTINUE
C
C      80  DO 160 I=1,N
C      IF( DABS(DTIL(I)).LT.EPS) GOTO 160
C      IF(DTIL(I) 100,160,120
C
C      100  T=(L(I)-X(I))/DTIL(I)
C      GOTO 140
C      120  T=(U(I)-X(I))/DTIL(I)
C      140  IF(T.GE.AMAX) GOTO 160
C      AMAX=T
C      160  CONTINUE
C      END
C
C      ARQUIVO ACHA.FOR

```

```

C      SUBROUTINE ACHIA(A,N,IND,IPOS,IA,IB)
C
C      PROPOSITO: FAZ UMA BUSCA SEQUENCIAL PARA DETERMINAR
C      A POSICAO "IPOS" DE UM INDICE "IND" DE UM VETOR
C      NAO-ORDENADO DE INTEIROS DADO POR A(.)
C      ENTRE OS LIMITES "IA" E "IB"
C      SE A BUSCA NAO ENCONTRAR O INDICE ENTAO
C      A VARIAVEL IPOS SAIRA' COM VALOR ZERO
C
C      INTEGER A(N),IND,IPOS
C      IPOS=0
C      IF( IA.GT.IB) RETURN
C      DO 20 K=IA,IB
C      IF(A(K).EQ.IND) GOTO 40
20    CONTINUE
C      RETURN
40    IPOS=K
C      RETURN
C      END
C
C      ARQUIVO ETRANSDBL.FOR
C
C      SUBROUTINE TRANS(M,N,A,COM,QUEM,INDT,NA,ICNA,NCNA,B,COMB,
1    QUEMB,NRAX,FAZ,DEG,COMR,QUEMR,R,NDIM,V,IQ,IV,NDIMV,EPS,AUX)
C
C      ESTA ROTINA FAZ A SELECAO DA MATRIZ DE TRABALHO B
C      DE ESTRUTURA ESPARSA COMB E QUEMB ATRAVES DA CHAMADA
C      DE SELEC. FAZ TAMBEM A ATUALIZACAO DO CONJUNTO
C      DE TRABALHO INDT, DETERMINA A ESTRUTURA DA MATRIZ
C      DE PRECONDICIONAMENTO PELA CHAMADA DE PRECO1
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 A(1),R(NDIM),B(1),V(NDIMV),AUX(N)
C      INTEGER COM(1),COMB(1),COMR(1)
C      INTEGER QUEM(1),QUEMB(1),QUEMR(NDIM)
C      INTEGER INDT(N),ICNA(N)
C      INTEGER IQ(NDIMV),IV(NDIMV)
C      LOGICAL FAZ,DEG,REPRE,RESE
C      COMMON/NAANT/NAANT
C      COMMON/FAIXA/ NF
C      REPRE=.TRUE.
C      RESE=.TRUE.
C      ITELA= 6
C      IF(NA.EQ.0) RETURN
C      IF( NF.EQ.0) NFA=NA
C      IF(NF.LE.2) NF=2
C      NFA=NF
C
C      FAZ A SELECAO DAS COLUNAS DA MATRIZ A
C
C      ICH=0
C      NX=NRAX
C      CALL SELEC(N,A,COM,QUEM,INDT,NA,ICNA,NCNA,B,COMB,QUEMB,AUX)
C
C      COMPLETA O CONJUNTO INDT
C
C      IF(NRAX.GE.N) GOTO 40
C      DO 20 J=NA+1,NCNA

```

```

NRAX=NRAX+1
INDT(NRAX)=ICNA(J)
IF(NRAX.EQ.N) GOTO 40
20  CONTINUE
40  IF(FAZ) GOTO 100
C   SE NAO FAZ O PRECONDICIONAMENTO ENTAO COLOQUE
C   SE FOR O CASO, UMA SUB-MATRIZ IDENTIDADE ONDE FOR NECESSARIO
C   PARA COMPLETAR A ESTRUTURA DE "R". SE "R" TIVER, POR ACASO
C   A MESMA DIMENSAO DA ITERACAO ANTERIOR, ELA ENTRARA'
C   COMO PRECONDICIONAMENTO
IF(NA.EQ.NAANT) RETURN
IF(NAANT.EQ.0 .OR. NA.LT.NAANT) GOTO 80
IC=COMR(NAANT+1)-1
IC1=IC+1
L=1
DO 60 K=NAANT+1,NA
COMR(K+1)=IC1+K-NAANT
QUEMR(IC+L)=NAANT+L
R(IC+L)=1.
60  L=L+1
RETURN
80  FAZ=.TRUE.
C
C
C   FAZ A ESTRUTURA DE R
C   COM "NFA" FAIXAS
C
100  NFA= MIN0(NA,NFA)
120  TYPE *, 'NUMERO DE FAIXAS = ', NFA
CALL ESTR(NA,COMR,QUEMR,NFA,NDIM)
IF(.NOT. FAZ) GOTO 160
C
C   FAZ O PRECONDICIONAMENTO
C
140  CALL PRECO1(NA,NA,COMB,QUEMB,B,COMR,QUEMR,R,V,IQ,IV,NDIMV,IER)
IF(IER.EQ.0) GOTO 160
WRITE(ITELA,*) 'NDIMV INSUFICIENTE'
STOP
C
160  TOLREL=1.D-9
TOLABS=1.D-9
C
C
CALL CORR2(NA,COMR,QUEMR,R,TOLREL,TOLABS,IERC,ICRASH)
IF(IERC.NE.0) WRITE(ITELA,*) 'HOUE CORRECAO PELA CORR2'
SE NAO E' PARA FAZER O PRECONDICIONAMENTO ENTAO SAI
C
IF(.NOT.FAZ) RETURN
IF(IERC.EQ.0) RETURN
C   O PRECONDICIONAMENTO PRODUZIU UMA MATRIZ "R" SINGULAR
C   DEVE-SE TENTAR ,SE FOR O CASO, FAZER O PRECONDICIONAMENTO
C   COM "NA" FAIXAS. SE ESTA TENTATIVA FALHAR DEVE-SE IR
C   PARA A SUB-ROTINA RESEL
IF(ICH.EQ.1) RETURN
IF(RESE) GOTO 180
IF(NF.LE.NA) RETURN
IF(.NOT. REPRE) RETURN
C   SE A VARIABEL REPRE FOR "TRUE" ENTAO UM NOVO

```

```

C      PRECONDICIONAMENTO E' FEITO COM "NA" FAIXAS
C      CASO CONTRARIO CONTINUA-SE COM A MATRIZ DE PRECONDICIONAMENTO
C      TIPO FAIXA QUE FOI "DESSINGULARIZADA" PELA ROTINA CORR2

C
C      SE RESE =.TRUE. ENTAO OCORRE UMA RESELECAO DAS COLUNAS
C      QUE FORMAM A BASE. CASO CONTRARIO ISTO SO' SERA' FEITO COMPULSO
C      RIAMENTE.
C      NO CASO DE USAR-SE O NUMERO DE FAIXAS SUFICIENTE RESE DEVERA
C      SER COLOCADO EM TRUE PARA GARANTIA
      IF(ICH.EQ.1) GOTO 220
      IF(NFA.EQ.NA) GOTO 180
      NFA=NA
      GOTO 120

C
C      FAZ UMA TENTATIVA DE RESELECAO PARA OBTENIR UMA BASE
C      NAO-SINGULAR
180    CALL RESEL(N,A,COM,QUEM,INDT,NA,ICNA,NCNA,B,COMB,QUEMB,AUX)
C
C      ICH=1
C      ATUALIZA INDT
C
      NRAX=NX
      DO 200 J=NA+1,NCNA
      NRAX=NRAX+1
      INDT(NRAX)=ICNA(J)
      IF(NRAX.EQ.N) GOTO 140
200    CONTINUE
      GOTO 140

C      OCORREU LINHAS L.D. PARA O SISTEMA LINEAR
C      NAS RESTRICOES ATIVAS. E' NECESSARIO PERTURBAR
C      CONVENIENTEMENTE O VETOR TERMO INDEPENDENTE
C      ISTO SERA' FEITO PELA ROTINA "PERTB" FORA DA TRANS
C
220    DEG=.TRUE.
      WRITE(ITELA,*) 'NAO CONSEGUI UMA BASE INVERSIVEL'
      WRITE(ITELA,*) 'TENTA UMA PERTURBACAO NO VETOR B'
      RETURN
      END

      SUBROUTINE SELEC(N,A,COM,QUEM,INDT,NA,ICNA,NCNA,
1          B,COMB,QUEMB,V)

C
C      PROPOSITO : FAZ A SELECAO DAS COLUNAS DA SUB-MATRIZ DE A
C                  DE LINHAS COM INDICES NO CONJUNTO DE RESTRICOES
C                  ATIVAS. SAO SELECIONADAS EXATAMENTE "NA" COLUNAS
C                  E SAO COLOCAS NA MATRIZ B COM ESTRUTURA ESPARSA
C                  POR LINHAS COMB E QUEMB
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),B(1),V(N)
      INTEGER COM(1),QUEM(1),COMB(1),QUEMB(1)
      INTEGER INDT(1),ICNA(1)
      ITEL=6
      IF( NA.EQ.N) GOTO 160
      DO 140 I=1,NA
      IAUX=INDT(I)
      AMAX=0.D0
      DO 20 J=1,N

```

```

20      V(J)=0D0
      DO 40 K=COM(IAUX),COM(IAUX+1)-1
40      V(QUEM(K))=A(K)
      DO 60 J=I,NCNA
      JAUX=ICNA(J)
      IF(DABS(V(JAUX))-LE.AMAX) GOTO 60
      AMAX=DABS(V(JAUX))
      JMAX=J
60      CONTINUE
      IF(AMAX.NE.0.D0) GOTO 120
C
C      TOME A PRIMEIRA COLUNA NAO-NULA DENTRO DOS INDICES ICNA(.)
C
      DO 80 J=I,NCNA
      JAUX=ICNA(J)
      DO 80 K=1,NA
      IAUX=INDT(K)
      DO 80 K1=COM(IAUX),COM(IAUX+1)-1
      IF(QUEM(K1).EQ.JAUX) GOTO 100
80      CONTINUE
      WRITE(ITELA,*) 'POSSIVEL ERRO NO MODELO'
      WRITE(ITELA,*) 'CONFIRA SE NA FORMACAO DAS COLUNAS OU DAS LINHAS'
      WRITE(ITELA,*) 'NAO HA ENGANO.'
      WRITE(ITELA,*) 'CONFIRA OS DADOS DE ENTRADA'
      WRITE(ITELA,*) 'PODE TER HAVIDO FALHA NA ROTINA "RESEL"'
      WRITE(ITELA,*) 'MUDE O PONTO INICIAL, ESCOLHENDO MENOS'
      WRITE(ITELA,*) 'RESTRICOES ATIVAS'
      STOP
C
C      SE PASSOU PARA CA' ENTAO TODAS AS COLUNAS SAO NULAS
C      NOS DEMAIS INDICES. NAO POSSO OBTER, DESTE MODO, UMA
C      BASE INVERSIVEL E POSSIVELMENTE HA' ERRO NO MODELO
C
100     JMAX=J
120     JTEMP=ICNA(I)
      ICNA(I)=ICNA(JMAX)
      ICNA(JMAX)=JTEMP
140     CONTINUE
160     CALL TBLOC(N,A,COM,QUEM,INDT,NA,ICNA,B,COMB,QUEMB,V)
      RETURN
      END
C
C
C      SUBROUTINE TBLOC(N,A,COM,QUEM,INDT,NA,ICNA,B,COMB,QUEMB,V)
C
C      TRANSFERE O BLOCO DE A DE COLUNAS ICNA(1),...,ICNA(NA)
C      PARA A MATRIZ B
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),B(1),V(1)
      INTEGER COM(1),QUEM(1),COMB(1),QUEMB(1)
      INTEGER INDT(1),ICNA(1)
      COMB(1)=1
      NEB=0
      DO 80 I=1,NA
      IAUX=INDT(I)
      DO 20 J=1,N
20      V(J)=0.D0
      DO 40 J=COM(IAUX),COM(IAUX+1)-1

```



```

40      V(QUEM(J))=A(J)
      DO 60 K=1,NA
      KAUX=ICNA(K)
      IF( V(KAUX).EQ.0.D0) GOTO 60
      NEB=NEB+1
      B(NEB)=V(KAUX)
      QUEMB(NEB)=K
60      CONTINUE
      COMB(I+1)=NEB+1
80      CONTINUE
      RETURN
      END

      SUBROUTINE RESEL(N,A,COM,QUEM,INDT,NA,ICNA,NCNA,B,COMB,
1      QUEMB,V)

C
C      FAZ A RESELECAO DAS COLUNAS DE "A" NA PRESENÇA DE UMA
C      BASE SINGULAR. E' FEITA UMA TROCA DOS INDICES DE ICNA
C      QUE SAO RUINS POR OUTROS RESTANTES.
C

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),B(1),V(N)
      INTEGER COM(1),QUEM(1),COMB(1),QUEMB(1)
      INTEGER INDT(1),ICNA(1)
      ITELA= 6
      WRITE(ITELA,*) 'ENTREI NA RESEL'

C
C
      DO 60 I=1,NA
      DO 20 J= NCNA,I,-1
      JAUX=ICNA(J)
      DO 20 K=1,NA
      IAUX= INDT(K)
      DO 20 K1=COM(IAUX),COM(IAUX+1)-1
      IF( QUEM(K1).EQ.JAUX) GOTO 40
20      CONTINUE
40      IF(J.EQ.I) GOTO 60
      JTEMP=ICNA(J)
      ICNA(I)=ICNA(J)
      ICNA(J)=JTEMP
60      CONTINUE

C
C      AGORA TRANSFERE O BLOCO CORRESPONDENTE
C

      CALL TBLOC(N,A,COM,QUEM,INDT,NA,ICNA,B,COMB,QUEMB,V)
      RETURN
      END

C
C      ARQUIVO EGRADBL.FOR
C

      SUBROUTINE GRADFI(M,NA,AS,COMAS,QUEMAS,A,COM,QUEM,R,
1      COMR,QUEMR,G,GFI,INDT,
2      ICNA,N,XA,V,Y,EPS,AUX1,AUX2,AUX3,AUX4,AUX5,KON,MAX)

C
C      CALCULA O GRADIENTE DA FUNCAO
C

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 AS(1),A(1),R(1)

```



```

      REAL*8 G(N),GFI(N)
      REAL*8 AUX1(N),AUX2(N),AUX3(N),AUX4(N),AUX5(N)
      REAL*8 XA(N),Y(N),V(N)
      INTEGER COM(1),COMAS(1),COMR(1)
      INTEGER QUEM(1),QUEMAS(1),QUEMR(1)
      INTEGER INDT(N),ICNA(N)
      S=0.D0
      IF( NA.EQ.0) GOTO 180
      IF(NA.LT.N) GOTO 20

C
C      CHAMA SOLUCAO DE SISTEMA LINEAR TRANSP.
C
      CALL HSATA1(AS,COMAS,QUEMAS,R,COMR,QUEMR,NA,NA,G,Y,GFI,EPS,AUX1,KON,
      & MAX,AUX2,AUX3,AUX4,V,AUX5,IER)
      RETURN

C
C      DO 40 J=1,N
C      GFI(J)=0.D0
C      DO 60 I=1,NA
C      JAUX=ICNA(I)
C      XA(I)=G(JAUX)

      CALL HSATA1(AS,COMAS,QUEMAS,R,COMR,QUEMR,NA,NA,XA,Y,GFI,EPS,AUX1,KON,
      & MAX,AUX2,AUX3,AUX4,V,AUX5,IER)
      DO 140 I=1,NA
      IAUX=INDT(I)
      DO 80 J=1,N
      V(J)=0.D0
      DO 100 K=COM(IAUX),COM(IAUX+1)-1
      V(QUEM(K))=A(K)
      DO 120 J=(NA+1),N
      JAUX=INDT(J)
      GFI(J)=GFI(J)-V(JAUX)*GFI(I)
      CONTINUE
      DO 160 J=(NA+1),N
      GFI(J)=GFI(J)+G(INDT(J))
      RETURN
      DO 200 J=(NA+1),N
      JAUX=INDT(J)
      GFI(J)=G(JAUX)
      RETURN
      END

      SUBROUTINE DIREC(NA,AS,COMAS,QUEMAS,A,COM,QUEM,R,COMR,QUEMR,
      1 D,DTIL,INDT,ICNA,N,EPS,RES,XAUX,V,P,G,AUX,W,KON,MAX)
C
C      CALCULA O VETOR DE DIRECAO
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 AS(1),A(1),R(1),D(N),DTIL(N)
      REAL*8 XAUX(N),V(N),P(N),G(N),AUX(N)
      REAL*8 RES(N),W(N)
      INTEGER COMAS(1),QUEMAS(1),COMR(1),QUEMR(1)
      INTEGER COM(1),QUEM(1)
      INTEGER INDT(N),ICNA(N)

C
      IF(NA.LT.N) GOTO 20
      RESOLVE O SISTEMA LINEAR AS*DTIL=D

```

```

CALL HSATA2(AS, COMAS, QUEMAS, R, COMR, QUEMR, NA, D, V, DTIL, EPS, RES, KON, MAX,
& P, G, AUX, IER)
RETURN
20 DO 40 I=(NA+1), N
IAUX=INDT(I)
40 DTIL(IAUX)=D(I)
IF(NA.EQ.0) RETURN
C
C
DO 60 I=1, N
60 XAUX(I)=0.D0
DO 140 I=1, NA
IAUX=INDT(I)
S=0.D0
DO 80 J=1, N
80 V(J)=0.D0
DO 100 K=COM(IAUX), COM(IAUX+1)-1
100 V(QUEM(K))=A(K)
DO 120 J=(NA+1), N
JAUX=INDT(J)
120 S=S+V(JAUX)*DTIL(JAUX)
XAUX(I)=D(I)-S
140 CONTINUE
DO 160 J=1, N
160 W(J)=0.D0
C RESOLVE O SISTEMA LINEAR AS*W=XAUX
C W SERA' PARTE DO VETOR DTIL COM OS INDICES FORNECIDOS
C PELO VETOR ICNA DOS INDICES DAS CANALIZACOES NAO-ATIVAS
C
CALL HSATA2(AS, COMAS, QUEMAS, R, COMR, QUEMR, NA, XAUX, V, W, EPS, RES, KON, MAX,
& P, G, AUX, IER)
DO 180 J=1, NA
JAUX=ICNA(J)
180 DTIL(JAUX)=W(J)
RETURN
END
C
C
ARQUIVO HSATA1DBL.FOR
SUBROUTINE HSATA1(A, COM, QUEM, R, COMR, QUEMR, M, N, B, Y, X, EPSX,
1 RES, KON, MAX, G, D, AG, AD, AUX, IER)
C OBJETIVO: RESOLVER  $A(T)*A*Y = B$ , ONDE A E' DE  $M \times N$ .
C O VETOR X =  $A*Y$  TAMBEM E' SAIDA. ESTE VETOR E' A SOLUCAO DE
C NORMA MINIMA DE  $A(T)*X=B$ .
C METODO: R E' UMA MATRIZ TRIANGULAR SUPERIOR TAL QUE  $A*(INV)$ 
C E' BEM CONDICIONADA. CONSIDERAMOS ENTAO O SISTEMA EQUIVALENTE:
C  $R(-T)*A(T)*A*R(-1)*Z = R(-T)*B$  (1)
C SEGUIDO DE  $Y = R(INV)*Z$  (2)
C O SISTEMA (1) E' RESOLVIDO PELO METODO DE HESTENES -STIEFFEL
C COM Z(INICIAL) = 0.
C CRITERIO DE PARADA:
C
C IER=0 A NORMA DO RESIDUO RES TORNOU-SE MUITO PEQUENA
C OU O PONTO OBTIDO ESTA' MUITO PROXIMO DO DA ITERACAO
C ANTERIOR
C RESIDUO =  $A(T)*A*R(-1)*Z-B$ 
C IER = 1 QUANDO O NUMERO DE ITERACOES FOI ESGOTADO
C IER = 2 QUANDO O PROCESSO FOI DETIDO POR IMPOSSIBILIDADE DE
C ATINGIR A SOLUCAO COM A PRECISAO REQUERIDA.
C IMPLICIT REAL*8 (A-H,O-Z)

```

```

      INTEGER COM(M+1),QUEM(1),COMR(N),QUEMR(1)
      REAL*8 X(M),Y(N),RES(N)
      REAL*8 G(N),D(N),AG(M),AD(M),AUX(N)
      REAL*8 B(N)
      REAL*8 A(1),R(1)
      MAX=N+2
      EPS=DMIN1(EPSX,1.D-8)
      DO 20 I=1,N
20      Y(I)=0.D0
      KON=0
      KONRES=0
      C
      40      CALCULO DO RESIDUO. PRIMEIRO X=A*R(-1)*Y
      C      CALL AXY(Y,M,N,X,A,COM,QUEM,R,COMR,QUEMR,AUX)
      DEPOIS RES=A(T)*X-B
      CALL ATXBSP(M,N,A,COM,QUEM,X,RES)
      RNE=0.D0
      DO 60 I=1,N
      RES(I)=RES(I)-B(I)
60      RNE=RNE+RES(I)*RES(I)
      IF(RNE.GT.EPS) GOTO 120
      IER=0
      C
      80      RECONSTRUIR Y
      CALL SOLTSE(N,R,COMR,QUEMR,AUX,Y)
      DO 100 I=1,N
100      Y(I)=AUX(I)
      RETURN
120      IF(KON.LT.MAX)GOTO140
      IER=1
      GOTO80
      C
140      CALCULAR O GRADIENTE G=R(-T)*RES
      CALL SOLTIE(N,R,COMR,QUEMR,G,RES)
      KON=KON+1
      KONRES=KONRES+1
      IF(KONRES.EQ.N+2)KONRES=1
      IF(KONRES.NE.1)GOTO180
      DO 160 i=1,n
160      d(i)=-g(i)
      goto240
180      call axy(g,m,n,ag,A,COM,QUEM,R,COMR,QUEMR,AUX)
      agad=0.D0
      DO 200 i=1,m
200      agad=agad+ag(i)*ad(i)
      beta=agad/ad2
      DO 220 i=1,n
220      d(i)=-g(i)+beta*d(i)
240      gd=0.D0
      DO 260 i=1,n
260      gd=gd+g(i)*d(i)
      IF(GD.LE.0.D0)GOTO300
      DO 280 I=1,N
280      D(I)=-G(I)
      KONRES=1
      GOTO240
300      call axy(d,m,n,ad,A,COM,QUEM,R,COMR,QUEMR,AUX)
      ad2=0.D0
      DO 320 i=1,m
320      ad2=ad2+ad(i)*ad(i)
      IF(AD2.GT.0.D0)GOTO340
      IER=2
      GOTO80

```

```

340      alfa=-gd/ad2
      do 360 i=1,n
360      Y(i)=Y(i)+alfa*d(i)
      IF( ALFA*AD2.GT.EPS) GOTO 40
      IER=0
      GOTO 90
      end

C
C      ARQUIVO HSATA2DBL.FOR
C
      SUBROUTINE HSATA2(A,COM,QUEM,R,COMR,QUEMR,N,B,Y,X,EPS,RES,
1      KON,MAX,P,G,AUX,IER)

C
C      OBJETIVO: RESOLVER O SISTEMA LINEAR AX=B OU EQUIVALENTEMENTE
C      O SISTEMA A*R(INV)*Y=B ONDE X=R(INV)*Y. A E' QUADRADA DE ORDEM N
C      METODO: HESTENES STIEFEL COM PRECONDICIONAMENTO DADA PELA
C      MATRIZ R. SENDO QUE A*R(INV) E' BEM CONDICIONADA.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 X(N),Y(N),RES(N),P(N),G(N),AUX(N)
      REAL*8 A(1),R(1),B(N)
      INTEGER COM(1),QUEM(1),COMR(1),QUEMR(1)

C
      IER=0
      EPSH= DMIN1(EPS,1.D-8)
      MAX=N+2
      DO 20 I=1,N
20      Y(I)=0.D0
      KON=0
      KONRES=0

C
C      RESIDUO= A*R(INV)*Y
C
      CALL AXY(Y,N,N,RES,A,COM,QUEM,R,COMR,QUEMR,AUX)
      DO 40 I=1,N
40      RES(I)=B(I)-RES(I)

C
C      P=R(-T)*A(T)*RES
C
      CALL ATXY(RES,N,N,P,A,COM,QUEM,R,COMR,QUEMR,AUX)
      RNE=0.D0
      PNE=0.D0
      DO 80 I=1,N
80      RI=RES(I)
      RNE=RNE+RI*RI
      PNE=PNE+P(I)*P(I)
      CONTINUE
      IF(RNE.LT.EPSH) GOTO 260

C
C
C
100      ALFA=RNE/PNE
      ALM=ALFA*PNE
      DO 120 I=1,N
120      ALFAP=ALFA*P(I)
      Y(I)=Y(I)+ALFAP

      KON=KON+1

```

```

C
C
C
C
C      TESTE DE CONVERGENCIA
C
C      CALCULA RESIDUO NO NOVO Y
C      CALL ATXY(Y,N,N,RES,A,COM,QUEM,R,COMR,QUEMR,AUX)
C      RNE=0.D0
C      DO 140 I=1,N
C      RES(I)=B(I)-RES(I)
140    RNE=RNE+RES(I)*RES(I)
C
C
C      IF(RNE.LT.EPSH) GOTO 260
C
C
160    IF(KON.GT.MAX) GOTO 240
C      KONRES=KONRES+1
C      IF(KONRES.LT.N+1) GOTO 180
C      KONRES=0
C      GOTO 60
C
C
C      G=R(-T)*A(T)*RES
C
C
180    CALL ATXY(RES,N,N,G,A,COM,QUEM,R,COMR,QUEMR,AUX)
C      GP=0.D0
C      DO 200 I=1,N
C      GP=GP+G(I)*P(I)
200    BETA=GP/PNE
C      PNE=0.D0
C      DO 220 I=1,N
C      P(I)=G(I)-BETA*P(I)
220    PNE=PNE+P(I)*P(I)
C      IF(PNE.GT.0.D0) GOTO 100
C      IER=2
C      GOTO 260
C
C
240    IER=1
C
C
C      RECONSTROI O VETOR X
C
C
260    CALL SOLTSE(N,R,COMR,QUEMR,X,Y)
C      RETURN
C      END
C
C
C      ARQUIVO      OPERDBL.FOR
C
C
C      ARQUIVO ATXBSP.FOR
C      SUBROUTINE ATXBSP(M,N,A,COM,QUEM,X,B)
C      DADA A MATRIZ DE M X N , CUJA  ESTRUTURA ESPARSA POR LINHAS
C      E DADA POR COM, QUEM E A,ESTA ROTINA CALCULA O PRODUTO
C      B = A(TRASPOSTA)*X.
C      IMPLICIT REAL*8 (A-H,O-Z)
C      REAL*8 A(1),X(M),B(N)
C      INTEGER COM(M+1),QUEM(1)
C      DO 20 I=1,N
C      B(I)=0.D0
20    DO 60 I=1,M
C      IF(COM(I).EQ.COM(I+1))GOTO60
C      DO 40 K=COM(I),COM(I+1)-1
C      K1=QUEM(K)
40    B(K1)=B(K1)+A(K)*X(I)

```



```

60      CONTINUE
      RETURN
      END

C
C      ARQUIVO AXBESP.FOR
C
      SUBROUTINE AXBESP(M,N,A,COM,QUEM,X,B)
C      DADA A MATRIZ DE M X N , CUJA ESTRUTURA ESPARSA POR LINHAS
C      E DADA POR COM, QUEM E A, ESTA ROTINA CALCULA O PRODUTO
C      B = A*X.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),X(N),B(M)
      INTEGER COM(M+1),QUEM(1)
      DO 40 I=1,M
      S =0.D0
      DO 20 J=COM(I),COM(I+1)-1
20      S = S +A(J)*X(QUEM(J))
      B(I)=S
40      CONTINUE
      RETURN
      END

C
C      arquivo soltie.for
C
      subroutine soltie(n,r,comr,quemr,x,b)
C      seja r triangular superior nao singular de n x n.
C      onde sua estrutura esparsa por linhas triangular
C      superior esta dada por comr e quemr.
C      esta rotina resolve r(trasposta)*x = b
      IMPLICIT REAL*8 (A-H,O-Z)
      real*8 x(n),b(n),r(1)
      integer comr(n),quemr(1)
      DO 20 I=1,N
20      x(i)=b(i)
      do 60 i=1,n
      x(i)=x(i)/r(comr(i))
      if(i.eq.n)goto60
      do 40 k=comr(i)+1,comr(i+1)-1
40      x(quemr(k))=x(quemr(k))-r(k)*x(i)
60      continue
      return
      end

C
C      arquivo soltse.for
C
      subroutine soltse(n,r,comr,quemr,x,b)
C      resolve o sistema triangular superior esparso de n x n
C      r x = b
C      onde a eepl triangular superior de r esta dada por comr e quemr
C      e r e nao singular.
      IMPLICIT REAL*8 (A-H,O-Z)
      real*8 r(1),x(n),b(n)
      integer comr(n),quemr(1)
      x(n)=b(n)/r(comr(n))
      if(n.eq.1)return
      do 40 i=n-1,1,-1
      ac=0.D0
      do 20 k=comr(i)+1,comr(i+1)-1
20      ac=ac+r(k)*x(quemr(k))
      x(i)=(b(i)-ac)/r(comr(i))

```


40 continue
return
end

SUBROUTINE AXY(X,M,N,Y,A,COM,QUEM,R,COMR,QUEMR,AUX)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(N),Y(M),A(1),R(1)
REAL*8 AUX(N)
INTEGER COMR(N),COM(M+1),QUEMR(1),QUEM(1)
CALL SOLTSE(N,R,COMR,QUEMR,AUX,X)
CALL AXBESP(M,N,A,COM,QUEM,AUX,Y)
RETURN
END

SUBROUTINE ATXY(X,M,N,Y,A,COM,QUEM,R,COMR,QUEMR,AUX)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(M),Y(N),A(1),R(1)
REAL*8 AUX(N)
INTEGER COMR(N),COM(M+1),QUEMR(1),QUEM(1)
CALL ATXBSP(M,N,A,COM,QUEM,X,AUX)
CALL SOLTIE(N,R,COMR,QUEMR,Y,AUX)
RETURN
END

C ARQUIVO G1G2.FOR
SUBROUTINE G1(A,B,CO,SI,SIG)
IMPLICIT REAL*8 (A-H,O-Z)
IF(DABS(A).LE.DABS(B))GOTO20
XR=B/A
YR=DSQRT(1.+XR*XR)
CO=DSIGN(1./YR,A)
SI=CO*XR
SIG=DABS(A)*YR
GOTO80
20 IF(B)40,60,40
40 XR=A/B
YR=DSQRT(1.+XR*XR)
SI=DSIGN(1./YR,B)
CO=SI*XR
SIG=DABS(B)*YR
GOTO80
60 SIG=0.D0
CO=0.D0
SI=1.0D0
80 RETURN
END

SUBROUTINE G2(CO,SI,X,Y)
IMPLICIT REAL*8 (A-H,O-Z)
XR=CO*X+SI*Y
Y=-SI*X+CO*Y
X=XR
RETURN
END

ARQUIVO ESTRDBL.FOR

SUBROUTINE ESTR(N,COMR,QUEMR,NFA,NDIM)

PROPOSITO: OBTEM A ESTRUTURA COMR E QUEMR DA MATRIZ

```

C      DE PRECONDICIONAMENTO "R" DO TIPO FAIXA.
C      "R" E' TRIANGULAR SUPERIOR COM NFA FAIXAS
C      NDIM E' A DIMENSAO DADA A QUEMR NO PROGRAMA PRINCIPAL
C
      INTEGER COMR(1),QUEMR(NDIM)
C
      ITELA= 6
      IF(N.EQ.1) GOTO 100
      IF(N.EQ.2) GOTO 120
      IF(NFA.LE.0) NFA=2
      IF(NFA.GT.N) NFA=N
20     NMEM=NFA*(2*N-NFA+1)/2
      IF(NMEM.GT.NDIM) GOTO 140
40     NFA1=NFA-1
      L=1
      COMR(1)=1
      DO 80 K=1,N
      K1=K+NFA1
      KA=K1
      IF(KA.GT.N) KA=N
      DO 60 J=K,KA
      QUEMR(L)=J
60     L=L+1
      COMR(K+1)=L
80     CONTINUE
      RETURN
100    COMR(1)=1
      COMR(2)=2
      QUEMR(1)=1
      RETURN
120    COMR(1)=1
      COMR(2)=3
      COMR(3)=4
      QUEMR(1)=1
      QUEMR(2)=2
      QUEMR(3)=2
      RETURN
C
140    WRITE(ITELA,9000)
9000   FORMAT(//,2X,'DIMENSIONAMENTO INSUFICIENTE PARA QUEMR',//)
      STOP
      END
C
C      ARQUIVO ECONSDBL.FOR
C
      SUBROUTINE CONSIS(M,N,A,COM,QUEM,L,U,G,Y)
C
C      FAZ A CONSISTENCIA DE ALGUNS DADOS DE ENTRADA
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),L(1),U(1)
      REAL*8 G(N),Y(N)
      INTEGER COM(M+1),QUEM(1)
      COMMON/OPCAO/IOP,IPARE,IPRI
      ITELA= 6
      IF(N.LT.1) GOTO 260
      IF(M.LT.1) WRITE(ITELA,9100)
      IF(IOP.EQ.0) GOTO 160
      IF(IOP.EQ.1) WRITE(ITELA,*) 'PROBLEMA LINEAR'

```

```

IF(IOP.EQ.2) WRITE(ITELA,*) 'PROBLEMA NAO-LINEAR'

DO 20 J=1,N
UL=U(J)-L(J)
IF(UL.LT.0.D0) GOTO 240
20 CONTINUE
IF(M.EQ.0) GOTO 80
IF(COM(M+1).LT.1) GOTO 220
DO 40 J=1,M
IF(COM(J).EQ.COM(J+1)) GOTO 280
40 CONTINUE
DO 60 J=1,COM(M+1)-1
IF(A(J).EQ.0.D0) GOTO 200
60 CONTINUE
80 HM=1.D0
HT=1.D-1
H1=1.D-5
DO 100 J=1,N
XU=U(J)
XL=L(J)
IF(L(J).LT.-HM) XL=-HM
IF(U(J).GT.HM) XU=HM
Y(J)=(XL+XU)/2.D0
IF(Y(J).EQ.0.D0) Y(J)=5.D-1
100 CONTINUE
MODO=1
KVO=1
120 CALL FUN(N,Y,F,G,MODO)
MODO=0
GK=G(KVO)
F1=F
Y(KVO)=Y(KVO)+H1
CALL FUN(N,Y,F,G,0)
IF (F.NE.F1) GOTO 140
Y(KVO)=Y(KVO)-H1
KVO=KVO+1
IF(KVO.GT.N) RETURN
GOTO 120
140 DF=(F-F1)/H1
IF(ABS(DF-GK).GT.HT) GOTO 180
C
RETURN
160 WRITE(ITELA,*) 'ERRO NA ESPECIFICACAO DA VARIAVEL "IOP" '
STOP
180 WRITE(ITELA,9000)
9000 FORMAT(2X,'POSSIVEL ERRO NA CODIFICACAO DO GRADIENTE',///)
TYPE*, 'DF= ',DF
TYPE *, 'GK= ',GK
TYPE *, 'KVO= ',KVO
PAUSE 'DE CONT SE DESEJA CONTINUAR'
WRITE(ITELA,*) ' RETOMOU O PROCESSO '
RETURN
200 WRITE(ITELA,9020)
9020 FORMAT(2X,'MATRIZ A FORNECIDA COM ELEMENTO(S) NULO(S)',/
1 ,2X,' NA SUA ESTRUTURA ESPARSA',//)
STOP
220 WRITE(ITELA,9040)
9040 FORMAT(2X,'VETOR "COM" APRESENTA IRREGULARIDADE',//)
STOP
240 WRITE(ITELA,9060)

```

```

9060  FORMAT(2X,'COMPONENTE DE VETOR U  MENOR ',/,
      1  2X,'QUE  COMPONENTE DE VETOR L',//)
      STOP
260   WRITE(ITELA,9080)
9080  FORMAT(2X,'VALOR DE N MENOR  QUE 1 ',//)
      STOP
9100  FORMAT(2X,'PROBLEMA IRRESTRITO',//)
280   WRITE(ITELA,9120)
9120  FORMAT(2X,'MATRIZ COM LINHA NULA',//)
      STOP
      END

```

הנהגות

FUNCAO SINAL DE ARGUMENTO DUPLA PRECISAO

```

      FUNCTION SGN(X)
      REAL*8 SGN,X
      IF(X) 20,40,60
20    SGN=-1.D0
      RETURN
40    SGN=0.D0
      RETURN
60    SGN=1.D0
      RETURN
      END

```

C
C
C
C

ARQUIVO BUSCA2DBL.FOR

SUBROUTINE BUSCA(X0,X,D,G,F,ALFA,DALFA,AMAX,N,EPS,NCFG,IER)

C
C
C
C

PROPOSITO: FAZ A BUSCA UNIDIMENSIONAL NA
DIRECAO "D" A PARTIR DE "X0".
"AMAX" E' O PASSO MAXIMO PERMITIDO
"X" CONTRA O PONTO OTIMO DA BUSCA, ALFA E'
O ESCALAR MULTIPLICATIVO PARA "D" NORMALIZADA
E "DALFA" E' O VALOR DA DERIVADA DIRECIONAL
"NCFG" CONTRA O NUMERO DE CHAMADAS A
SUB-ROTINA FUN.

CCCC

IMPORTANTE : NA ENTRADA DEVE-SE TER DISPONIVEL
O VALOR DA FUNCAO "F" NO PONTO INICIAL
X0

הנהגה

```
IER = 0 - SAIDA NORMAL
      1 - A DIRECAO INICIAL E' DE SUBIDA
          OU A FUNCAO JA NAO PODE SER MELHORADA PARA
          A TOLERANCIA ESPECIFICADA
      2 - ATINGIU "MAXC" CHAMADAS A SUB-ROTINA FUN
      3 - ATINGIU UM INTERVALO MUITO PEQUENO SEM
          ATINGIR A TOLERANCIA PEDIDA
```

CCCE

CCC

```

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X0(N),X(N),D(N),G(N)
      COMMON/OPCAD/IOP,IPARE,IPRI

```

```

INTEGER MODE
ITELA= 6
IER=0
GRANDE=1.D+6
TOL=DMAX1(EPS*1.D-3,1.D-12)
EPS1=TOL
EPSF=DMAX1(DABS(F),1.D0)*TOL
NCFG=0
AC=1.D-2
MAXC=100

C
C   GUARDA A FUNCAO DO PONTO DE ENTRADA
C
C   FINIC = F
C
C
C   SE IOP=1 ==> FUNCAO E' LINEAR
C   SE IOP=2 ---> FUNCAO E' NAO-LINEAR GERAL
C
      ALFA=AMAX
      GOTO (20,100) IOP
20    IF(ALFA.GE.GRANDE) GOTO 80
      DO 40 J=1,N
40    X(J)=X0(J)+ALFA*D(J)
      CALL FUN(N,X,F,G,MODE)
      NCFG=1
      DALFA=0.D0
      DO 60 J=1,N
60    DALFA=DALFA+D(J)*G(J)
      RETURN
80    WRITE(ITELA,*)
9000  FORMAT(2X,'SOLUCAO ILIMITADA PARA PROGRAMA LINEAR')
      STOP

C
C
C   FAZ A BUSCA PARA O CASO NAO LINEAR
C
C
100   A=0.D0
      DALFA=0.D0
      DO 120 J=1,N
120  DALFA=DALFA+G(J)*D(J)
C
      IF(AMAX .LE. TOL) GOTO 540
C
C   RETORNE SE A DIRECAO NAO E' DE DESCIDA
C
      IF( DALFA.LE.-TOL) GOTO 160
      ALFA=0.D0
      DO 140 J=1,N
140  X(J)=X0(J)
      GOTO 520
C
C   INICIALIZAR VARIAVEIS DE CONTROLE DA BUSCA
C
160  A=0.D0
      EPS1=EPS
C
C   MAXC= MAXIMO DE CHAMADAS A SUB-ROTINA FUN
C

```



```

C      AC - FATOR PARA REGRA DE ARMIJO-GOLDSTEIN
C
C      CALCULO DE UMA ESTIMATIVA DO MINIMO
C
C      FA=F
C      FANT=FA
C      FLA=DALFA
C      GANT=DALFA
C      FEST =FA
C      IF(FEST.LT.0.D0) FEST=-2.D0*FA
C      IF(FEST.EQ.0.D0) FEST=1.0D0
C
C      COLOQUE O VALOR DA ESTIMATIVA INICIAL EM B
C
C      B=-FEST/DALFA
C      IF( DALFA.LT.2.D-1 .AND. B.GT.1.D0) B=1.D0
C      IF( B.LT. 1.D0 .AND. AMAX.GT. 1.D0) B=1.D0
C
C      SE A ESTIMATIVA VAI ALEM DO PASSO MAXIMO PERMITIDO
C      FAÇA-A IGUAL AO PASSO MAXIMO PERMITIDO
C
180    IF(B.GE.AMAX) B=AMAX
C
C      CALCULE FUNCAO E DERIVADA EM B
C
C      DO 200 J=1,N
200    X(J)=X0(J)+B*D(J)
C      CALL FUN(N,X,F,G,MOD0)
C      FB=F
C      FLB=0.D0
C      DO 220 J=1,N
220    FLB=FLB+G(J)*D(J)
C
C      DALFA=FLB
C      ALFA=B
C      NCFG=NCFG+1
C      IF (NCFG.GT.MAXC) GOTO 420
C
C      SE B E' IGUAL AO PASSO MAXIMO E A DIRECAO E'
C      DE DESCIDA , ENTAO PODE RETORNAR
C
C      IF( B.EQ.AMAX .AND. FLB.LT.EPS) GOTO 400
C
C      SE FB E' MAIOR QUE FA OU FLB E' MAIOR QUE ZERO
C      JA' TEMOS UM INTERVALO PARA A BUSCA
C
C      IF( FB.GE.FA .OR. FLB.GE.EPS) GOTO 280
C
C      A PASSAGEM PARA ESTE BLOCO SIGNIFICA QUE B
C      E' UMA ESTIMATIVA AQUEM DO MINIMO E UMA
C      NOVA ESTIMATIVA DEVERA' SER FEITA
C
C      DELTA=(FB-FA)/(B-A)
C      IF(FLB.GT.DELTA) GOTO 240
C
C      DELTA=FLB-1.D0
C
240    XM=(B+A)/2.

```



```

      BNOVO= (B*DELTA - FLB*XM)/(DELTA-FLB)
C
C      RESETE A EXTREMIDADE A
C      SE FOR O CASO
C
      IF( FA.LT.FB .AND. FLB.LT.EPS) GOTO 260
      A=B
      FLA=FLB
      FA=FB
C
C      RESETE B
C
260    B = BNOVO
C
C      VOLTE COM A NOVA ESTIMATIVA
C
      GOTO 180
C
C      AQUI JA' PODEMOS FAZER A INTERPOLACAO CUBICA
C
280    FANT=F
300    IF( DABS(B-A) .LT.TOL) GOTO 460
      U1=FLA+FLB-3.D0*(FA-FB)/(A-B)
      U2= SQRT(U1*U1-FLA*FLB)
      ALFA=B -(B-A)*(FLB+U2-U1)/(FLB-FLA+2.*U2)
      DO 320 J=1,N
320    X(J)=X0(J) + ALFA*D(J)
      CALL FUN(N,X,F,G,MOD0)
      DALFA=0.D0
      DO 340 J=1,N
340    DALFA=DALFA+G(J)*D(J)
      FALFA=F
      NCFG=NCFG+1
      IF(NCFG.GT.MAXC) GOTO 420
C
C      EPS2 = MEMBRO DA REGRA DE ARMIJO
C
      EPS2=DABS(GANT)*AC
C
C      RESETE O FATOR SE FOR NECESSARIO
C
      IF( EPS2.LT.EPS1) EPS2=EPS1
C
C      VERIFICA SE SATISFAZ A REGRA DE ARMIJO
C
      IF(FALFA.LE. FANT .AND. DABS(DALFA).LE. EPS2) GOTO 500
C
C      SE ALFA FOR PEQUENO, ENTAO TEMOS QUE ESGOTAR A BUSCA
C      EM TODA A PRECISAO. FAZENDO AC=EPS1
C
      IF(ALFA.GT.1.D-5) GOTO 360
      AC=EPS1
360    IF( DALFA.GE. 0.D0) GOTO 380
C
C      RESETA A EXTREMIDADE A
C
      A=ALFA
      FA=FALFA
      FLA=DALFA
      GANT=FLA

```

```

FANT=FALFA
GOTO 300

C
C
C
380  B=ALFA
      FB=FALFA
      FLB=DALFA
      GANT=FLB
      FANT=FALFA
      GOTO 300

C
C
C
      SAIDA NA EXTREMIDADE DO INTERVALO

400  ALFA=B
      DALFA=FLB
      RETURN

420  WRITE(ITELA,*) 'ATINGIU O MAXIMO PERMITIDO DE CHAMADAS'
      WRITE(ITELA,*) ' A SUB-ROTINA FUN'
      DO 440 J=1,N
440  X(J)=X0(J)+ALFA*D(J)
      IER=2
      RETURN

460  IER=3
      ALFA=B
      DO 480 J=1,N
480  X(J)=X0(J)+ALFA*D(J)
      RETURN

500  IF((FINIC-FALFA) .GT.EPSF ) RETURN
      IF(ALFA.GT.EPS) RETURN
520  IER=1
      RETURN
540  DO 560 J=1,N
560  X(J)=X0(J)
      IER=3
      RETURN
      END

C
C
C
      ARQUIVO CORR2DBL.FOR

SUBROUTINE CORR2(N,COMR,QUEMR,R,TOLREL,TOLABS,IER,ICRASH)
C
C  SEJA R MATRIZ DE N X N, TRIANGULAR SUPERIOR, CUJA ESTRUTURA
C  ESPARSA POR LINHAS TRIANGULAR SUPERIOR COM DIAGONAL INCLUIDA
C  ESTA DADA POR COMR, QUEMR E R.
C  ESTA ROTINA TESTA SE HA ALGUM ELEMENTO DA DIAGONAL NULO, COM
C  TOLERANCIAS TOLABS E TOLREL. CASO ENCONTRE UM, A SUBMATRIZ
C  DE R CORRESPONDENTE A ESSE ELEMENTO EM DIANTE, E SUBSTITUIDA
C  POR UMA SUBMATRIZ IDENTIDADE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 R(1)
      INTEGER COMR(N),QUEMR(1)
      IER=0
      DO 40 I=1,N
      I1=COMR(I)
      Z=DABS(R(I1))
      IF(Z.LT.TOLABS)GOTO60
      IF(I.EQ.N)GOTO40
      I2=COMR(I+1)
      IF(I1+1.EQ.I2)GOTO40
      Z1=0.

```

```

DO 20 J=I1+1,I2-1
20  Z1=DMAX1(Z1,DABS(R(J)))
    IF(Z.LT.Z1*TOLREL)GOTO60
40  CONTINUE
    RETURN
50  IER=1
    ICRASH=I
    DO 80 I=ICRASH,N
80  R(COMR(I))=1.
    DO 120 I=1,N
    IF(I.EQ.N)GOTO120
    I1=COMR(I)
    I2=COMR(I+1)
    IF(I1.EQ.I2-1)GOTO120
    DO 100 J=I1+1,I2-1
100  R(J)=0.
120  CONTINUE
    RETURN
    END

C
C  ARQUIVO PRECO1.FOR
C
    subroutine PRECO1 (m,n,com,quem,a,COMR,QUEMR,R,V,IQ,IV,NDIMV,IER)
C
C  SEJA A UMA MATRIZ DE MXN, M.GE.N, CUJA EEPL E DADA POR COM E
C  QUEM. ESTA ROTINA CALCULA UMA MATRIZ TRIANGULAR SUPERIOR NAO
C  SINGULAR DE NXN R CUJA EEPL TRIANGULAR SUPERIOR E DADA POR
C  COMR E QUEMR, TAL QUE A*R*(-1) E MELHOR CONDICIONADA.
C  NDIMV DEVE SER A DIMENSAO DADA A IV,IQ E V NO PROGRAMA DE
C  CHAMADA. N SEMPRE SERA SUFICIENTE.
C  SE IER=1, NDIMV FOI INSUFICIENTE.
    IMPLICIT REAL*8 (A-H,O-Z)
    INTEGER COM(M+1),QUEM(1),COMR(N),QUEMR(1),IV(NDIMV),IQ(NDIMV)
    REAL*8 A(1),R(1),V(NDIMV)
    COMMON/NFAT/NFAT
C
C  COPIAR A PRIMEIRA LINHA DE A NA ESTRUTURA DA PRIMEIRA LINHA DE R.
    IER=0
    NFAT=NFAT+1
    TYPE *, '          PRECONDICIONOU'
    KR=1
    KA=1
20  IF(QUEM(KA).LT.QUEMR(KR))GOTO60
    IF(QUEM(KA).GT.QUEMR(KR).OR.KA.GE.COM(2))GOTO40
    R(KR)=A(KA)
    IF(N.EQ.1)GOTO80
    IF(KR.EQ.COMR(2)-1)GOTO80
    KR=KR+1
    KA=KA+1
    GOTO20
40  R(KR)=0.
    IF(KR.EQ.COMR(2)-1)GOTO80
    KR=KR+1
    GOTO20
60  KA=KA+1
    GOTO20
C
C  CICLO PRINCIPAL DA SUBROTINA
80  IF(M.EQ.1) GOTO 440
    DO 400 LIN=2,M
C
C  COPIAR A LINHA LIN DE A NO VETOR AUXILIAR V
    IF(COM(LIN).EQ.COM(LIN+1))GOTO400
    I=1

```

```

DO 100 K=COM(LIN),COM(LIN+1)-1
  IQ(I)=QUEM(K)
  V(I)=A(K)
100  I=I+1
      NLIN=I-1
      IF(NLIN.GT.NDIMV)GOTO420
      IV(NLIN)=M+1
      IF(NLIN.EQ.1)GOTO140
      DO 120 I=1,NLIN-1
120  IV(I)=I+1
140  J=1
160  IF(J.EQ.M+1.OR.IQ(J).GE.LIN)GOTO300
      C  ELIMINAR V(J)
          LPIV=IQ(J)
          I=COMR(LPIV)
          IF(V(J).EQ.0.D0)GOTO280
          CALL G1(R(I),V(J),CO,SI,SIG)
          R(I)=SIG
      C  APLICAR A ROTACAO AO RESTO
          IF(LPIV.EQ.N)GOTO280
          K1=COMR(LPIV)+1
          K2=IV(J)
          K2ANT=J
180  IF(K1.EQ.COMR(LPIV+1).AND.K2.EQ.M+1)GOTO280
          IF(K1.EQ.COMR(LPIV+1))GOTO260
          IF(QUEMR(K1).LT.IQ(K2).OR.K2.EQ.M+1)GOTO220
          IF(QUEMR(K1).GT.IQ(K2).OR.K1.EQ.COMR(LPIV+1))GOTO260
          CALL G2(CO,SI,R(K1),V(K2))
          K1=K1+1
200  K2ANT=K2
          K2=IV(K2)
          GOTO180
220  IF(NLIN.GE.NDIMV)GOTO420
          IV(K2ANT)=NLIN+1
          IV(NLIN+1)=K2
          K2=NLIN+1
          IQ(K2)=QUEMR(K1)
          V(K2)=0.D0
          CALL G2(CO,SI,R(K1),V(K2))
          NLIN=NLIN+1
          K2ANT=K2
          K2=IV(K2)
240  K1=K1+1
          GOTO180
260  Z=0.D0
          CALL G2(CO,SI,Z,V(K2))
          GOTO200
280  J=IV(J)
          GOTO160
      C  COPIAR V NA LINHA LIN DE R, SE FOR O CASO.
300  IF(LIN.GT.N)GOTO400
          K2=1
          I1=COMR(LIN)
          I2=COMR(LIN)
          IF(LIN.LT.N)I2=COMR(LIN+1)-1
          DO 320 K=I1,I2
320  IF(QUEMR(K).LT.IQ(K2).OR.K2.EQ.M+1)GOTO340
          IF(QUEMR(K).GT.IQ(K2))GOTO360
          R(K)=V(K2)
          K2=IV(K2)

```

```

340      GOTO380
      R(K)=0.D0
      GOTO330
360      K2=IV(K2)
      GOTO320
380      CONTINUE
400      CONTINUE
      RETURN
420      IER=1
      RETURN
440      R(1)=1./A(1)
      RETURN
      END

C
C      ARQUIVO SELINDBL.FOR
C
C      SUBROUTINE SELIND(M,N,A,COM,QUEM,B,X,EPSR,IRATIV,NRAX,I,ISGN)
C
C      SELECIONA OS INDICES DAS RESTRICOES ATIVAS EM X,
C      ATRAVES DE SUCESSIVAS CHAMADAS. OS INDICES SAO
C      COLOCADOS EM IRATIV.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),X(N),B(M)
      INTEGER COM(M+1),QUEM(1),IRATIV(M)
      INTEGER*2 ISGN(M)
      ITEL= 6
      CALL AIX(I,N,A,COM,QUEM,X,AX)
      SAUX=B(I)-AX
      EPS=EPSR*DMAX1(1.0D0,DABS(B(I)))
      IF(DABS(SAUX).LT.EPS) GOTO 20
      IF(ISGN(I).EQ.0) GOTO 40
      IF(SAUX.GT.EPS.AND.ISGN(I).EQ.1) GOTO 40
      IF(SAUX.LT.-EPS.AND.ISGN(I).EQ.-1) GOTO 40
      RETURN
20      NRAX=NRAX+1
      IRATIV(NRAX)=I
      RETURN
40      WRITE(ITEL,9000) SAUX,I,ISGN(I)
9000      FORMAT(//,2X,'PONTO NAO FACTIVEL',//,
1      2X,'RESIDUO = ',G,2X,'RESTRICAO = ',I5,
2      //,2X,'SINAL DA RESTRICAO = ',I2,//)
      PAUSE 'DE CONT SE QUISE CONTINUAR'
      RETURN
      END

      SUBROUTINE AIX (I,N,A,COM,QUEM,X,AX)
C
C      FAZ O PRODUTO DA LINHA I DE A PELO VETOR X
C      COLOCANDO O RESULTADO EM AX
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),X(N)
      INTEGER COM(1),QUEM(1)
      AX=0.D0
      DO 20 K=COM(I),COM(I+1)-1
      AX=AX+A(K)*X(QUEM(K))
20      RETURN
      END

```



```

C
C      ARQUIVO REFORDBL.FOR
C
      SUBROUTINE REFOR(IRATIV,INDT,ISGN,A,COM,QUEM,
&      B,L,U,X,N,NRAX,NRGA,NA,EPS,NAANT)
C      FAZ UMA PERTURBACAO ADEQUADA NO VETOR "B" DE MODO
C      A CONSEGUIR SAIR DE UMA DEGENERACAO. FAZ TAMBEM A CORRECAO
C      NO CONJUNTO IRATIV
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 B(1),L(1),U(1),X(1),A(1)
      INTEGER IRATIV(1),INDT(1),COM(1),QUEM(1)
      INTEGER*2 ISGN(1)
      ITELA=6
      WRITE(ITELA,*) 'ENTROU NA REFOR'
      EPS1=EPS*1.E+2
      IF(NAANT.EQ.0) GOTO 180
      NAUX=NRAX-N
      DO 80 K=1,NRGA
        IND=IRATIV(K)
        CALL ACHA(INDT,N,IND,IPOS,1,NAANT)
        IF (IPOS.NE.0) GOTO 80
        IRATIV(K)=-IND
        NA=NA-1
        EPSB=EPS1*DMAX1(1.D0,DABS(B(IND)))
        IF(ISGN(IND)) 20,20,40
20      B(IND)=B(IND)+EPSB
        GOTO 60
40      B(IND)=B(IND)-EPSB
60      NAUX=NAUX-1
        IF(NAUX.EQ.0) GOTO 120
        CONTINUE

      IF(NRAX.EQ.NRGA) GOTO 120
      DO 100 K=NRGA+1,NRAX
        IND=IRATIV(K)
        CALL ACHA(INDT,N,IND,IPOS,NAANT,N)
        IF(IPOS.NE.0) GOTO 100
        IRATIV(K)=-IND
        NAUX=NAUX-1
        IF(NAUX.EQ.0) GOTO 120
100     CONTINUE
      C
      C
120     J=1
        DO 160 K=1,NRAX
          IND=IRATIV(K)
          IF(IND.LT.0) GOTO 140
          INDT(J)=IND
          J=J+1
          GOTO 160
140     IRATIV(K)=-IND
160     CONTINUE
180     DX=EPS*1.D+1
        DO 200 J=1,N
          IF(U(J)-L(J).LT.EPS) GOTO 200
          IF(DABS(X(J)-L(J)).LT.EPS) X(J)=X(J)+DX*DMAX1(1.D0,DABS(L(J)))
          IF(DABS(X(J)-U(J)).LT.EPS) X(J)=X(J)-DX*DMAX1(1.D0,DABS(U(J)))
200     CONTINUE

```



```

DO 220 I=1,NRGA
J=IRATIV(I)
CALL AIX(J,N,A,COM,QUEM,X,AX)
220 B(J)=AX
RETURN
END

```

```

C
C
C ARQUIVO PERTBDBL.FOR
C

```

```

SUBROUTINE PERTB(M,N,B,INDT,ISGN,NA,EPS)

```

```

C
C PROPOSITO: PERTURBAR O VETOR "B" TERMO INDEPENDENTE
C NA PRESENÇA DE UMA DEGENERACAO TECNICA. A ROTINA
C "SELEC" NAO CONSEGUIU SELECIONAR "NA" COLUNAS L.I. E
C POSSIVELMENTE AS LINHAS DA MATRIZ "A" QUE FORAM SELECIONAS
C TEM POSTO MENOR QUE "NA" .
C NESTE CASO NAO SE PODE ESTABELECEER A PRIORI QUAIS AS
C RESTRICDOES A SEREM ABANDONADAS.
C COLOCA-SE O PONTO PARA DENTRO DO CONVEXO ATRAVES
C DA PERTURBACAO NO VETOR "B"
C

```

```

IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 B(M)
INTEGER INDT(N)
INTEGER*2 ISGN(M)
TYPE *, ' DEGENERACAO TECNICA '

```

```

C
C
C IMAX=0
C EPSB=1.D+2*EPS
C BMAX=0.D0
C DO 20 K=1,NA
C IAUX=INDT(K)
C IF( ISGN(IAUX).EQ.0 ) GOTO 20
C AB=DABS(B(IAUX))
C IF( AB.LT.BMAX ) GOTO 20
C BMAX=AB
C IMAX=IAUX
20 CONTINUE
C IF( IMAX.EQ.0 ) GOTO 80
C TOL=EPSB*DMAX1(1.D0,BMAX)
C IF( ISGN(IMAX) ) 40,40,60
40 B(IMAX)=B(IMAX)+TOL
C RETURN
60 B(IMAX)=B(IMAX)-TOL
C RETURN

```

```

C
C
C 80 TYPE *, ' NAO POSSO SAIR DA PRESENTE DEGENERACAO '
C STOP
C END

```

```

C
C
C ARQUIVO SAIDADBL.FOR
C

```

```

SUBROUTINE SAIDA(NOME,N,M,X,F,G,GFI,NA,STATUS,INDT,ICNA,
1 NCNA,ITMG,ITOT,IER)

```

```

C
C PROPOSITO: FAZ O RELATORIO DE SAIDA DOS PRINCIPAIS
C DADOS DE SAIDA DA SUB-ROTINA BUGRE
C SAIRA' ATRAVES DO ARQUIVO FORNNN.DAT, ONDE

```

C
C
C
C
C

NNN E' DADO PELA VARIÁVEL IOUT

SE A VARIÁVEL LÓGICA "SAI" FOR FALSE SOMENTE
RESULTADOS RESUMIDOS SERÃO APRESENTADOS.

```

IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(N),G(N),GFI(N)
INTEGER ICNA(N),INDT(1)
INTEGER*2 STATUS(1)
REAL*8 NOME
REAL*4 TEMPO
LOGICAL SAI
COMMON/DADOS/ NC,NL,NB,ISH,NRI
COMMON/TEMPO/ TEMPO
COMMON/MEMO/MEM
COMMON/BASE/ NAMED,NAMAX
COMMON/DEV/IN,IOUT
COMMON/NCFUN/NCFG
COMMON/NFAT/NFAT
COMMON/RELAT/ SAI

```

C
C
C

```

IF(.NOT. SAI) GOTO 120
IPAGE=1
WRITE(IOUT,9000) NOME,IPAGE
9000  FORMAT(1H1,40('*'),' RESULTADOS DO PROBLEMA ',A8,2X,40('*'),
2    ' PAGINA ',I3,/,
1    /,T2,'INDICE',I13,'VETOR X',T30,'STATUS',T50,'GRAD(X)',T70,
2    'GRAD(Y)',///)
DO 100 I=1,N
IF( STATUS(I) ) 20,40,60
20    S= 'INFERIOR'
GOTO 80
40    S= 'LIVRE '
GOTO 80
60    S= 'SUPERIOR'
80    WRITE(IOUT,9020) I, X(I),S,G(I),GFI(I)
IF((I/20)*20.NE.I) GOTO 100
IPAGE=IPAGE+1
WRITE(IOUT,9000) NOME,IPAGE
100   CONTINUE
9020  FORMAT(T2,I5,T11,D12.6,T30,A8,T47,D12.6,T67,D12.6)
120   WRITE(IOUT,9040)
9040  FORMAT(1H1)
IF(.NOT. SAI) WRITE(IOUT,9060) NOME
9060  FORMAT(5X,'RESULTADOS RESUMIDOS DO PROBLEMA ',A8,/)
WRITE(IOUT,9080) F
9080  FORMAT(///,10X,'VALOR DA FUNCAO OBJETIVO = ',D12.6,/)
IF(.NOT. SAI) GOTO 140
WRITE(IOUT,9100)
9100  FORMAT(//,10X,'CONJUNTO DE INDICES DAS RESTRICOES DE TRA'
1    'BALHO NA SOLUCAO',/)
WRITE(IOUT,9120) (INDT(J),J=1,NA)
9120  FORMAT(/,5X,1000(10I6,/)
WRITE(IOUT,9140)
9140  FORMAT(//,10X,'CONJUNTO DE INDICES DAS CANALIZACOES '
1    'NAO-ATIVAS NA SOLUCAO',/)
WRITE(IOUT,9120) (ICNA(J),J=1,NCNA)
140   WRITE(IOUT,9160) ITOT

```

```

9160  FORMAT(/,10X,'NUMERO DE ITERACOES = ',I6,/)
      WRITE(IOUT,9180) NFAT
9180  FORMAT(/,10X,'NUMERO DE FATORACOES USADAS = ',I6,/)
      WRITE(IOUT,9200) NCFG
9200  FORMAT(/,10X,'NUMERO DE CHAMADAS A FUNCAO E GRADIENTE = ',I6,/)
      WRITE(IOUT,9220) ITMG
9220  FORMAT(/,10X,'NUMERO DE BASES USADAS = '
      1  ,I6,/)
      WRITE(IOUT,9240) NC,NL,NB,ISH,NRI
9240  FORMAT(/,10X,'NC= ',I5,5X,'NL= ',I5,5X,'NB= ',I5,5X,'ISH = ',
      1  ,I5,5X,'NRI = ',I5,/)
      WRITE(IOUT,9260) N
9260  FORMAT(/,10X,'NUMERO DE VARIAVEIS = ',I7,/)
      WRITE(IOUT,9280) M
9280  FORMAT(/,10X,'NUMERO DE RESTRICOES GERAIS = ',I7,/)
      WRITE(IOUT,9300) IER
9300  FORMAT(/,10X,'CODIGO DE ERRO = ',I6,/)
      WRITE(IOUT,9320) MEM
9320  FORMAT(/,10X,'MEMORIA USADA EM EM PALAVRA DE 8 BYTES (APROX)'
      1  ,I8,/)
      WRITE(IOUT,9340) TEMPO
9340  FORMAT(/,10X,'TEMPO DE CPU (SEGUNDOS) = ',F14.4)
      IF( NAMED.EQ.0) RETURN
      WRITE(IOUT,9360) NAMED
9360  FORMAT(/,10X,'TAMANHO MEDIO DA BASE = ',I5)
      WRITE(IOUT,9380) NAMAX
9380  FORMAT(/,10X,'TAMANHO MAXIMO DA BASE = ',I5)
      RETURN
      END

C
C  ARQUIVO DATAEDBL.FOR
C
      SUBROUTINE DATAE(N,M,COM,QUEM,A,B,ISGN,L,U,X,NOME)

C
C  OBJETIVO: JOGA NO ARQUIVO FORnnn.DAT OS DADOS DE
C            ENTRADA DO PROBLEMA. nnn E' DADO PELA
C            VARIÁVEL IARQ
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),L(N),U(N),X(N),B(1)
      REAL*8 NOME
      INTEGER COM(1),QUEM(1),IARQ
      INTEGER*2 ISGN(1)
      COMMON/OPCAO/IOP,IPARE,IPRI

C
      IARQ= 32
      WRITE(IARQ,9000) NOME
9000  FORMAT(1H1,10X,'DADOS DE ENTRADA DO PROBLEMA ',A8,/)
      WRITE(IARQ,9020) N,M
9020  FORMAT(5X,'NUMERO DE VARIAVEIS = ',I5,/,5X,
      1  'NUMERO DE RESTRICOES = ',I5,/)
      IF( M.EQ.0) GOTO 20
      WRITE(IARQ,9040)
9040  FORMAT(10X,'VETOR COM',/)
      WRITE(IARQ,9060) (COM(J),J=1,M+1)
9060  FORMAT(100(3X,10(I5),/))
      WRITE(IARQ,9080)
9080  FORMAT(/,10X,'VETOR QUEM',/)
      NQ=COM(M+1)-1
      WRITE(IARQ,9060) (QUEM(J),J=1,NQ)

```

```
WRITE(IARQ,9100)
9100  FORMAT(//,10X,'MATRIZ DE TECNOLOGIA',/)
      WRITE(IARQ,9120) (A(J),J=1,NQ)
9120  FORMAT(100(3X,5(D15.5),/))
      WRITE(IARQ,9140)
9140  FORMAT(//,10X,'VETOR TERMO INDEPENDENTE',/)
      WRITE(IARQ,9120) (B(J),J=1,M)
      WRITE(IARQ,9160)
9160  FORMAT(//,10X,'VETOR DE SINAL DAS RESTRICOES',/)
      WRITE(IARQ,9060) (ISGN(J),J=1,M)
20    WRITE(IARQ,9180)
9180  FORMAT(///,10X,'VETORES "L" E "U"',/)
      WRITE(IARQ,9200) (L(J),U(J),J=1,N)
9200  FORMAT(10000(5X,D15.5,4X,D15.5,/))
      WRITE(IARQ,9220)
9220  FORMAT(//,10X,'VETOR PONTO INICIAL',/)
      WRITE(IARQ,9240) (X(J),J=1,N)
9240  FORMAT(10000(3X,5(D15.5,3X),/))
      WRITE(IARQ,9260)
9260  FORMAT(////)
      RETURN
      END
```