EMPRESA BRASILEIRA DE PESQUISA AGROPECUÁRIA

DEPARTAMENTO DE MÉTODOS QUANTITATIVOS

SISTEMA DE PROGRAMAÇÃO NÃO LINEAR

Moacir Pedroso Júnior

Novembro, 1979

1. INTRODUÇÃO

O Sistema de Programação Não Linear (SPNL) é um conjunto integrado de rotinas para resolver problemas de programação não linear restritos. Tais problemas podem formalmente ser assim expressos:

Encontrar um vetor $\hat{x} \epsilon R^n$, tal que

$$f(\hat{x}) = \min f(x),\ x \epsilon S = \{x \mid g_i(x) \geq \emptyset,\ i = 1,2,\ldots,m\},$$

onde f e $g_i : R^n \rightarrow R$, $i = 1,2,\ldots, m$.

Deve ser observado que o sistema minimiza e as restrições devem ser do tipo $\geq \emptyset$. Qualquer problema diferente deve ser alterado para atender essas especificações.

O SPNL foi escrito em FORTRAN IV (G) e apenas uma rotina em ASSEMBLER. Está implantado em um computador IBM 37Ø/145.

O SPNL implementa oito técnicas para resolução de problemas do tipo descrito acima, através da combinação de métodos para pesquisa restrita, irrestrita e unidimensional. Essas combinações estão expressas na Tabela 1 e também na Figura 1, onde cada ramo completo compõe um algoritmo, que deve ser selecionado pelo usuário.

Combinação	Otimização c/ RESTRIÇÃO	Otimização IRRESTRITA	Pesquisa LINEAR
111	MPI	DFP	I.C.
112	MPI	DFP	S.A.
121	MPI	FR	I.C.
122	MPI	FR	S.A.
211	MPE	DFP	I.C.
212	MPE	DFP	S.A.
221	MPE	FR	I.C.
222	MPE	FR	S.A.

Tabela 1. Algoritmos usados.

MPI - Métodos de pontos interiores (barreira)
MPE - Método de pontos exteriores (penalidade)
DFP - Davidon, Fletcher e Powell
FR - Fletcher e Reeves
IC - Interpolação cúbica (Davidon)
SA - Seção áurea

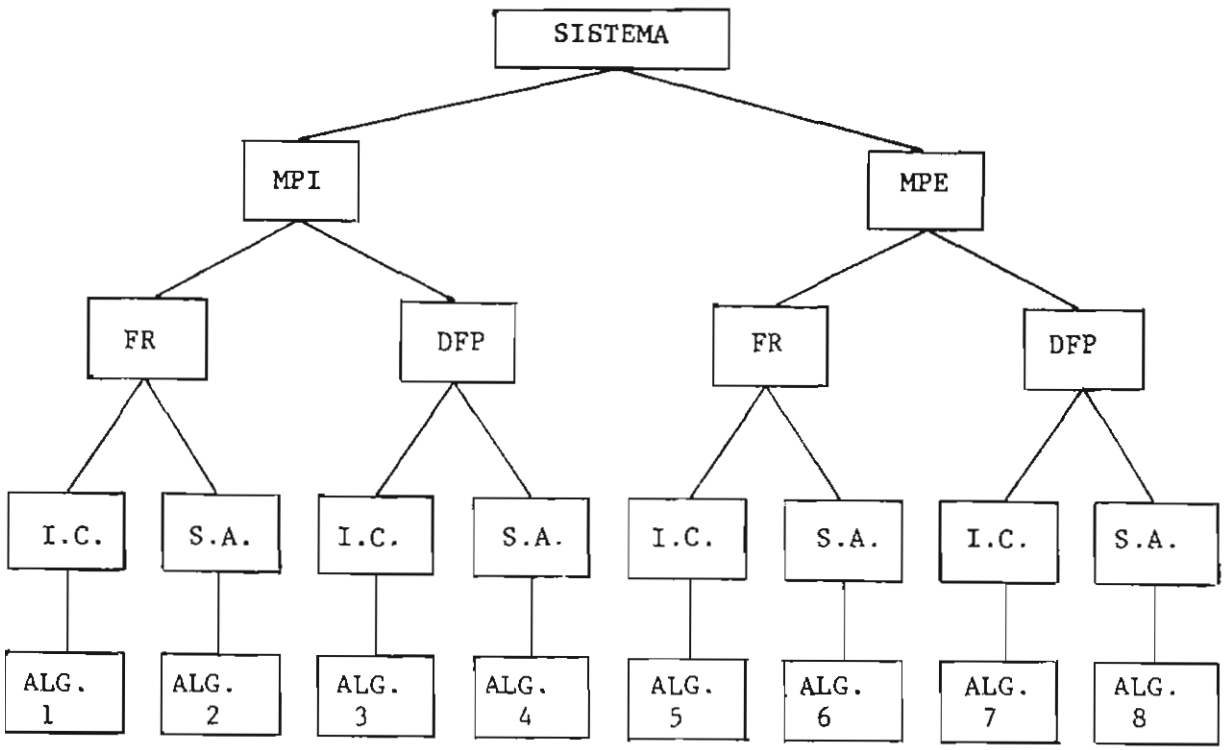


Figura 1. Esquema de composição dos algoritmos.

Internamente ao SPNL, todos os cálculos são feitos com dupla precisão (REAL*8). Uma descrição detalhada dos algoritmos, incluindo especificações da implementação e também uma comparação dos mesmos pode ser encontrada na referência citada.

2. UTILIZAÇÃO

2.1. Introdução.

Ao usar o sistema, o usuário deve escrever três pequenas rotinas: um programa principal dimensionando áreas de trabalho e chamando a rotina de controle, uma subrotina FORTRAN descrevendo as restrições e uma função FORTRAN descrevendo a função objetiva.

2.2. O programa principal.

A função do programa principal é dimensionar as áreas de trabalho e chamar a rotina de controle, passando como argumentos o nome da função e da subrotina e as áreas dimensionadas. Cada chamada da rotina de controle faz com que um conjunto de dados seja lido e o algoritmo especificado no conjunto de dados seja usado para resolver o problema definido pela subrotina e função fornecidas pelo usuário.

Através do comando REAL*8, o usuário deve dimensionar dez áreas, sendo que oito delas com n elementos cada uma, uma com m elementos e uma com n^2 elementos, onde n é o número de variáveis do problema e m é o número de restrições.

Adicionalmente, deve ser declarado que a função FORTRAN que define a função objetiva é de dupla precisão, o que deve ser feito também através do comando REAL*8. A função FORTRAN que define a função objetiva e o nome da subrotina que define as restrições devem também aparecer em um comando EXTERNAL.

O nome da rotina de controle é $MONIT e o usuário deve chamá-la através do comando CALL. Os argumentos que devem ser passados aparecem após o nome da rotina, entre parênteses, separados por vírgulas, na seguinte ordem: nome da função que define a função obje

tiva, nome da subrotina definindo as restrições, nome da área 1,.., nome da área 1Ø. A área 1Ø deve ser a que contém n^2 elementos e a área 9 a que contém m elementos.

A rotina de controle pode ser chamada tantas vezes quantas se queira. Em cada chamada o mesmo problema será resolvido com novos parâmetros fornecidos no conjunto de dados relativos àquela execução ou um novo problema definido pelos subprogramas com outros nomes será resolvido. De qualquer forma, cada vez que $MONIT é chamada, um novo conjunto de dados é lido.

Os dois últimos comandos do programa principal são , como usual, o STOP e o END.

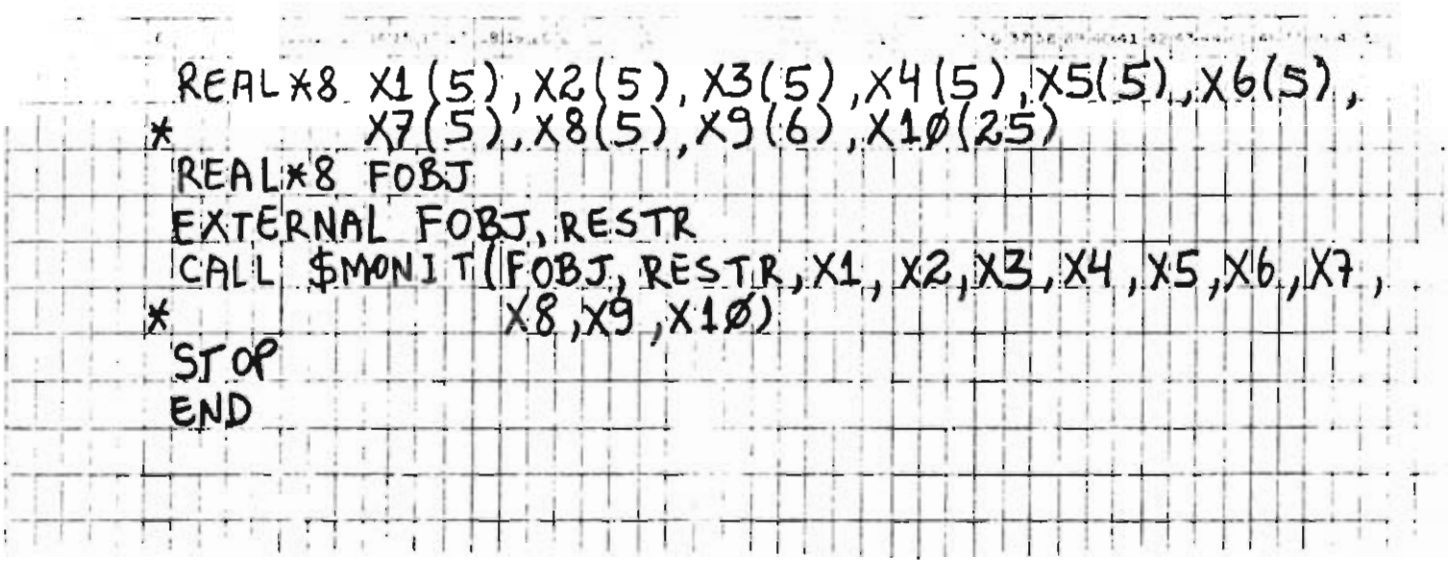
Se a precisão dos resultados impressos não for satisfatória, o usuário poderá requerer que seja gerado um arquivo contendo informações parciais com toda a precisão com que foram calculados. Isso pode ser conseguido incluindo-se o seguinte comando no programa principal:

```
COMMON/EXTEND/IXT,NARQ,NPROB
```

O arquivo será gerado na unidade definida por NARQ.Se NARQ=xx, necessariamente deverá existir o cartão de " Job Control Language" (JCL) //FTxxFØØ1 definindo o arquivo. NPROB pode ser qualquer número inteiro que servirá apenas para identificar os registros gerados e IXT deve ser feito igual a 1 para que o arquivo seja gerado para aquele problema específico.

Exemplos:

a) Resolver um problema com 5 variáveis e 6 restrições. A função objetiva está definida pelo subprograma chamado FOBJ e as restrições pela subrotina chamada RESTR:



```
      REAL*8 X1(5),X2(5),X3(5),X4(5),X5(5),X6(5),
     *       X7(5),X8(5),X9(6),X10(25)
      REAL*8 FOBJ
      EXTERNAL FOBJ,RESTR
      CALL $MONIT(FOBJ,RESTR,X1,X2,X3,X4,X5,X6,X7,
     *            X8,X9,X10)
      STOP
      END
```

b) Resolver um problema com 3 variáveis e 9 restrições usando dois algoritmos. A função objetiva está definida pela função FORTRAN chamada FX e as restrições pela subrotina R. Na resolução desse problema deve ser gerado um arquivo com informações precisas. Este mesmo programa principal deve ser usado para resolver outro problema definido pela função objetiva que tem nome OBJ e restrições chamadas REGIAO. A dimensão desse problema é de 4 variáveis e 4 restrições. Não se deseja que o arquivo contenha dados da resolução desse problema

A estratégia adotada no dimensionamento consiste em dimensionar de acordo com o máximo de variáveis e o máximo de restrições dos dois problemas. Assim:

```
      REAL*8 A(4),B(4),C(4),D(4),E(4),F(4),G(4),
     *       H(4),I(9),J(16)
      REAL*8 FX,OBJ
      EXTERNAL FX,OBJ,R,REGIAO
      COMMON /EXTEND/ IXT,NARG,NPROB
      NARG = 15
      NPROB = 1
      IXT = 1
      CALL $MONIT(FX,R,A,B,C,D,E,F,G,H,I,J)
      CALL $MONIT(FX,R,A,B,C,D,E,F,G,H,I,J)
      IXT = 0
      CALL $MONIT(OBJ,REGIAO,A,B,C,D,E,F,G,H,I,J)
      STOP
      END
```

No exemplo acima, no arquivo definido pelo cartão "DD" FT15F001, serão gerados registros contendo informações precisas e detalhadas sobre a resolução dos problemas definidos pela função FX e conjunto de restrições R. A resolução do problema definido pela função OBJ e conjunto de restrições REGIAO não causará a geração de qualquer registro em FT15F001, pois IXT foi feito igual a zero . Detalhe sobre o formato dos registros gerados podem ser vistos no Capítulo 3.

2.3. A função objetiva.

A função objetiva deve ser definida por meio de um subprograma FUNCTION FORTRAN. Seu propósito é avaliar o valor da função objetiva no ponto que é passado como único argumento da função.

É a seguinte a estrutura deste subprograma:

```
REAL FUNCTION nome-da-função*8 (nome-da-área)
REAL*8 nome-da-área (1)
nome-da-função=expressão
RETURN
END
```

Exemplo:

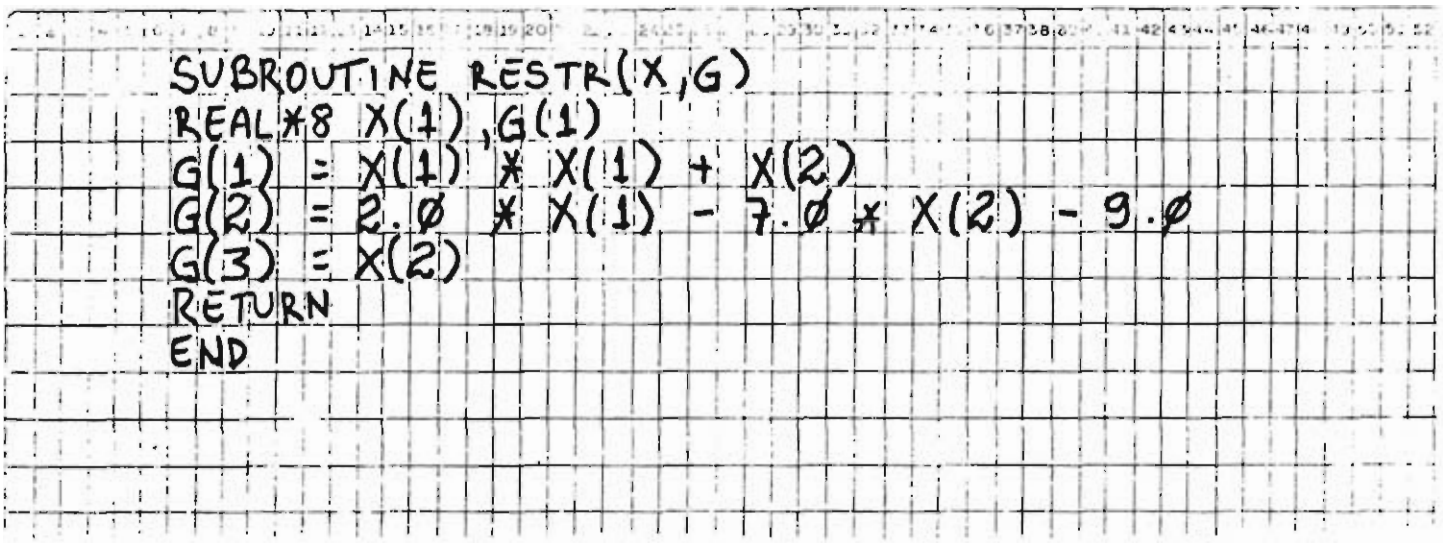
```
REAL FUNCTION F*8(X)
REAL*8 X(1)
F = 2.Ø * X(1) * X(1) - 3.Ø * X(2) ** 3
RETURN
END
```

2.4. As restrições.

As restrições do problema de programação não linear são informadas ao SPNL por meio de uma subrotina FORTRAN, cujos dois únicos argumentos são o nome da área contendo as coordenadas do ponto no qual se deseja calcular o valor das restrições e o nome da área que conterá os valores das restrições, calculadas no ponto fornecido. Sua estrutura é a seguinte:

```
SUBROUTINE nome-da-subrotina(nome-da-área 1, nome-da-área 2)
REAL*8 nome-da-área-1(1),nome-da-área-2(1)
nome-da-área-2(1) = expressão-1
nome-da-área-2(2) = expressão-2
    .
    .
    .
nome-da-área-2(m)=expressão-m
RETURN
END
```

Exemplo:



```
      SUBROUTINE RESTR(X,G)
      REAL*8 X(1),G(1)
      G(1) = X(1) * X(1) + X(2)
      G(2) = 2.Ø * X(1) - 7.Ø * X(2) - 9.Ø
      G(3) = X(2)
      RETURN
      END
```

2.5. Os cartões de dados.

Existem três tipos de cartões de dados, todos obrigatórios: (a) cartão de título; (b) cartão de parâmetros e (c) cartões de ponto inicial. Nenhum cartão tem identificação própria e devem ser fornecidos na ordem acima. Cada chamada da rotina de controle $MONIT faz com que um conjunto dos três tipos de cartões seja lido.

2.5.1. Cartão de título.

O cartão de título serve para identificar o problema a ser resolvido e pode conter, nas colunas de 1 até 8Ø qualquer combinação de caracteres.

2.5.2. Cartão de parâmetros.

Todas as informações contidas no cartão de parâmetros, exceto número de variáveis e número de restrições, podem ser omitidas desde que o usuário aceite os valores que o sistema assume por omissão. Os valores do cartão de parâmetros devem ser ajustados pela direita de cada campo, ou deve ser explicitado um ponto decimal.

O formato é o seguinte:

Col.1 : Cód. do algoritmo para otimização com restrição:

1 = MPI (assumido por omissão),
2 = MPE

Col.2 : Cód. do algoritmo para otimização irrestrita:
1 = DFP (assumido por omissão),
2 = FR

Col.3 : Cód. do algoritmo para pesquisa unidimensional:
1 = INT. CUB.(assumido por omissão),
2 = SEÇÃO ÁUREA

Col.5-7 : número variáveis (≥ 2),

Col.8-1Ø : número de restrições.

Col.11-2Ø : R - constante para pertubação inicial.

Col.21-3Ø : C - constante para alteração de R. Se for usado o MPE, $1Ø^{-2} < C < 1$; se for usado o MPI, $C > 1$ ($C=1Ø$, por omissão, para o MPI e $C=Ø.1$, por omissão, para o MPE)

Col.31-4Ø : FE - Estimativa inicial para o valor mínimo da função ($FE=Ø$, por omissão).

Col.41-5Ø : APA - Estimativa de "passo significativo" para o problema ($APA=1$, por omissão).

Col.51-6Ø : TOL-Critério de convergência. Tolerância para a diferença entre valores consecutivos da função($TOL=1Ø^{-3}$, por omissão)

Col.61-7Ø : EPS - Critério de convergência. Tolerância para o módulo do gradiente da função objetiva ($EPS=1Ø^{-3}$, por omissão)

Col.71-8Ø : H - incremento "infinitesimal" usado para o cálculo numérico do gradiente da função ($H=1Ø^{-6}$, por omissão).

2.5.3. Cartões de ponto inicial.

Os cartões de ponto inicial são lidos no formato 8D1Ø.4, e serão lidos tantos valores quantas forem as variáveis do problema, informadas no cartão de parâmetros. Para problemas com até oito variáveis, existirá apenas um cartão de ponto inicial.

3. RESULTADOS

Ao fim de cada iteração completa da pesquisa linear , da otimização irrestrita e da otimização com restrição, é impressa uma linha identificada respectivamente com $LINE, $MULTI e $SUMT.Ao final do processo, se é encontrada uma solução para o problema, é impresso um sumário final. Todo resultado impresso é autoidentificado, conforme pode ser observado no exemplo do capítulo seguinte .

Conforme descrito na Seção 2.2., o usuário poderá solicitar que seja gerada uma fita contendo resultados detalhados e precisos. São gerados 13 tipos de registros, com todas as informações gravadas no formato binário. Na descrição abaixo, é usada a seguinte codificação:

TIPO = tipo do registro (inteiro 4 bytes)
TÍTULO = título do problema (8Ø bytes)
NPROB = número do problema, como indicado no programa principal (inteiro,4 bytes)
LC = cód. alg. otimização com restrição (inteiro, 4 bytes)
LS = cód. alg. otimização sem restrição (inteiro, 4 bytes)
LU = cód. alg. pesquisa linear (inteiro, 4 bytes)
R = ver descrição no cartão de parâmetros (real, 8 bytes)
C = ver descrição no cartão de parâmetros (real, 8 bytes)
EPS = ver descrição no cartão de parâmetros (real, 8 bytes)
TOL = ver descrição no cartão de parâmetros (real, 8 bytes)
x_i^k = i-ésima componente do ponto final da iteração k (real, 8 bytes)
FVAL = valor da função no ponto X^k (real, 8 bytes)
ITER = número da iteração (inteiro, 4 bytes)
NAFO = número de avaliações da função objetiva, na etapa concluída (inteiro, 4 bytes)

TEMPO = tempo em microssegundos (inteiro, 4 bytes)

Registro tipo 1 - TIPO, TITULO.

Registro tipo 2 - TIPO, NPROB, LC,LS,LU,N,M,R,C,EPS,TOL.

Registro tipo 3 - TIPO, X_i^0 ,i=1,...,n.

Registro tipo 4 - TIPO,R,FVAL,ITER,NAFO,TEMPO - gerado ao final da pesquisa linear.

Registro tipo 5 - TIPO, X_i^k, i=1,...n - gerado ao final da pesquisa linear.

Registro tipo 6 - TIPO,R,FVAL,ITER,NAFO,TEMPO - gerado ao final de otimização irrestrita.

Registro tipo 7 - TIPO, X_i^k, i=1,...n - gerado ao final da otimização irrestrita.

Registro tipo 8 - TIPO,R,FVAL,ITER,NAFO,TEMPO - gerado ao final de cada iteração da otimização com restrição.

Registro tipo 9 - TIPO, X_i^k, i=1,...n - gerado ao final de cada iteração da otimização com restrição.

Registro tipo 1Ø- TIPO,R,FVAL,ITER,NAFO,TEMPO - resultado final.

Registro tipo 11- TIPO, X_i^k, i=1,...n - ponto final.

Registro tipo 12- TIPO,R,FVAL,ITER,NAFO,TEMPO - resultado final: não convergiu.

Registro tipo 13- TIPO, X_i^k, i=1,...n - resultado final: não convergiu.

Os registros 3,5,7,9,11 e 13 são de tamanho variável , dependendo da dimensão do problema. Uma possível especificação para o DCB do arquivo onde são gravados os registros acima seria DCB=(RECFM=VBS,LRECL=r, BLKSIZE=b), onde r=max (84,8*n+4) e b=kr+4 , n é o número de variáveis do problema e k é um número inteiro.

4. EXEMPLO COMPLETO

O exemplo (Rosen e Suzuki) a seguir ilustra todos os cartões necessários para resolver o seguinte problema:

$$\min f(x) = x_1^2+x_2^2+2x_3^2+x_4^2-5x_1-5x_2-21x_3+7x_4$$

sujeito a

$$x_1^2+x_2^2+x_3^2+x_4^2+x_1-x_2+x_3-x_4 \leq 8$$

$$x_1^2+2x_2^2+x_3^2+2x_4^2-x_1-x_4 \leq 10$$

$$2x_1^2+x_2^2+x_3^2+2x_1-x_2-x_4 \leq 5$$

O resultado exato é $x=(0,1,2,-1)$. Para usar o SPNL, em primeiro lugar é necessário preparar as restrições para tê-las na forma apropriada:

$$8-x_1^2-x_2^2-x_3^2-x_4^2-x_1+x_2-x_3+x_4 \geq 0$$

$$10-x_1^2-2x_2^2-x_3^2-2x_4^2+x_1+x_4 \geq 0$$

$$5-2x_1^2-x_2^2-x_3^2-2x_1+x_2+x_4 \geq 0$$

Observe-se neste exemplo o nome do "step" e o "ddname" usados na compilação das rotinas fornecidas pelo usuário(FORT.PROB) e o "step" e "ddname" onde vão os dados (GO.DATA).

EMBRAPA

GABARITO DE PERFURAÇÃO

1/2

SISTEMA

SUB-SISTEMA

CURSO

CÓDIGO

EXEMPLO DO SPNL

DATA

PROGRAMADOR: MOACIR

```
//NOMEJOB JOB , etc.
//  EXEC SPNL
//FORT.PROB DD *
C
C-----PROGRAMA PRINCIPAL
C
      REAL*8 XA(4), XB(4), GA(4), GB(4), D(4), A(4), G(3),
     *       B(4), S(16), Q(4)
      REAL*8 FOBJ
      EXTERNAL FOBJ, REST
      CALL $MONIT(FOBJ,REST,XA,XB,GA,GB,D,A,B,Q,G,S)
      STOP
      END
      REAL FUNCTION FOBJ*8(X)
C
C-----FUNCAO OBJETIVA
C
      REAL*8 X(1)
      FOBJ = X(1)**2+X(2)**2+2*X(3)**2+X(4)**2-5*X(1)
     *       -5*X(2)-21*X(3)+7*X(4)
      RETURN
      END
```

DMQ/001-1

EMBRAPA

GABARITO DE PERFURAÇÃO

1/2

SISTEMA | CURSO | EXEMPLO DO SPNL | PERFURADO

SUB-SISTEMA | CÓDIGO | DATA | SIM

PROGRAMADOR: MOACIR | DATA PERFURADO

```
//NOMEJOB JOB , etc.
//  EXEC SPNL
//FORT.PROB DD *
C
C-----PROGRAMA PRINCIPAL
C
      REAL*8 XA(4),XB(4),GA(4),GB(4),D(4),A(4),G(3),
     *       B(4),S(16),Q(4)
      REAL*8 FOBJ
      EXTERNAL FOBJ,REST
      CALL $MONIT(FOBJ,REST,XA,XB,GA,GB,D,A,B,Q,G,S)
      STOP
      END
      REAL FUNCTION FOBJ*8(X)
C
C-----FUNCAO OBJETIVA
C
      REAL*8 X(1)
      FOBJ = X(1)**2+X(2)**2+2*X(3)**2+X(4)**2-5*X(1)
     *       -5*X(2)-21*X(3)+7*X(4)
      RETURN
      END
```

DMQ/001-1

EMBRAPA | SISTEMA | CODIGO | DATA | EXEMPLO DO SPNL | ALTERADO SIM

PROGRAMADOR | OBSERVAÇÕES:

```
      SUBROUTINE REST(X,G)
C
C-----DEFINICAO DAS RESTRICOES
C
      REAL*8 X(1),G(1)
      G(1) = 8-X(1)**2-X(2)**2-X(3)**2-X(4)**2-X(1)
     *          +X(2)-X(3)+X(4)
      G(2) = 10-X(1)**2-2*X(2)**2-X(3)**2-2*X(4)
     *          +X(1)+X(4)
      G(3) = 5-2*X(1)**2-X(2)**2-X(3)**2-2*X(1)+X(2)
     *          +X(4)
      RETURN
      END
/*
//GO.DATA DD *
EXEMPLO DO SPNL - ROSEN & SUZUKI
112    4  3
    0.0       0.0
/*
//
```

****

S I S T E M A D E O T I M I Z A C A O N A O L I N E A R

****

TITULO : EXEMPLO DO SPNL - ROSEN E SUZUKI

OTIMIZACAO COM RESTRICAO - ALGORITMO : INTERIOR (BARREIRA)

OTIMIZACAO SEM RESTRICAO - ALGORITMO : DAVIDON-FLETCHER-POWELL (DFP)

OTIMIZACAO UNIDIMENSIONAL - ALGORITMO : SECAO AUREA

PARAMETROS :
R (PERTURBACAO INICIAL : 0.10000 02
C (CTE. P/ ALTERACAO DE R) : 0.10000 02

CONDICOES DE CONVERGENCIA :
EPS (P/ MODULO DO GRADIENTE) : 0.10000-02
TOL (P/ CI - F(X1 - X2) : 0.10000-02

DIMENSAO DO PROBLEMA :
NUMERO DE VARIAVEIS : 4
NUMERO DE RESTRICOES : 3

PONTO INICIAL :

X(1)= 0.0 X(2)= 0.0 X(3)= 0.0 X(4)= 0.0 X(

```
.....$LINE..... # DE ITERACOES= 14;  # AVAL. F. OBJ.=  52;  TEMPO=      172709 US;  F=-0.335110 02
.....$LINE..... # DE ITERACOES= 13;  # AVAL. F. OBJ.=  32;  TEMPO=      109582 US;  F=-0.349580 02
.....$LINE..... # DE ITERACOES= 12;  # AVAL. F. OBJ.=  30;  TEMPO=      104567 US;  F=-0.356280 02
.....$LINE..... # DE ITERACOES= 12;  # AVAL. F. OBJ.=  31;  TEMPO=      105624 US;  F=-0.357470 02
.....$LINE..... # DE ITERACOES= 13;  # AVAL. F. OBJ.=  32;  TEMPO=      110651 US;  F=-0.356740 02
.....$LINE..... # DE ITERACOES= 12;  # AVAL. F. OBJ.=  30;  TEMPO=      106640 US;  F=-0.357360 02
.....$LINE..... # DE ITERACOES= 13;  # AVAL. F. OBJ.=  32;  TEMPO=      111849 US;  F=-0.357230 02
.....$LINE..... # DE ITERACOES= 13;  # AVAL. F. OBJ.=  32;  TEMPO=      109244 US;  F=-0.357270 02
.....$LINE..... # DE ITERACOES= 13;  # AVAL. F. OBJ.=  32;  TEMPO=      110572 US;  F=-0.356980 02
.....$MULTI..... # DE ITERACOES=  9;  # AVAL. F. OBJ.= 353;  TEMPO=     1726041 US;  F=-0.357090 02;  |GRAD|= 0.64489D-01
.....$SUMT..... SUPERFICIE #   1;  R= 0.100000 02;  PENALIDADE= 0.103260 02;  F=-0.357090 02

                X(  1)=-0.464770-01 X(  2)= 0.792200 00 X(  3)= 0.166560 01 X(  4)=-0.512320 00 X(

.....$LINE..... # DE ITERACOES= 11;  # AVAL. F. OBJ.=  28;  TEMPO=      101432 US;  F=-0.410730 02
.....$LINE..... # DE ITERACOES= 12;  # AVAL. F. OBJ.=  30;  TEMPO=      107499 US;  F=-0.410320 02
.....$LINE..... # DE ITERACOES= 10;  # AVAL. F. OBJ.=  26;  TEMPO=       94661 US;  F=-0.413910 02
.....$LINE..... # DE ITERACOES= 12;  # AVAL. F. OBJ.=  31;  TEMPO=      107942 US;  F=-0.414710 02
.....$LINE..... # DE ITERACOES= 10;  # AVAL. F. OBJ.=  26;  TEMPO=       93203 US;  F=-0.415130 02
.....$LINE..... # DE ITERACOES= 13;  # AVAL. F. OBJ.=  32;  TEMPO=      110209 US;  F=-0.414660 02
.....$LINE..... # DE ITERACOES= 12;  # AVAL. F. OBJ.=  30;  TEMPO=      111598 US;  F=-0.414690 02
.....$LINE..... # DE ITERACOES= 12;  # AVAL. F. OBJ.=  30;  TEMPO=      106796 US;  F=-0.414700 02
.....$MULTI..... # DE ITERACOES=  9;  # AVAL. F. OBJ.= 279;  TEMPO=     1144937 US;  F=-0.414700 02;  |GRAD|= 0.21433D-01
.....$SUMT..... SUPERFICIE #   2;  R= 0.100000 01;  PENALIDADE= 0.284611 01;  F=-0.414700 02

                X(  1)=-0.152330-01 X(  2)= 0.812540 00 X(  3)= 0.189890 01 X(  4)=-0.835800 00 X(

.....$LINE..... # DE ITERACOES=  9;  # AVAL. F. OBJ.=  24;  TEMPO=       90208 US;  F=-0.431090 02
.....$LINE..... # DE ITERACOES=  9;  # AVAL. F. OBJ.=  24;  TEMPO=       89953 US;  F=-0.432740 02
.....$LINE..... # DE ITERACOES= 11;  # AVAL. F. OBJ.=  28;  TEMPO=       98330 US;  F=-0.432060 02
.....$LINE..... # DE ITERACOES= 11;  # AVAL. F. OBJ.=  29;  TEMPO=      101666 US;  F=-0.432360 02
.....$LINE..... # DE ITERACOES= 13;  # AVAL. F. OBJ.=  32;  TEMPO=      112291 US;  F=-0.432070 02
.....$LINE..... # DE ITERACOES=  9;  # AVAL. F. OBJ.=  24;  TEMPO=       86614 US;  F=-0.432170 02
.....$LINE..... # DE ITERACOES= 10;  # AVAL. F. OBJ.=  26;  TEMPO=       95416 US;  F=-0.432180 02
.....$LINE..... # DE ITERACOES=  9;  # AVAL. F. OBJ.=  24;  TEMPO=       96510 US;  F=-0.432280 02
```

```
.....$LINE...... # DE ITERACOES=  7;  # AVAL. F. OBJ.=  20;  TEMPO=      76432 US;  F=-0.432480 02
.....$MULTI..... # DE ITERACOES=  9;  # AVAL. F. OBJ.= 231;  TEMPO=    1202916 US;  F=-0.432390 02;  |GRAD|= 0.539840 00
.....$SUMT...... SUPERFICIE  #   3;  R= 0.100000 00;  PENALIDADE= 0.948840 00;  F=-0.432390 02

                 X(  1)=-0.802240-02 X(  2)= 0.971410 00 X(  3)= 0.196770 01 X(  4)=-0.957970 00 X;

.....$LINE...... # DE ITERACOES=  7;  # AVAL. F. OBJ.=  20;  TEMPO=      79322 US;  F=-0.437460 02
.....$LINE...... # DE ITERACOES=  6;  # AVAL. F. OBJ.=  18;  TEMPO=      70192 US;  F=-0.437680 02
.....$LINE...... # DE ITERACOES= 11;  # AVAL. F. OBJ.=  28;  TEMPO=     101510 US;  F=-0.437510 02
.....$LINE...... # DE ITERACOES= 10;  # AVAL. F. OBJ.=  26;  TEMPO=      93359 US;  F=-0.437580 02
.....$LINE...... # DE ITERACOES=  6;  # AVAL. F. OBJ.=  18;  TEMPO=      74895 US;  F=-0.437650 02
.....$MULTI..... # DE ITERACOES=  5;  # AVAL. F. OBJ.= 141;  TEMPO=     620911 US;  F=-0.437630 02;  |GRAD|= 0.561700 00
.....$SUMT...... SUPERFICIE  #   4;  R= 0.100000-01;  PENALIDADE= 0.255510 00;  F=-0.437630 02

                 X(  1)=-0.357270-02 X(  2)= 0.999310 00 X(  3)= 0.198990 01 X(  4)=-0.989330 00 X;

.....$LINE...... # DE ITERACOES=  4;  # AVAL. F. OBJ.=  14;  TEMPO=      62526 US;  F=-0.439240 02
.....$LINE...... # DE ITERACOES=  8;  # AVAL. F. OBJ.=  22;  TEMPO=      82109 US;  F=-0.439180 02
.....$LINE...... # DE ITERACOES=  4;  # AVAL. F. OBJ.=  14;  TEMPO=      60260 US;  F=-0.439230 02
.....$LINE...... # DE ITERACOES=  9;  # AVAL. F. OBJ.=  24;  TEMPO=      91432 US;  F=-0.439230 02
.....$LINE...... # DE ITERACOES=  4;  # AVAL. F. OBJ.=  14;  TEMPO=      62786 US;  F=-0.439330 02
.....$MULTI..... # DE ITERACOES=  5;  # AVAL. F. OBJ.= 119;  TEMPO=     553775 US;  F=-0.439290 02;  |GRAD|= 0.252030 01
.....$SUMT...... SUPERFICIE  #   5;  R= 0.100000-02;  PENALIDADE= 0.835400-01;  F=-0.439290 02

                 X(  1)=-0.117240-02 X(  2)= 0.998650 00 X(  3)= 0.199650 01 X(  4)=-0.997010 00 X;

.....$LINE...... # DE ITERACOES=  2;  # AVAL. F. OBJ.=  10;  TEMPO=      51328 US;  F=-0.439800 02
.....$LINE...... # DE ITERACOES=  2;  # AVAL. F. OBJ.=  10;  TEMPO=      52656 US;  F=-0.439770 02
.....$LINE...... # DE ITERACOES=  3;  # AVAL. F. OBJ.=  12;  TEMPO=      54973 US;  F=-0.439770 02
.....$MULTI..... # DE ITERACOES=  3;  # AVAL. F. OBJ.=  53;  TEMPO=     374859 US;  F=-0.439760 02;  |GRAD|= 0.837020 00
.....$SUMT...... SUPERFICIE  #   6;  R= 0.100000-03;  PENALIDADE= 0.245660-01;  F=-0.439760 02

                 X(  1)=-0.170270-03 X(  2)= 0.999300 00 X(  3)= 0.199910 01 X(  4)=-0.998130 00 X;
```

```
.....$LINE..... # DE ITERACOES=  3;  # AVAL. F. OBJ.=  12;  TEMPO=      59895 US;  F=-0.439920 02
.....$LINE..... # DE ITERACOES=  1;  # AVAL. F. OBJ.=   8;  TEMPO=      43906 US;  F=-0.439930 02
.....$MULTI..... # DE ITERACOES=  2;  # AVAL. F. OBJ.=  36;  TEMPO=     196432 US;  F=-0.439930 02;  |GRAD|= 0.143630 01
.....$SUMT..... SUPERFICIE  #   7;  R= 0.100000-04;  PENALIDADE= 0.795280-02;  F=-0.439930 02


                X(  1)=-0.410610-03 X(  2)= 0.999420 00 X(  3)= 0.200000 01 X(  4)=-0.999210 00 X(


.....$LINE..... # DE ITERACOES=  1;  # AVAL. F. OBJ.=   8;  TEMPO=      47291 US;  F=-0.439960 02
.....$LINE..... # DE ITERACOES=  1;  # AVAL. F. OBJ.=   8;  TEMPO=      50624 US;  F=-0.439970 02
.....$MULTI..... # DE ITERACOES=  2;  # AVAL. F. OBJ.=  32;  TEMPO=     176458 US;  F=-0.439960 02;  |GRAD|= 0.757380 01
.....$SUMT..... SUPERFICIE  #   8;  R= 0.100000-05;  PENALIDADE= 0.169550-02;  F=-0.439960 02


                X(  1)=-0.559530-03 X(  2)= 0.999430 00 X(  3)= 0.200020 01 X(  4)=-0.999580 00 X(


.....$LINE..... # DE ITERACOES=  1;  # AVAL. F. OBJ.=   8;  TEMPO=      49270 US;  F=-0.439980 02
.....$LINE..... # DE ITERACOES=  1;  # AVAL. F. OBJ.=   8;  TEMPO=      51822 US;  F=-0.439990 02
.....$MULTI..... # DE ITERACOES=  2;  # AVAL. F. OBJ.=  32;  TEMPO=     180546 US;  F=-0.439990 02;  |GRAD|= 0.700310 01
.....$SUMT..... SUPERFICIE  #   9;  R= 0.100000-06;  PENALIDADE= 0.519230-03;  F=-0.439990 02


                X(  1)=-0.309930-03 X(  2)= 0.999500 00 X(  3)= 0.200040 01 X(  4)=-0.999410 00 X(


CRITERIO DE CONVERGENCIA ATENDIDO
```

RESULTADOS FINAIS PARA EXEMPLO DO SPNL - ROSEN & SUZUKI

TEMPO TOTAL EM SEGUNDOS.....	6.1788
NUMERO DE AVALIACOES DA FUNCAO OBJETIVA.	1325
NUMERO DE SUPERFICIES OTIMIZADAS.....	9
MODULO DO GRADIENTE DA FUNCAO PENALIDADE	0.700310 01
VALOR DA PENALIDADE.....	0.519230-03
ULTIMO VALOR DE R.....	0.100000-06
VALOR FINAL DA FUNCAO OBJETIVA.....	-0.43999D 02

PONTO OTIMO :

X(1)=-0.309830-03 X(2)= 0.999500 00 X(3)= 0.200040 01 X(4)=-0.999410 00 X(

VALORES DAS RESTRICOES :

G(1)= 0.765860-03 G(2)= 0.100320 01 G(3)= 0.257370-03 G(

****

5. REFERÊNCIA

PEDROSO JR., MOACIR . "Comparação de oito algoritmos para problemas de programação não linear, com restrições". Dissertação de Mestrado, UnB 1977. Publicações do DMQ, série A,número 23.