EXEMPLO 7 — RESOLUÇAO DE SISTEMAS DE EQUAÇOES LINEARES PELO MÉTODO ITERATIVO DE GAUSS-SEIDEL.

Escrevamos o sistema de equações na forma

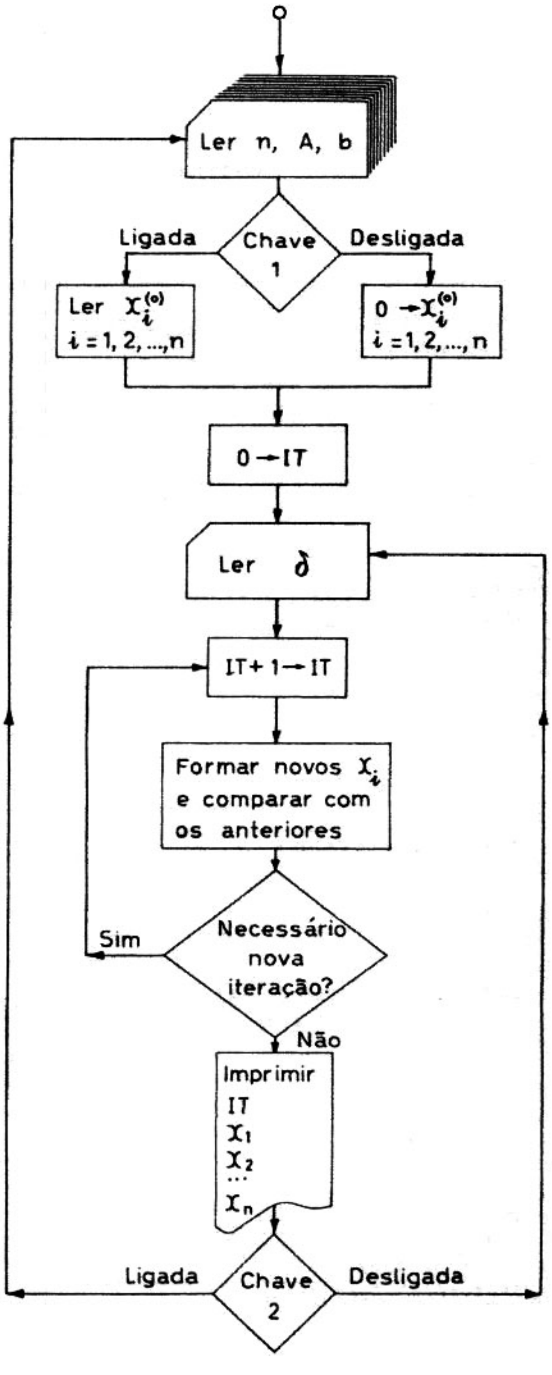
$$x_i = (b_i - \sum_{j \neq i} a_{ij} x_j)/a_{ii} \qquad i = 1(1)n$$

com todos os a_{ii} não nulos.

Partindo dum vector solução aproximado $(x_1^0, x_2^0, \dots, x_n^0)$ constroi-se nova aproximação $(x_1^1, x_2^1, \dots, x_n^1)$ pelo esquema

$$\begin{aligned}
x_1^1 &= (b_1 - a_{1,2}x_2^0 - a_{1,3}x_3^0 - a_{1,4}x_4^0 - . \quad - a_{1,n}x_n^0)/a_{1,1} \\
x_2^1 &= (b_2 - a_{2,1}x_1^1 - a_{2,3}x_3^0 - a_{2,4}x_4^0 - \dots - a_{2,n}x_n^0)/a_{2,2} \\
x_3^1 &= (b_3 - a_{3,1}x_1^1 - a_{3,2}x_2^1 - a_{3,4}x_4^0 - \dots - a_{3,n}x_n^0)/a_{3,3} \\
&\dots \\
x_n^1 &= (b_n - a_{n,1}x_1^1 - a_{n,2}x_2^1 - a_{n,3}x_3^1 - \dots - a_{n,n-1}x_{n-1}^1)/a_{n,n}
\end{aligned}$$

onde os x_i^1 são utilizados nos segundos membros logo que estejam calculados. Se $| x_i^1 - x_i^0 | < \delta$, $(\delta > 0,\ i = 1, 2, \dots n)$, com δ desprezável para o rigor exigido considera-se obtida a solução. Teremos $x_i \doteq x_i^1$. Em caso contrário os x_i^1 tomam o papel dos x_i^0 e calculam-se novos x_i^1; repete-se o processo até que, havendo convergência (ver [14]), se atinja a situação anterior.


Ler n, A, b
Chave 1
Ligada
Desligada
Ler $X_i^{(0)}$ $i = 1, 2, ..., n$
$0 \to X_i^{(0)}$ $i = 1, 2, ..., n$
$0 \to IT$
Ler δ
$IT + 1 \to IT$
Formar novos X_i e comparar com os anteriores
Necessário nova iteração?
Sim
Não
Imprimir
IT
X_1
X_2
...
X_n
Chave 2
Ligada
Desligada

```
C
C     SISTEMAS DE EQUACOES ALGEBRICAS LINEARES
C     METODO DE GAUSS-SEIDEL
C
      DIMENSIØN A(49,49),B(49),X(49)
C     LEITURAS-ORDEM,ELEMENTOS DA MATRIZ,TERMOS INDEPENDENTES
    1 READ 100,N
  100 FØRMAT(I3)
      READ 101,((A(I,J),J=1,N),I=1,N),(B(I),I=1,N)
  101 FØRMAT(E14.8)
C
      PRINT 102
  102 FØRMAT(50HLIGUE A CHAVE 1 SE DESEJA FØRNECER UMA APRØXIMACAØ/
     1  30HDØUTRA FØRMA TØMAR-SE-A X(I)=0)
      PAUSE
      IF(SENSE SWITCH 1)2,3
    2 READ 101,(X(I),I=1,N)
      GØ TØ 4
    3 DØ 5I=1,N
    5 X(I)=0.
C
    4 IT=0
C     IT-CONTADOR DE ITERACOES
    6 READ 101,DELTA
C
C     NOVA APROXIMACAO
    7 IT=IT+1
      IES=0
C     IES-CONTADOR DE SITUACOES ABSF(X1(I)-X0(I))MENOR QUE DELTA
C     SOLUCAO OBTIDA QUANDO IES=N
      DØ 8I=1,N
      S=B(I)
      DØ 9J=1,N
      IF(J-I)10,9,10
   10 S=S-A(I,J)*X(J)
    9 CØNTINUE
      S=S/A(I,I)
C     S CONTEM A NOVA COMPONENTE I
C     QUE SERA COMPARADA COM A ANTIGA
C     E TRANSMITIDA PARA X(I)
      IF(ABSF(S-X(I))-DELTA)11,8,8
   11 IES=IES+1
    8 X(I)=S
C
      IF(IES-N)7,13,12
   12 STØP
C
```

```
C     IMPRESSAO DOS RESULTADOS
   13 PRINT 103,IT,(I,X(I),I=1,N)
  103 FØRMAT(//3HIT=,I5//(2HX(,I3,2H)=,E14.8))
C
      PRINT 104
  104 FØRMAT(///38HCHAVE 2 LIGADA-RESØLVER ØUTRØ PRØBLEMA/5X
     145HDESLIGADA-LER NØVØ DELTA E MELHØRAR A SØLUCAØ)
      PAUSE
      IF(SENSE SWITCH2)1,6
      END
```

Para

$$A=\begin{bmatrix}6 & -2 & 1 & 0\\ 0 & -12 & 4 & 1\\ 1 & 2 & -7 & 4\\ 0 & -3 & 0 & 5\end{bmatrix} \qquad B=\begin{bmatrix}19\\ 33\\ -39\\ -2\end{bmatrix}$$

cuja solução é $x_1=2$ $x_2=-1$ $x_3=5$ $x_4=-1$, com chave 1 desligada e $\delta=10^{-6}$, o computador forneceu a resposta seguinte:

```
LIGUE A CHAVE 1 SE DESEJA FORNECER UMA APROXIMACAO
DOUTRA FORMA TOMAR-SE-A X(I)=0

IT=   16

X(  1)= .19999998E+01
X(  2)=-.10000001E+01
X(  3)= .49999995E+01
X(  4)=-.10000000E+01


CHAVE 2 LIGADA-RESOLVER OUTRO PROBLEMA
     DESLIGADA-LER NOVO DELTA E MELHORAR A SOLUCAO
```

Bibliografia: [14], [27], [31], [33].

EXEMPLO 8 — RAIZ CARACTERÍSTICA DE MÓDULO MÁXIMO DE UMA MATRIZ.

Seja A uma matriz de ordem n, com valores próprios λ_j tais que

$$|\lambda_1| > |\lambda_2| \geqslant \ \ldots \ \geqslant |\lambda_n| \qquad \text{com } \lambda_1 \text{ real.}$$

Partindo dum vector quase arbitrário v_0 (por exemplo, com todas as componentes iguais a 1), e utilizando as fórmulas de recorrência

$$\begin{aligned} z_k &= A v_{k-1} \\ v_k &= z_k / \mu_k \end{aligned} \qquad\qquad k = 1, 2, 3, \ldots$$

onde cada escalar μ_k é componente de módulo máximo de z_k (i. e. v_k fica com componentes de módulo entre 0 e 1 e componente máxima 1), prova-se que em geral μ_k tende para o valor próprio λ_1 e v_k para um vector próprio correspondente, bastando admitir que v_0 se pode exprimir por uma composição linear de vectores próprios x_j associados aos λ_j

$$(1) \qquad\qquad v_0 = \sum_{j=1}^{n} \alpha_j x_j \qquad \text{com } \alpha_1 \neq 0 .$$

Tomando um vector v_k da sucessão, temos

$$v_k = \frac{1}{\mu_k} z_k = \frac{1}{\mu_k} A v_{k-1} = \frac{1}{\mu_k} A \frac{1}{\mu_{k-1}} A \ldots \frac{1}{\mu_1} A v_0$$

ou, fazendo $C_k = 1/(\mu_k \, \mu_{k-1} \cdots \mu_1)$

e atendendo a (1) e a que $Ax_j = \lambda_j x_j$

$$v_k = C_k A^k v_0 = C_k \sum_{j=1}^{n} \alpha_j A^k x_j = C_k \sum_{j=1}^{n} \alpha_j \lambda_j^k x_j = C_k \lambda_1^k \sum_{j=1}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1}\right)^k x_j .$$

As razões λ_j/λ_1, com $j \neq 1$, decrescem para zero com o aumento de k, pelo que v_k tende a confundir-se com $C_k \lambda_1^k \alpha_1 x_1$.

Com o crescimento de k, o vector v_k torna-se pois paralelo a x_1, e a obrigação de ter componente máxima 1 acaba por lhe fixar o sentido e o módulo.

Em conclusão, v_k tende para um vector próprio (normalizado) associado a λ_1.

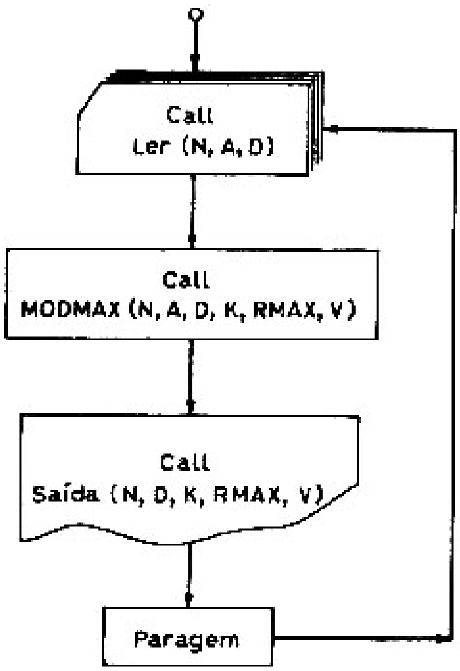
Comparando dois vectores da sucessão vizinhos do limite,

$$v_{k-1} \doteq C_{k-1} \lambda_1^{k-1} \alpha_1 x_1 \doteq v_k \doteq \frac{C_{k-1}}{\mu_k} \lambda_1^k \alpha_1 x_1$$

conclui-se que μ_k tende para λ_1.

Se v_0 tivesse $\alpha_1 = 0$ e $\alpha_2 \neq 0$, as proposições apresentadas deveriam transferir-se para λ_2, caso este fosse real e $|\lambda_2| > |\lambda_3|$.

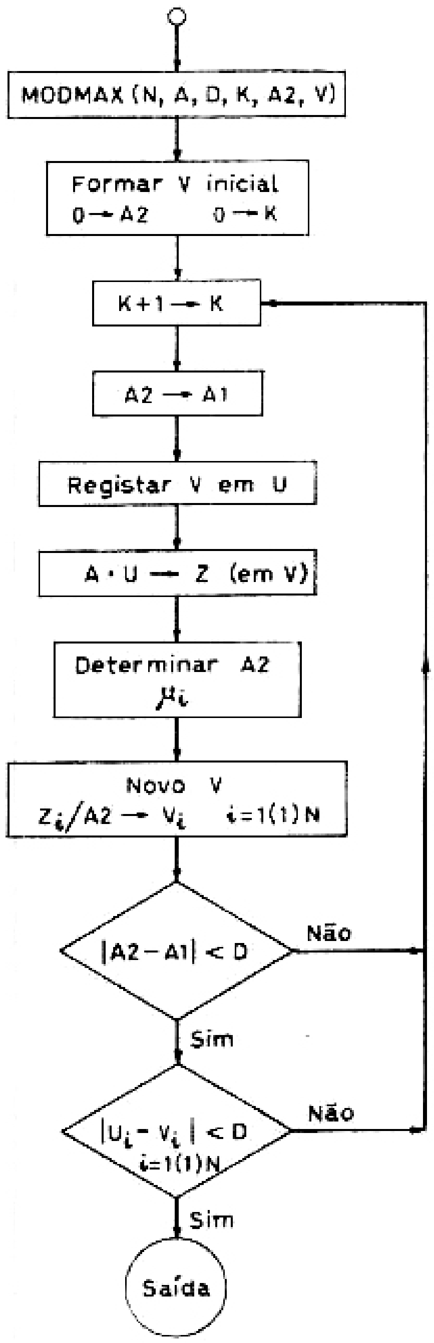
Notemos ainda que a matriz $A + \beta I$ tem valores próprios $\lambda_i' = \lambda_i + \beta$ e conserva os vectores próprios de A. Um β conveniente pode acentuar a dominância de λ_1', o que favorece a convergência, ou tornar dominante outro λ_i', permitindo calcular outra raiz característica.



```
C     PROGRAMA PRINCIPAL
C
      DIMENSIØN A(48,48),V(48)
    1 CALL LER(N,A,D)
      CALL MØDMAX(N,A,D,K,RMAX,V)
      CALL SAIDA(N,D,K,RMAX,V)
      PAUSE
      GØTØ 1
      END
```

```
      SUBRØUTINE LER(N,A,D)
C     LEITURA DA MATRIZ(ORDEM E ELEMENTOS)
C             E ERRO ADMISSIVEL(APROX.)
      DIMENSIØN A(48,48)
      READ1,N,((A(I,J),J=1,N),I=1,N),D
    1 FØRMAT(I5/(E14.8))
      RETURN
      END
```

```
      SUBRØUTINE SAIDA(N,D,K,RMAX,V)
      DIMENSIØN V(48)
      PRINT1,K,D,RMAX,(I,V(I),I=1,N)
    1 FØRMAT(/I5,10H ITERACØES,4H  D=E10.4//
     1  7H  RMAX=,E14.8//(2HV(,I3,2H)=,E14.8) )
      RETURN
      END
```


MODMAX (N, A, D, K, A2, V)
Formar V inicial
0 → A2 0 → K
K+1 → K
A2 → A1
Registar V em U
A · U → Z (em V)
Determinar A2
μ_i
Novo V
$Z_i/A2 \to V_i$ $i=1(1)N$
$|A2-A1| < D$
Não
Sim
$|U_i - V_i| < D$
$i=1(1)N$
Não
Sim
Saída

```
      SUBRØUTINE MØDMAX(N,A,D,K,A2,V)
      DIMENSIØN A(48,48),V(48)
      DIMENSIØN U(48)
C     V INICIAL
      DØ 1 I=1,N
    1 V(I)=1.
      A2=0.
      K=0
C     K-CONTADOR DE ITERACOES
    2 K=K+1
      A1=A2
C     COLOCAR V EM U
      DØ 3 I=1,N
    3 U(I)=V(I)
C     Z=A*U PARA V
      DØ 4 I=1,N
      VI=0.
      DØ 5 J=1,N
    5 VI=VI+A(I,J)*U(J)
    4 V(I)=VI
C
      A2=V(1)
      DØ 6 I=2,N
      IF(ABSF(A2)-ABSF(V(I)))7,6,6
    7 A2=V(I)
    6 CØNTINUE
C     A1 E A2 CONTEM AS COMPONENTES DE MODULO
C     MAXIMO DOS VECTORES Z DAS ITERACOES K-1 E K
C     (K MAIOR QUE 1)
C
C     NOVO V
      DØ 8 I=1,N
    8 V(I)=V(I)/A2
C     CONVERGENCIA DO VALOR PROPRIO
C     E VECTOR PROPRIO
      IF(ABSF(A2-A1)-D)9,2,2
    9 DØ 10 I=1,N
      IF(ABSF(V(I)-U(I))-D)10,2,2
   10 CØNTINUE
      RETURN
      END
```

$$\text{Com} \quad A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix} \quad \text{e} \quad \delta = 10^{-7}$$

o computador forneceu os seguintes resultados (número de iterações, D, λ_1 e vector próprio associado)

```
  9 ITERACOES   D= .1000E-06

 RMAX= .26304703E+02

V(  1)= .69680547E-01
V(  2)= .23290556E+00
V(  3)= .53034003E+00
V(  4)= .10000000E+01
```

Para $A + 5I$ obteve-se

```
 13 ITERACOES   D= .1000E-06

 RMAX= .31304703E+02

V(  1)= .69680549E-01
V(  2)= .23290556E+00
V(  3)= .53034002E+00
V(  4)= .10000000E+01
```

Bibliografia: [3], [10], [27].

EXEMPLO 9 — VALORES PRÓPRIOS DE MATRIZES TRIDIAGONAIS SIMÉTRICAS.

As matrizes reais simétricas têm valores próprios reais. Destas, as tridiagonais (com elementos todos nulos fora da diagonal principal e das duas que a envolvem) possuem uma propriedade que facilita extremamente a determinação numérica das suas raízes características.

Seja A uma matriz nessas condições, onde designaremos os elementos como mostra a figura junta

$$A = \begin{bmatrix} c_1 & b_1 & & & & \\ b_1 & c_2 & b_2 & & & \\ & b_2 & c_3 & b_3 & & \\ & & \cdots & \cdots & \cdots & \\ & & & b_{n-2} & c_{n-1} & b_{n-1} \\ & & & & b_{n-1} & c_n \end{bmatrix}$$

A sucessão de determinantes menores principais da matriz $A - \lambda I$ $D_1(\lambda), D_2(\lambda), \ldots, D_n(\lambda) = |A - \lambda I|$ a que juntaremos $D_0 = 1$ constitui uma sucessão de Sturm desde que $b_i \neq 0$ $(i = 1, 2, \ldots, n-1)$. Esta condição garante ainda valores próprios todos distintos.

Mostra-se então que o *número $K(\lambda)$ de concordâncias de sinal de termos consecutivos da sucessão $D_i(\lambda)$, para $\lambda = p$, nos dá o número de valores próprios superiores a p.*

Se a matriz A em estudo tiver algum $b_i = 0$, cortando a faixa diagonal por esse b_i obtemos duas matrizes tridiagonais que repartem entre si os valores próprios de A.

Não desejando fragmentar a matriz, é possível substituir os b_i nulos por uma quantidade suficientemente pequena de modo a não alterar sensivelmente os valores próprios. Wilkinson [36] utiliza o produto da precisão relativa do computador pela norma infinita da matriz, definida por $\max_i \sum_j |a_{ij}|$, alterando os resultados até cerca de duas vezes aquele produto (àparte erros de arredondamento).

O intervalo [—norma, +norma] contém todos os valores próprios (teorema de Frobenius, [3] pág. 75). Numeremos estes de modo que

$$\lambda_1 > \lambda_2 > \ldots > \lambda_n$$

Para determinar λ_i, tomemos um intervalo $[g,h]$, variável, que de início será [—norma, +norma]. Para $\lambda <$ —norma teriamos $K(\lambda)=n$, e para $\lambda \geqslant$ norma, $K(\lambda)=0$.

$K\left(\frac{g+h}{2}\right)$ indicará se λ_i está ou não à direita do ponto médio do intervalo, conforme for ou não for $K \geqslant i$; em caso afirmativo muda-se o extremo esquerdo g para o antigo médio; doutra forma, muda-se o extremo h.

Esquemàticamente

$$\left.\begin{array}{l} K(g) \geqslant i \\ K(h) < i \end{array}\right\} \Rightarrow \lambda_i \in [g,h]$$

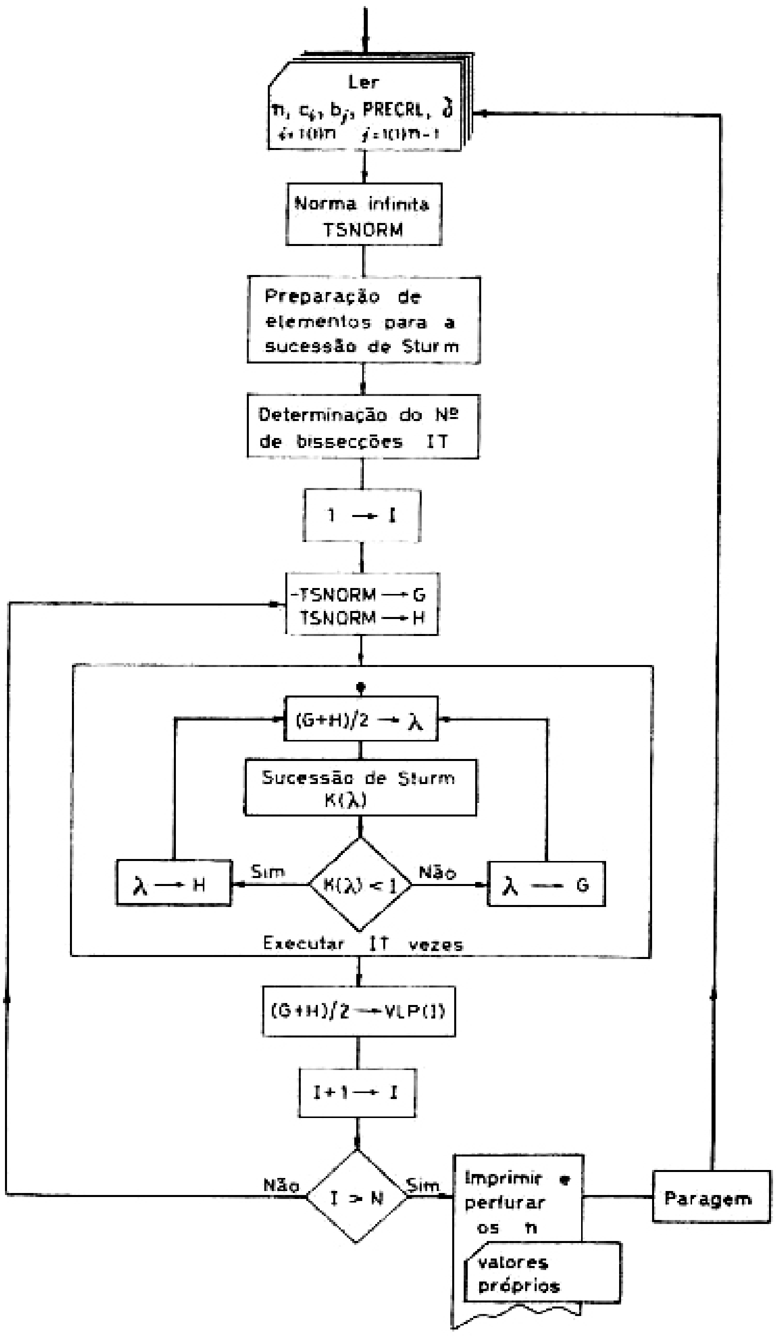
$$K\left(\frac{g+h}{2}\right)\begin{cases} \geqslant i \Rightarrow \lambda_i \in \left[\frac{g+h}{2}, h\right] & \text{mudar } g \\ < i \Rightarrow \lambda_i \in \left[g, \frac{g+h}{2}\right] & \text{mudar } h \end{cases}$$

Ao fim de m aplicações do processo de bissecção, o intervalo $[g, h]$, que contém sempre λ_i, está reduzido a uma amplitude $2 \times \text{norma}/2^m$. Obtém-se λ_i com erro inferior a δ tomando o ponto médio de $[g, h]$ ao fim de um número de iterações superior a $\log_e\left(\frac{\text{norma}}{\delta}\right)/\log_e 2+1$.

Os menores satisfazem a relação de recorrência

$$D_i(\lambda) = (c_i - \lambda) D_{i-1}(\lambda) - b_i^2 D_{i-2}(\lambda)$$

partindo de $D_0 = 1$ e $D_1(\lambda) = c_1 - \lambda$.


Ler
n, c_i, b_j, PRECRL, δ
$i=1(1)n$ $j=1(1)n-1$
Norma infinita TSNORM
Preparação de elementos para a sucessão de Sturm
Determinação do Nº de bissecções IT
1 → I
-TSNORM → G
TSNORM → H
(G+H)/2 → λ
Sucessão de Sturm K(λ)
K(λ) < 1
Sim
Não
λ → H
λ → G
Executar IT vezes
(G+H)/2 → VLP(I)
I+1 → I
I > N
Não
Sim
Imprimir e perfurar os n
valores próprios
Paragem

```
C        VALORES PROPRIOS DE MATRIZES SIMETRICAS TRIDIAGONAIS
C
         DIMENSIØN C(100),B(100),P(100),VLP(100)
       1 READ 2,N
       2 FØRMAT (I5)
         N1=N-1
         READ 3,(C(I),I=1,N),(B(I),I=1,N1),PRECRL,DELTA
       3 FØRMAT (E14.8)
C        N-ORDEM DA MATRIZ
C        C(I)-ELEMENTOS DA DIAGONAL
C        B(I)-ELEMENTOS DA SUBDIAGONAL
         B(N)=0.
C        PRECRL-PRECISAO RELATIVA DO COMPUTADOR
C        VLP(I)-VALORES PROPRIOS
C        DELTA-ERRO ABSOLUTO DOS VALORES PROPRIOS
         TSNØRM=ABSF(C(1))+ABSF(B(1))
         DØ 11 I=2,N
         R=ABSF(B(I-1))+ABSF(C(I))+ABSF(B(I))
C        SOMA DOS MODULOS DOS ELEMENTOS DE CADA LINHA
         IF(TSNØRM-R)12,11,11
      12 TSNØRM=R
      11 CØNTINUE
C        TSNORM-NORMA INFINITA DA MATRIZ
         PN2=(PRECRL*TSNØRM)**2
         P(1)=0.
C        P(I)=QUADRADO DE B(I-1) EXCEPTO SE ESTE FOR NULO
         DØ 13 I=2,N
         IF(B(I-1))15,14,15
      14 P(I)=PN2
         GØ TØ 13
      15 P(I)=B(I-1)**2
      13 CONTINUE
         IT=LØGF(TSNØRM/DELTA)/LØGF(2.)+2.
C        IT-NUMERO DE BISSECCOES NECESSARIAS PARA O RIGOR PEDIDO
         DØ 16 I=1,N
C        CALCULO DO VALOR PROPRIO NUMERO I
C        VLP(1)MAIOR QUE VLP(2)-----MAIOR QUE VLP(N)
         G=-TSNØRM
         H= TSNØRM
         DØ 117 J=1,IT
         ALAM=(G+H)*0.5
C        CONCORDANCIAS DA SUCESSAO DE STURM
         K=0
         P1=0.
         P2=1.
         DØ 111 L=1,N
         P3=(C(L)-ALAM)*P2-P(L)*P1
         P1=P2
         P2=P3
         IF (P1)112,113,113
     112 IF (P2)114,111,111
     113 IF (P2)111,114,114
     114 K=K+1
     111 CØNTINUE
```

```
C     TOMAM-SE OS TERMOS NULOS COMO POSITIVOS
C     EXCPTO PARA I=N
C     (NAO SE ANULAM CONSECUTIVAMENTE DOIS TERMOS
C     E ANTES E DEPOIS DE TERMO NULO HA SINAIS OPOSTOS)
      IF(P2)107,115,107
  115 IF(P1)107,107,116
  116 K=K-1
C     K-NUMERO DE CONCORDANCIAS DE SINAL
  107 IF(K-I)118,119,119
C     K MENOR QUE I-MENOS DE I PONTOS VLP A DIREITA DE ALAM
  118 H=ALAM
      GØ TØ 117
  119 G=ALAM
  117 CØNTINUE
   16 VLP(I)=0.5*(G+H)
      PRINT 3,(VLP(I),I=1,N)
      PUNCH 3,(VLP(I),I=1,N)
      PAUSE
      GØ TØ 1
      END
```

A matriz $\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}$ de valores próprios

$\lambda_i = 2 + 2\cos\frac{i\pi}{6}$ com $i = 1(1)5$, foi resolvida por este programa $\delta = 10^{-7}$, tendo-se obtido os seguintes resultados:

```
.37320508E+01
.29999999E+01
.19999999E+01
.99999995E+00
.26794917E+00
```

Bibliografia: [3], [10], [27], [36].

EXEMPLO 10 — SOMA DE SÉRIES DE POLINÓMIOS DE CHEBYSHEV PELO MÉTODO DE CLENSHAW.

Os polinómios de Chebyshev de que nos vamos ocupar (existem outros) *definem-se por*

$$T_k(x)=cos\ (k\ arc\ cos\ x) \qquad x\in[-1,\ +1]$$

Pode verificar-se que é válida a relação

$$T_{r+1}(x)-2xT_r(x)+T_{r-1}(x)=0$$

Esta relação permite gerar os $T_k(x)$ por recorrência, começando com $T_0(x)=1$, $T_1(x)=x$, e mostra ao mesmo tempo, que a função $T_k(x)$ é um polinómio de grau k.

Prova-se que uma função $f(x)$, contínua e de variação limitada no intervalo $[-1, +1]$, admite um desenvolvimento convergente

$$f(x)=\frac{a_0}{2}+\sum_{r=1}^{\infty} a_r T_r(x) \qquad x\in[-1,\ +1]$$

A acentuada convergência destas séries tem-nas imposto cada vez mais em Análise Numérica.

Conhecem-se [6] os coeficientes a_i do desenvolvimento de algumas funções de uso corrente, para o intervalo referido.

Para somar a série até ao termo $a_n T_n$, Clenshaw [5] propôs o uso das relações

$$b_r=2xb_{r+1}-b_{r+2}+a_r \qquad \text{com } b_{n+1}=b_{n+2}=0$$

vindo

$$f(x)\doteq\frac{1}{2}(b_0-b_2)$$

Também é frequente a utilização dos polinómios $T_k^*(x)$, adaptados ao intervalo $[0,1]$, definidos por $T_k^*(x) = cos\,[k\ arc\ cos\,(2x-1)] = T_k(2x-1)$. Podem gerar-se pela mesma relação de recorrência dos T_k, partindo de

$$T_0^*(x) = 1,\ T_1^*(x) = 2x-1 .$$

Um desenvolvimento em série de polinómios T_k^* pode ainda somar-se pelo processo de Clenshaw, substituindo apenas x por $2x-1$.

Como em Fortran não é admissível o índice 0, o programa apresenta os índices avançados de uma unidade. Ao calcular os bb, só três deles precisam de ser registados em cada passo; designam-se os seus localizadores por $B0$, $B1$, $B2$.

```
      FUNCTIØN SCHCBY(N,A,X)
      DIMENSIØN A(65)
      B1=0.
      B0=0.
      DØ1I=1,N
      B2=B1
      B1=B0
      J=N-I+1
    1 B0=2.*X*B1-B2+A(J)
      SCHEBY=0.5*(B0-B2)
      RETURN
      END
```

Bibliografia: [5], [6], [27].

EXEMPLO 11 — AJUSTAMENTO DE CURVAS POR SÉRIES DE POLINÓMIOS ORTOGONAIS. CRITÉRIO DOS MÍNIMOS QUADRADOS.

Na programação deste problema seguimos de perto as orientações de Forsythe [12] e Clenshaw [7] das quais passamos a enunciar alguns tópicos.

Suponhamos que se pretende ajustar um polinómio de grau m

$$y_m(x)=a_0+a_1x+a_2x^2+\ldots+a_mx^m \tag{1}$$

a um conjunto de pontos $P_l(X_l, Y_l)$, $l=1, 2, \ldots, n$ de forma a minimizar

$$\sum_{l=1}^{n}\left[y_m(X_l)-Y_l\right]^2 \tag{2}$$

Forsythe prova que existe um e um só polinómio nestas condições se $m<n$. No entanto, o sistema de equações normais

$$\frac{\partial\Sigma\left[y_m(X_l)-Y_l\right]^2}{\partial a_i}=0 \qquad i=0, 1, \ldots, m \tag{3}$$

que determina os a_i, é muitas vezes numèricamente instável, sobretudo para $m\geqslant 7$. Este obstáculo é removido se substituirmos o desenvolvimento em potências de x por um desenvolvimento em polinómios $p_i(x)$ da forma

$$y_m(x)=c_0p_0(x)+c_1p_1(x)+\ldots+c_mp_m(x) \tag{4}$$

em que cada p_i é de grau i e satisfaz a relação de ortogonalidade

$$\sum_{l=1}^{n}p_i(X_l)p_j(X_l)=0, \qquad j\neq i. \tag{5}$$

Esta relação é insuficiente para definir uma família de polinómios p_i, mas ficarão determinados se escolhermos p_0 e calcularmos os restantes pelas relações

$$
(6) \qquad \begin{aligned} p_1 &= 2(x-\alpha_1)p_0 \\ p_i &= 2(x-\alpha_i)p_{i-1} - \beta_i p_{i-2} \qquad i = 2, 3, \ldots, m \end{aligned}
$$

Fazendo, por exemplo, $p_0 = \dfrac{1}{2}$, obtém-se de (5) e (6),

$$
(7) \qquad \alpha_i = \frac{\sum_{l=1}^{n} X_l p_{i-1}^2(X_l)}{\sum_{l=1}^{n} p_{i-1}^2(X_l)} \qquad i = 1, 2, \ldots, m
$$

e

$$
(8) \qquad \beta_i = \frac{\sum_l p_{i-1}^2(X_l)}{\sum_l p_{i-2}^2(X_l)} \qquad i = 2, 3, \ldots, m
$$

As equações normais

$$
(9) \qquad \frac{\partial \Sigma (y_m(X_l) - Y_l)^2}{\partial c_i} = 0 \qquad i = 0, 1, \ldots, m
$$

com y_m na forma (4), reduzem-se agora à forma diagonal, obtendo-se

$$
(10) \qquad c_i = \frac{\sum_l Y_l p_i(X_l)}{\sum_l p_i^2(X_l)}
$$

A função $y_m(x)$ pode obter-se por sucessivas aproximações da forma

$$(11)\qquad \begin{cases} y_0(x)=c_0 p_0(x)=\dfrac{1}{n}\sum\limits_t Y_t \\ \\ y_i(x)=y_{i-1}(x)+c_i p_i(x) \qquad\qquad i=1,2,\ldots,m \end{cases}$$

Parte-se de p_0, y_0, α_1 (e $\beta_1=0$) e calcula-se p_1, c_1, y_1, α_2 e β_2.

Passa-se depois a $p_2, c_2, y_2, \alpha_3, \beta_3$ e assim por diante até p_m, c_m, y_m.

Para representar os polinómios p_i e y_i no interior da máquina, Clenshaw sugere a memorização dos coeficientes dos seus desenvolvimentos em séries de polinómios de Chebyshev definidos para o intervalo $[-1, +1]$.

Vamos então supor que todo o problema do ajustamento se processa com $x\in[-1, +1]$, o que não traz perda de generalidade, pois é sempre possível transformar o intervalo (finito) a que pertencem as abcissas X_j no intervalo $[-1, 1]$ por mudança de variável.

Sejam

$$(12)\qquad p_i(x)=\frac{1}{2}P_0^{(i)}T_0(x)+P_1^{(i)}T_1(x)+\ldots+P_{i-1}^{(i)}T_{i-1}(x)+T_i(x)$$

$$(13)\qquad y_i(x)=\frac{1}{2}A_0^{(i)}T_0(x)+A_1^{(i)}T_1(x)+\ldots+A_{i-1}^{(i)}T_{i-1}(x)+A_i^{(i)}T_i(x)$$

os desenvolvimentos de p_i e y_i (*).

As relações (6) e (11) tomam agora a forma

$$(14)\qquad \begin{cases} P_0^{(i)}=2P_1^{(i-1)}-2\alpha_i P_0^{(i-1)}-\beta_i P_0^{(i-2)} \\ \\ P_j^{(i)}=P_{j-1}^{(i-1)}+P_{j+1}^{(i-1)}-2\alpha_i P_j^{(i-1)}-\beta_i P_j^{(i-2)} & \begin{array}{l} i=1,2,\ldots,m \\ j=1,2,\ldots,i-1 \end{array} \\ \\ P_i^{(i)}=1 \end{cases}$$

que se deduzem igualando os coeficientes de T_j depois de substituir em (6) cada p_j pelo seu desenvolvimento (atender a que $2xT_j=T_{j-1}+T_{j+1}$); e

$$(15)\qquad A_j^{(i)}=A_j^{(i-1)}+c_i P_j^{(i)} \qquad\qquad j=0,1,\ldots,i$$

É este processo de representação de p_i e y_i que usamos no programa.

(*) Em (12) fez-se $P_i^{(i)}=1$ atendendo a (6).

A resolução do problema é precedida de transformação de coordenadas

$$x \rightarrow \frac{2(x-a)}{b-a} - 1$$

em que $a = min\{X_i\}$ e $b = max\{X_i\}$.

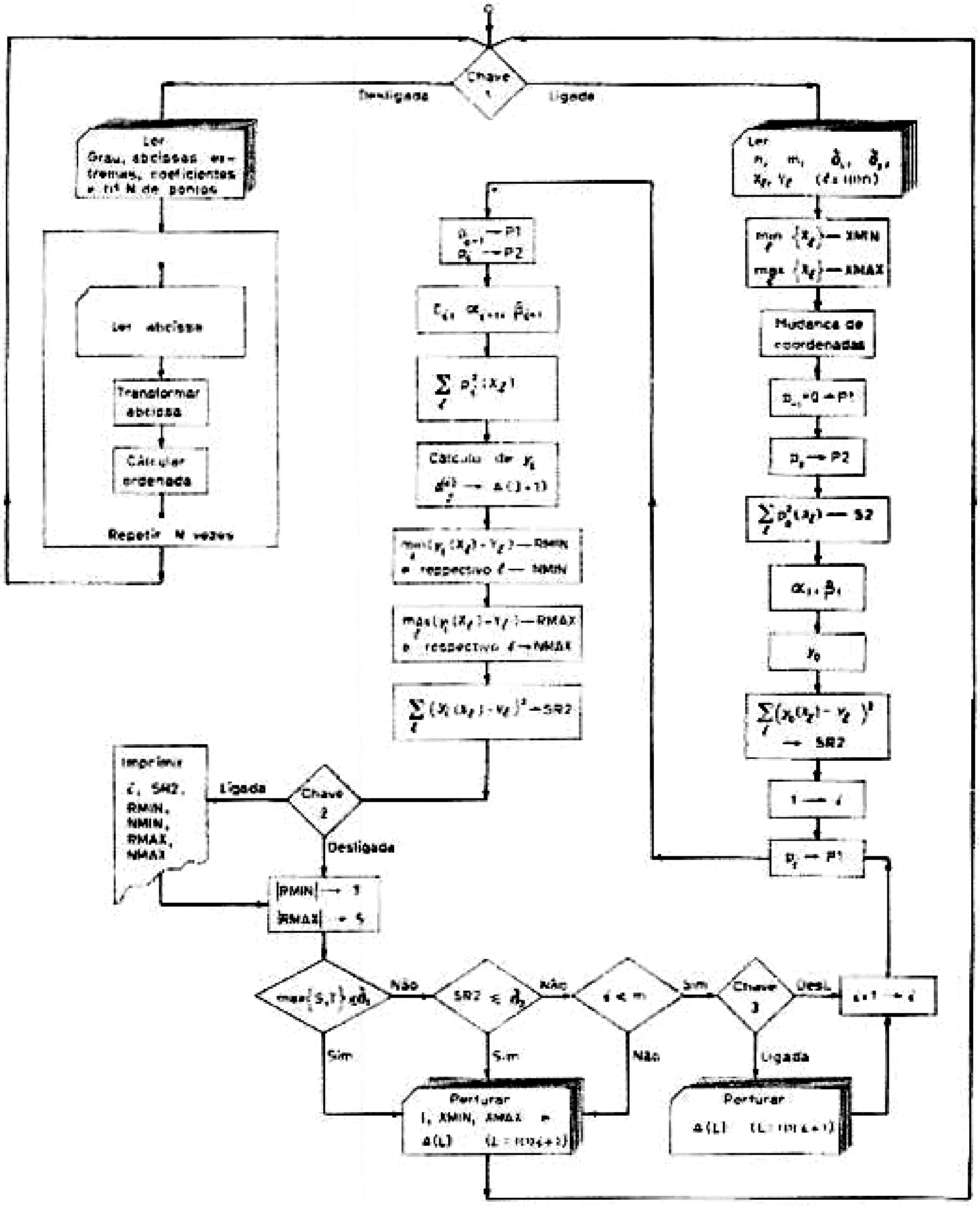
Os valores dos desenvolvimentos (12) e (13) em cada ponto são calculados com o subprograma *FUNCTION SCHEBY* do exemplo 10.

O problema que se põe na prática é o da escolha do grau m. É que, de modo geral, não interessa apenas que a curva passe o mais perto possível dos pontos mas que tenha também uma forma suave, sem oscilações, condição intimamente ligada ao grau da curva. A solução terá então de representar um equilíbrio entre estas duas condições.

No programa considera-se atingida a aproximação final quando

$\max_i |y_m(X_i) - Y_i| \leqslant \delta_1$ ou $\sum_i [y_m(X_i) - Y_i]^2 \leqslant \delta_2$, ou ainda quando $m = M$ sendo M, δ_1 e δ_2 constantes definidas no início do programa.

Determinada a curva solução, o programa possibilita também o cálculo de pontos (X, Y) desta curva, com $a \leqslant X \leqslant b$ (Desligar chave 1).


Chave 1
Desligada
Ligada
Ler
Grau, abcissas extremas, coeficientes e nº N de pontos
Ler abcissa
Transformar abcissa
Calcular ordenada
Repetir N vezes
Mudança de coordenadas
Cálculo de
Imprimir
SR2,
RMIN,
NMIN,
RMAX,
NMAX
Ligada
Chave 2
Desligada
Não
Sim
Chave 3
Desl.
Ligada
Perfurar
Perfurar
XMIN, XMAX e

```
C     AJUSTAMENTO DE CURVAS POR SERIES DE POLINOMIOS ORTOGONAIS
C                      CRITERIO DOS MINIMOS QUADRADOS
C
C     ASSOCIE-SE A ESTE PROGRAMA O SUBPROGRAMA  FUNCTION SCHEBY
C
      DIMENSIØN A(25),P1(25),P2(25),X(100),Y(100)
   36 PRINT 1
    1 FØRMAT(//52HCHAVE 1 LIGADA   -DETERMINACAØ DA CURVA APRØXIMADØRA/
     1               8X,44HDESLIGADA-       ,,        DE PØNTØS DESTA CURVA)
      PAUSE
      IF(SENSE SWITCH 1)2,3
C
    2 PRINT 4
    4 FØRMAT(/50HCHAVE 2 LIGADA-IMPRESSAØ DØ GRAU,RESIDUØS EXTREMØS
     2 11H(RMIN,RMAX)/15X,42HE SØMA DØS QUADRADØS DØS RESIDUØS(SR2) DAS/
     3 15X,24HAPRØXIMACØES INTERMEDIAS)
      PRINT 5
    5 FØRMAT(/54HCHAVE 3 LIGADA-PERFURACAØ DE CØEF. DE APRØX. INTERMEDIAS)
      PAUSE
C
C     N-NUMERO DE PONTOS, M-GRAU MAXIMO
      READ 6,N,M
C     DELTA1-LIMITE SUPERIOR DO VALOR ABSOLUTO DOS RESIDUOS
C     DELTA2-    ,,        ,,     DA SOMA DOS QUADRADOS DOS RESIDUOS
      READ 7,DELTA1,DELTA2
C     LEITURA DAS COORDENADAS DOS PONTOS
      READ 7,(X(L),Y(L),L=1,N)
C     PROCURA DAS ABCISSAS EXTREMAS
C     XMIN-ABCISSA MINIMA, XMAX-ABCISSA MAXIMA
      XMAX=X(1)
      XMIN=X(1)
      DØ 8 L=2,N
      S=X(L)
      IF(XMAX-S)9,8,10
    9 XMAX=S
      GØ TØ 8
   10 IF(XMIN-S)8,8,11
   11 XMIN=S
    8 CØNTINUE
C     MUDANCA DE COORDENADAS
      S=2./(XMAX-XMIN)
      DØ 12 L=1,N
   12 X(L)=S*(X(L)-XMIN)-1.
C
C     CONVENCAO- P1 LOCALIZA INICIALMENTE UM POLINOMIO IDENT. NULO
      P1(1)=0.
      P1(2)=0.
C     POLINOMIO ORTOGONAL DE GRAU 0.
      P2(1)=1.
      P2(2)=0.
      P2(3)=0.
```

```
C
      FN=N
      S2=.25*FN
C     DEFINICAO DE ALFA(1) E BETA(1)
      ALFA=0.
      DØ 13 L=1,N
   13 ALFA=ALFA+X(L)
      ALFA=ALFA/FN
      BETA=0.
C     APROXIMACAO DE GRAU 0
      S=0.
      DØ 14 L=1,N
   14 S=S+Y(L)
      S=S/FN
      A(1)=S+S
      A(2)=0.
C
      I=1
C     CALCULO DE NOVO POLINOMIO ORTOGONAL
   35 P1(1)=2.*(P2(2)-ALFA*P2(1))-BETA*P1(1)
      DØ 15 L=2,I
   15 P1(L)=P2(L+1)+P2(L-1)-2.*ALFA*P2(L)-BETA*P1(L)
      I1=I+1
      P1(I1)=1.
C     PERMUTA DOS VECTORES P1 E P2
      DØ 16 L=1,I1
      S=P1(L)
      P1(L)=P2(L)
   16 P2(L)=S
      P2(I+2)=0.
C
C     CALCULO DE C(I),ALFA(I+1),BETA(I+1)
      C=0.
      ALFA=0.
      BETA=0.
      T=0.
      DØ 17 L=1,N
      S=SCHEBY(I1,P2,X(L))
      C=C+Y(L)*S
      T=T+S*S
   17 ALFA=ALFA+X(L)*S*S
      C=C/T
      ALFA=ALFA/T
      BETA=T/S2
C
      S2=T
C     APROXIMACAO DE GRAU I
      DØ 18 L=1,I1
   18 A(L)=A(L)+C*P2(L)
      A(I+2)=0.
```

```
C     RESIDUOS EXTREMOS E SOMA DOS QUADRADOS DOS RESIDUOS
C     RMIN-RESIDUO MINIMO, RMAX-RESIDUO MAXIMO
C     SR2-SOMA DOS QUADRADOS DOS RESIDUOS
      RMAX=SCHEBY(I1,A,X(1))-Y(1)
      RMIN=RMAX
      NMAX=1
      NMIN=1
      SR2=RMAX*RMAX
      DØ 19 L=2,N
      S=SCHEBY(I1,A,X(L))-Y(L)
      IF(S-RMIN)20,19,21
   20 RMIN=S
      NMIN=L
      GØ TØ 19
   21 IF(S-RMAX)19,19,22
   22 RMAX=S
      NMAX=L
   19 SR2=SR2+S*S
C
      IF(SENSE SWITCH2)23,24
C     IMPRESSAO DO GRAU,SOMA DOS QUAD.DOS RESIDUOS,RESID.EXTREMOS
   23 PRINT 25,I,SR2,RMIN,NMIN,RMAX,NMAX
   25 FØRMAT(/5HGRAU I3/27HSØMA DØS QUAD.DØS RESIDUØS E14.8/
     1 19HRESIDUØS EXTREMØS  E14.8,6H(PØNTØI3,2H),E14.8,6H(PØNTØI3,1H))
C     TESTES DE FINALIZACAO
   24 S=ABSF(RMAX)
      T=ABSF(RMIN)
      IF(S-T)26,27,27
   26 S=T
   27 IF(S-DELTA1)28,28,29
   29 IF(SR2-DELTA2)28,28,30
   30 IF(I-M)31,28,28
C
   31 IF(SENSE SWITCH3)32,33
   32 PUNCH34,(A(L),L=1,I1)
   33 I=I+1
      GØ TØ 35
C     PERFURACAO DO GRAU,ABCISSAS EXTREMAS E COEFICIENTES
   28 PUNCH6,I
      PUNCH7,XMIN,XMAX
      PUNCH34,(A(L),L=1,I1)
      PAUSE
      GØ TØ 36
```

```
C
C     DETERMINACAO DE PONTOS DA CURVA APROXIMADORA
C
C     LEITURA DO GRAU
    3 READ 6,I
C     LEITURA DOS EXTREMOS DO INTERVALO DE AJUSTAMENTO
      READ 7,XMIN,XMAX
      II=I+1
C     LEITURA DOS COEFICIENTES
      READ 34,(A(L),L=1,II)
C     LEITURA DO NUMERO DE ORDENADAS A DETERMINAR
      READ 6,N
      PRINT 37
   37 FØRMAT(/6X,1HX,14X,1HY)
C     AS COORDENADAS DOS PONTOS(NO QUE SEGUE)SAO S,T.
      C=2./(XMAX-XMIN)
      DØ 38 L=1,N
      READ 34,S
      S2=C*(S-XMIN)-1.
      T=SCHEBY(II,A,S2)
   38 PRINT 7,S,T
      PAUSE
      GØ TØ 36
    6 FØRMAT(2I4)
    7 FØRMAT(E14.8,1X,E14.8)
   34 FØRMAT(E14.8)
      END
```

Apresentamos a seguir dois exemplos de aplicação (fez-se $f=12$).

1. $\{X_i, Y_i\}=\{(-2,-5), (-0.5, 0.625), (0,1), (1,4), (1.5, 8.125), (2,15)\}$
   $\delta_1=5\times10^{-8}$, $\delta_2=0$, $M=5$.

Executando o programa com as chaves 1 e 2 ligadas e 3 desligada obtém-se

```
GRAU    1
SOMA DOS QUAD.DOS RESIDUOS   .27087980E+02
RESIDUOS EXTREMOS  -.36634615E+01(PONTO  6), .29096153E+01(PONTO  4)

GRAU    2
SOMA DOS QUAD.DOS RESIDUOS   .12949416E+02
RESIDUOS EXTREMOS  -.18891382E+01(PONTO  2), .20256732E+01(PONTO  4)

GRAU    3
SOMA DOS QUAD.DOS RESIDUOS   .41420000E-18
RESIDUOS EXTREMOS  -.25000000E-09(PONTO  1), .50000000E-09(PONTO  6)
```

Como a aproximação de grau 3 satisfaz as exigências de precisão, são perfurados em cartões o grau, os extremos do intervalo de ajustamento e os coeficientes desta aproximação, obtendo-se, para os últimos, os valores 6, 8, 2 e 2.

A curva solução é, portanto,

$$\frac{1}{2}6T_0+8T_1\left(\frac{x}{2}\right)+2T_2\left(\frac{x}{2}\right)+2T_3\left(\frac{x}{2}\right)=1+x+x^2+x^3$$

Note-se que por o intervalo de ajustamento ser [—2, +2] se substitui, no programa, x por $\frac{x}{2}$.

2. O segundo conjunto $\{X_i, Y_i\}$ é formado pelos pontos da curva $y=sen\,x$, correspondentes às abcissas — 40° (10°) 40°.

Os valores X_i (em radianos) e Y_i foram dados com 7 decimais.

Ao longo do cálculo obtivemos

```
GRAU     1
SOMA DOS QUAD.DOS RESIDUOS  .10512263E-02
RESIDUOS EXTREMOS  -.14390888E-01(PONTO  1), .14390888E-01(PONTO  9)

GRAU     2
SOMA DOS QUAD.DOS RESIDUOS  .10512263E-02
RESIDUOS EXTREMOS  -.14390888E-01(PONTO  1), .14390888E-01(PONTO  9)

GRAU     3
SOMA DOS QUAD.DOS RESIDUOS  .36442049E-07
RESIDUOS EXTREMOS  -.96973280E-04(PONTO  2), .96973285E-04(PONTO  8)

GRAU     4
SOMA DOS QUAD.DOS RESIDUOS  .36442049E-07
RESIDUOS EXTREMOS  -.96973275E-04(PONTO  2), .96973290E-04(PONTO  8)

GRAU     5
SOMA DOS QUAD.DOS RESIDUOS  .20756657E-12
RESIDUOS EXTREMOS  -.21775400E-06(PONTO  6), .21774600E-06(PONTO  7)
```

e, perfurados em cartões, os valores

```
 .14650020E-10
 .65645387E-00
 .60582273E-11
-.13750975E-01
-.66106801E-11
 .84712709E-04
```

para coeficientes da curva de grau 5.

Ao reiniciar o programa, desligou-se a chave 1 e fez-se a determinação dos pontos de abcissas −35°, −14°, −6°, 1°, 3°, 7°, 16°, 28°, 34°, obtendo-se os seguintes resultados:

X	Y
−.61086520E−00	−.57357589E−00
−.24434610E−00	−.24192181E−00
−.10471980E−00	−.10452834E−00
.17453300E−01	.17452380E−01
.52359900E−01	.52335883E−01
.12217300E−00	.12186912E−00
.27925270E−00	.27563736E−00
.48869220E−00	.46947159E−00
.59341190E−00	.55919241E−00

Bibliografia: [1], [2], [7], [12], [24], [26].

EXEMPLO 12 — A FUNÇÃO *GAMA*.

Em [6] estão publicados os coeficientes a_r, com 20 decimais, do desenvolvimento

$$\Gamma(1+x)=\frac{a_0}{2}+\sum_{r=1} a_r T_r^*(x) \qquad \text{para } x\in[0,1]$$

Para $x\notin[0,1]$, o cálculo é facilitado pela importante igualdade

$$\Gamma(x)=(z-1)\Gamma(z-1)$$

Assim

$$\Gamma(3.2)=2.2\Gamma(2.2)=2.2\times 1.2\Gamma(1+0.2)$$

$$\Gamma(-2.6)=\Gamma(-1.6)/(-2.6)=\Gamma(1+0.4)/\left[(-2.6)(-1.6)(-0.6)(0.4)\right].$$

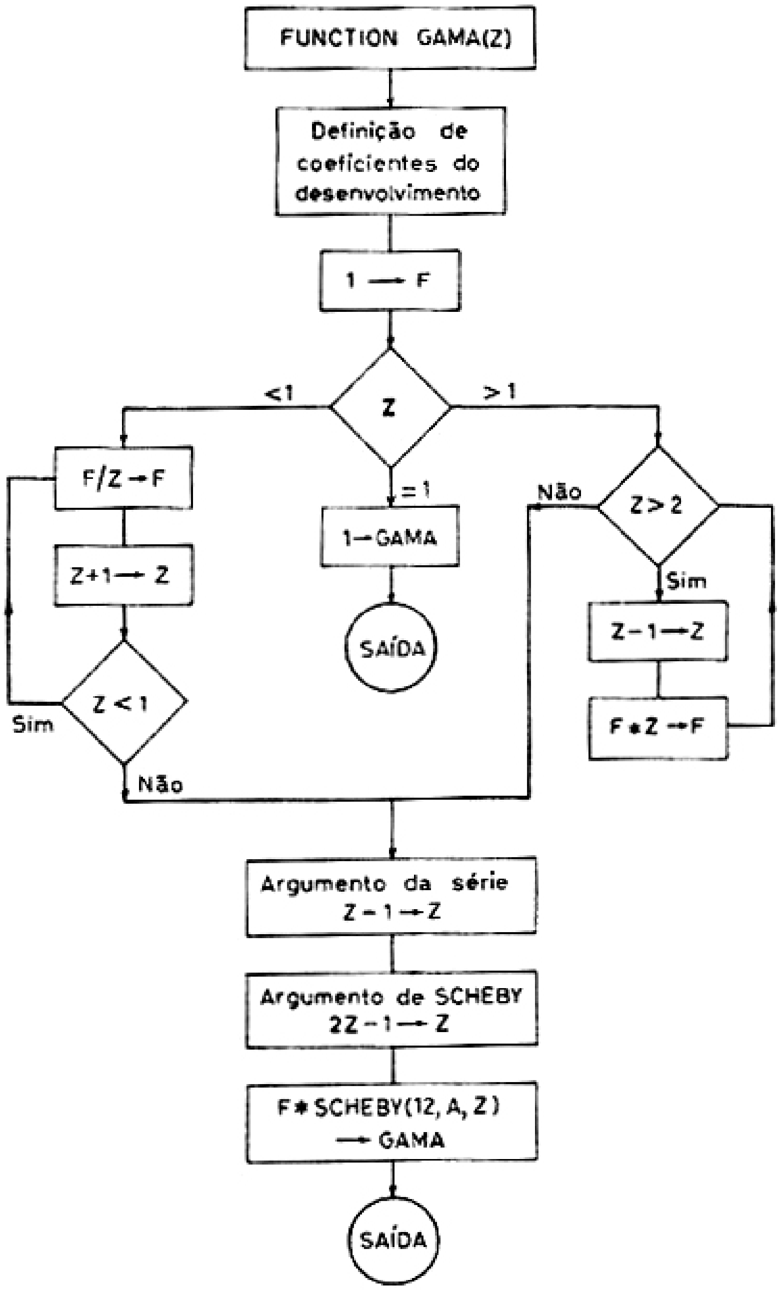
Pondo

$$\Gamma(z)=f\cdot\Gamma(1+\theta) \qquad \text{com } \theta\in[0,1]$$

e calculando f como sugerem os exemplos, resta-nos somar a série relativa a $\Gamma(1+\theta)$, truncando-a em altura conveniente para a precisão do resultado (*). Dispomos já de um subprograma para somar uma série de polinómios de Chebyshev no intervalo $[-1, +1]$. Como estamos a trabalhar em $[0,1]$, impõe-se a substituição de x por $2x-1$ antes de utilizar *SCHEBY*.

Convém apontar que os inteiros negativos e zero são polos de $\Gamma(z)$. No programa utilizámos 12 termos da série; $|a_{12}|$ é inferior a $0{,}5\times10^{-8}$.

(*) Ao leitor familiarizado com a linguagem *ALGOL*, recomenda-se a leitura de [8], onde se prefere o desenvolvimento de $1/\Gamma(x)$, que converge mais ràpidamente, calculando-se depois o inverso do valor obtido.


FUNCTION GAMA(Z)
Definição de coeficientes do desenvolvimento
1 → F
Z
<1
>1
=1
F/Z → F
Z+1 → Z
Z < 1
Sim
Não
1 → GAMA
SAÍDA
Z > 2
Não
Sim
Z − 1 → Z
F * Z → F
Argumento da série
Z − 1 → Z
Argumento de SCHEBY
2Z − 1 → Z
F * SCHEBY(12, A, Z)
→ GAMA
SAÍDA

```
      FUNCTIØN GAMA(Z)
      DIMENSIØN A(12)
      A( 1)= 1.8835712
      A( 2)= .44153813 E-02
      A( 3)= .56850437 E-01
      A( 4)=-.42198354 E-02
      A( 5)= .13268082 E-02
      A( 6)=-.18930245 E-03
      A( 7)= .36069253 E-04
      A( 8)=-.60567619 E-05
      A( 9)= .10558295 E-05
      A(10)=-.18119674 E-06
      A(11)= .31177250 E-07
      A(12)=-.53542196 E-08
C
      F=1.
      IF(Z-1.)2,1,3
    1 GAMA=1.
      RETURN
    2 F=F/Z
      Z=Z+1.
      IF(Z-1.)2,4,4
C
    3 IF(Z-2.)4,4,5
C
    5 Z=Z-1.
      F=F*Z
      GØ TØ 3
C
C     ARGUMENTO DA SERIE
    4 Z=Z-1.
C     PREPARAR O ARGUMENTO DE SCHEBY
      Z=2.*Z-1.
      GAMA=F*SCHEBY(12,A,Z)
      RETURN
      END
```

Utilizando o programa principal

```
  111 ACCEPT11,Z
   11 FØRMAT(E14.8)
      Z=GAMA(Z)
      PRINT11,Z
      GØ TØ 111
      END
```

obtivemos

```
-.15000000E+01RS
 .23632719E+01
+.15000000E+01RS
 .88622700E+00
+.17000000E+01RS
 .90863875E+00
+.50000000E+01RS
 .24000000E+02
```

onde a cada valor lido por máquina de escrever se segue imediatamente o correspondente valor da função Gama.

Bibliografia: [6], [8].

EXEMPLO 13 — RAÍZES DE POLINÓMIOS. MÉTODO DE BAIRSTOW.

Seja $f(z) = z^n + \sum_{i=1}^{n} a_i z^{n-i}$ um polinómio de grau n com coeficientes a_i reais.

No método de Bairstow procura-se atingir um factor quadrático $z^2 - s^* z + p^*$, de $f(z)$, partindo de valores aproximados de s^* e p^*. Uma vez conseguido o objectivo, dentro de certo rigor, o factor quadrático fornece um par de raízes (reais ou complexas) e o polinómio cociente da divisão de $f(z)$ por aquele factor conserva os restantes zeros, podendo aplicar-se-lhe técnica idêntica. Obtêm-se, assim, sucessivamente todas as raízes de $f(z)$.

São os seguintes os passos do método, que justificaremos mais adiante.

Divide-se $f(z)$ por um factor aproximado $m(z) = z^2 - sz + p$

$$f(z) = z^n + \sum_{i=1}^{n} a_i z^{n-i} = m(z)g(z) + r(z) \tag{1}$$

obtendo-se um resto $r(z) = r_1 z + r_2$ e um cociente $g(z)$ que voltaremos a dividir por $m(z)$

$$g(z) = z^{n-2} + \sum_{i=1}^{n-2} g_i z^{n-2-i} = m(z)h(z) + t(z) \tag{2}$$

com

$$h(z) = z^{n-4} + \sum_{i=1}^{n-4} h_i z^{n-4-i} \qquad t(z) = t_1 z + t_2$$

Temos

$$
\begin{array}{ll}
g_1=a_1+s & h_1=g_1+s \\
g_2=a_2+sg_1-p & h_2=g_2+sh_1-p \\
g_3=a_3+sg_2-pg_1 & h_3=g_3+sh_2-ph_1 \\
g_4=a_4+sg_3-pg_2 & h_4=g_4+sh_3-ph_2 \\
\ldots & \ldots \\
g_{n-2}=a_{n-2}+sg_{n-3}-pg_{n-4} & h_{n-4}=g_{n-4}+sh_{n-5}-ph_{n-6}
\end{array}
$$

Podemos calcular os g_k (e anàlogamente os h_k) pelas fórmulas de recorrência

$$g_k=a_k+sg_{k-1}-pg_{k-2} \qquad k=1, 2, \ldots, n-2$$

impondo

$$g_0=1 \qquad g_{-1}=0$$

Avançando mais dois termos nas relações de recorrência, obtemos

$$
\begin{array}{ll}
g_{n-1}=a_{n-1}+sg_{n-2}-pg_{n-3}=r_1 & h_{n-3}=g_{n-3}+sh_{n-4}-ph_{n-5}=t_1 \\
g_n=a_n+sg_{n-1}-pg_{n-2}=r_2+sg_{n-1} & h_{n-2}=g_{n-2}+sh_{n-3}-ph_{n-4}=t_2+sh_{n-3}
\end{array}
$$

donde

$$r(z)=g_{n-1}(z-s)+g_n \qquad t(z)=h_{n-3}(z-s)+h_{n-2}$$

Fazendo agora

$$C=sh_{n-2}-ph_{n-3} \quad \text{e} \quad D=h_{n-2}^2-Ch_{n-3}$$

obtêm-se *correcções* Δs e Δp para os valores s e p, pelas fórmulas

$$
\begin{aligned}
\Delta s&=(h_{n-3}g_n-h_{n-2}g_{n-1})/D \\
\Delta p&=(h_{n-2}g_n-Cg_{n-1})/D
\end{aligned}
$$

Repete-se o processo enquanto o valor relativo ou o valor absoluto das correcções seja superior a um valor δ prefixado.

Justificação do método:

Retomemos as expressões (1) e (2), donde

$$f(z)=(z^2-sz+p)^2h(z)+(z^2-sz+p)t(z)+r(z).$$

Sejam Δs e Δp as correcções que fazem $s+\Delta s=s^*$ e $p+\Delta p=p^*$ e sejam z_1 e z_2 as raízes de $z^2-s^*z+p^*$, o que permite escrever

$$z_i^2-sz_i+p=\Delta s\, z_i-\Delta p \qquad (z_i=z_1 \text{ ou } z_2)$$

Como

$$f(z_i)=0=(z_i^2-sz_i+p)^2h(z_i)+(z_i^2-sz_i+p)t(z_i)+r(z_i)$$

vem

$$(\Delta s\, z_i-\Delta p)^2h(z_i)+(\Delta s\, z_i-\Delta p)\left[h_{n-3}(z_i-s)+h_{n-2}\right]+g_{n-1}(z_i-s)+g_n=0$$

Desprezando termos de 2.ª ordem em Δs e Δp (cai imediatamente a primeira parcela) somos conduzidos a

$$(z_i-s)\left[h_{n-2}\Delta s-h_{n-3}\Delta p+g_{n-1}\right]+\left[(sh_{n-2}-ph_{n-3})\Delta s-h_{n-2}\Delta p+g_n\right]=0$$

Este polinómio do 1.º grau em z_i anula-se em dois pontos (se $z_1\neq z_2$) o que arrasta

$$h_{n-2}\Delta s-h_{n-3}\Delta p+g_{n-1}=0$$
$$(sh_{n-2}-ph_{n-3})\Delta s-h_{n-2}\Delta p+g_n=0$$

sistema que resolvido em ordem a Δs e Δp nos dá as expressões anteriormente apresentadas.

No programa que preparámos, começa-se pela leitura dum valor para δ, após o que são lidos o grau e coeficientes do polinómio.

Se os últimos m coeficientes são nulos há m raízes nulas e baixa-se o grau de n para $n-m$.

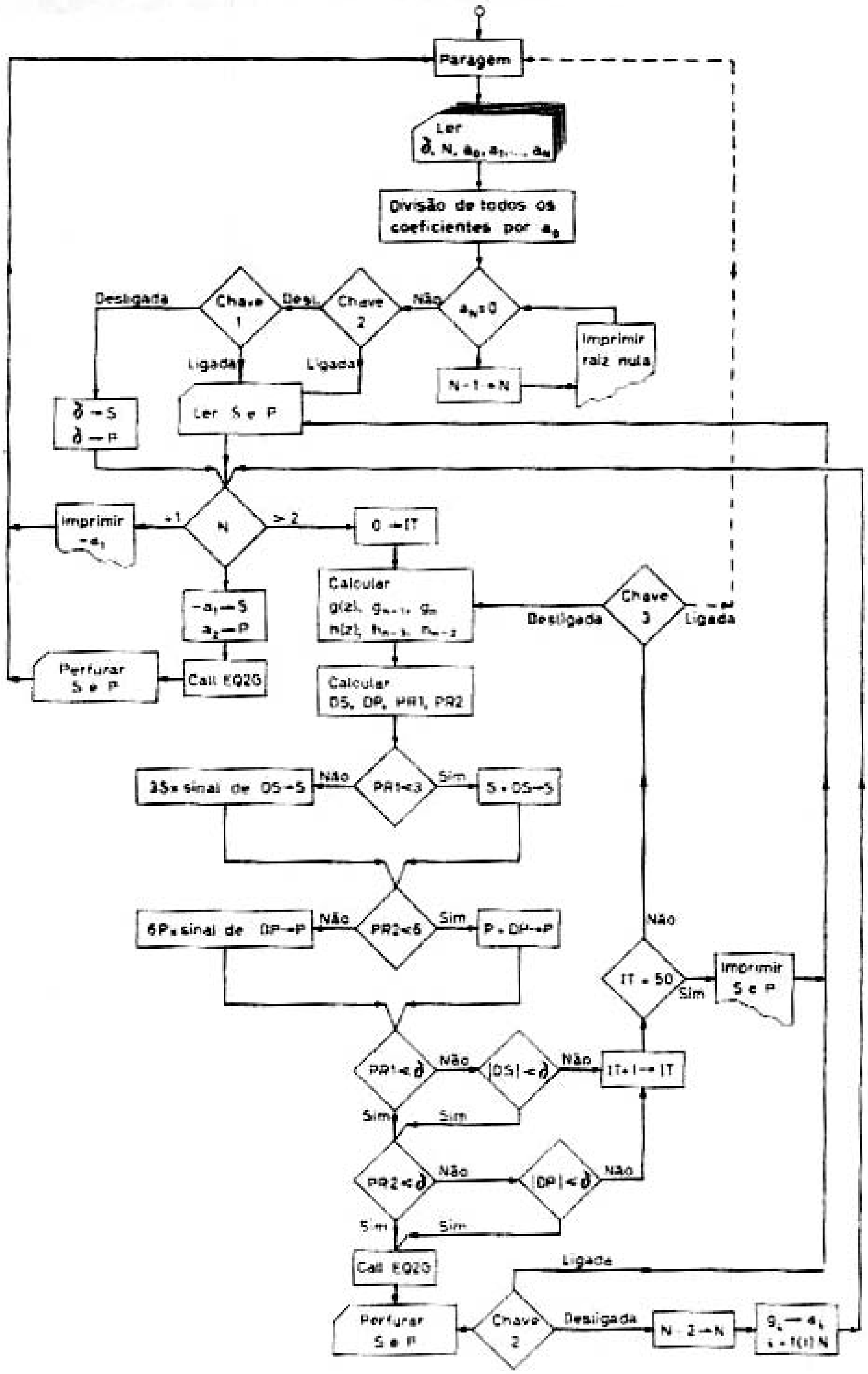
Numa 1.ª fase (com chave 2 desligada) o programa tenta obter todos os factores quadráticos como indicámos no 2.º parágrafo.

Numa 2.ª fase (chave 2 ligada) voltam a ser lidos δ, n e os coeficientes do polinómio original, com o fim de melhorar os factores quadráticos obtidos na 1.ª fase, fazendo a iteração ùnicamente com $f(z)$, e não com os polinómios cocientes, para evitar perdas de precisão.

Se se efectuarem 50 iterações sem se verificar convergência (para certos valores iniciais de s e p) o programa pede novas sugestões para s e p. O programa pode ser reiniciado em qualquer altura ligando a chave 3.

Para evitar saltos bruscos nas correcções, não se permite que os valores de $|\Delta s|$ e $|\Delta p|$ tenham variações superiores a $|3s|$ e $|6p|$.

Para $n=3$, h_{n-3} teria índice nulo; introduzimos por isso a variável $HNM3$ que toma o valor de h_{n-3} com $n>3$ e o valor 1 se $n=3$.


Paragem
Ler
∂, N, a0, a1, ..., aN
Divisão de todos os coeficientes por a0
aN = 0
Não
Imprimir raiz nula
N - 1 → N
Chave 2
Desl.
Chave 1
Desligada
Ligada
Ligada
Ler S e P
∂ → S
∂ → P
N
+1
> 2
Imprimir -a1
-a1 → S
a2 → P
Call EQ2G
Perfurar S e P
0 → IT
Calcular
DS, DP, PR1, PR2
Chave 3
Desligada
Ligada
PR1<3
Não
Sim
S + DS → S
PR2<6
6P x sinal de DP → P
P + DP → P
IT = 50
Imprimir S e P
PR1<∂
|DS|<∂
IT + 1 → IT
PR2<∂
|DP|<∂
Call EQ2G
Perfurar S e P
Chave 2
Ligada
Desligada
N - 2 → N

```
C     RAIZES DE POLINOMIOS - METODO DE BAIRSTOW
C     P(Z)=A(0)*Z**N+A(1)*Z**(N-1)+....+A(N-1)*Z+A(N)
C
C     CHAVE 2 DESLIGADA-DETERMINAR TODOS OS FACTORES
C                           QUADRATICOS  Z**2-S*Z+P E SUAS RAIZES
C                          (CHAVE 1 LIGADA-LER SUGESTOES PARA S E P )
C                          (          DESLIGADA- S=P=DELTA            )
C     CHAVE 2 LIGADA   --APERFEICOAR FACTORES QUADRATICOS(E RAIZES)
C
C     CHAVE 3 LIGADA- O PROGRAMA VOLTA AO PRINCIPIO
C
      DIMENSIØN A(50),G(50),H(50)
    1 PAUSE
      READ 100,DELTA
C     DELTA-ERRO RELATIVO OU ABSOLUTO ADMISSIVEL EM S OU P
  100 FØRMAT(E14.8)
C     N-GRAU DO POLINOMIO
      READ 101,N
  101 FØRMAT(I5)
C     LEITURA DOS COEFICIENTES DO POLINOMIO
      READ 100,A0,(A(I),I=1,N)
      DØ 2 I=1,N
    2 A(I)=A(I)/A0
C     DETECCAO DE RAIZES NULAS
    3 IF(A(N))4,5,4
    5 N=N-1
      PRINT 102
  102 FØRMAT(14H .00000000E+00)
      GØ TØ 3
    4 IF(SENSE SWITCH2)6,7
    7 IF(SENSE SWITCH1)6,8
    8 S=DELTA
      P=DELTA
      GØ TØ 9
    6 READ 100,S,P
    9 IF(N-2)10,11,120
C     F(Z)=Z+A(1)
   10 Z=-A(1)
      PRINT 100,Z
      GØ TØ 1
C     F(Z)=Z**2-S*Z+P=Z**2+A(1)*Z+A(2)
   11 S=-A(1)
      P=A(2)
      CALL EQ2G(S,P)
      PUNCH 100,S,P
      GØ TØ 1
```

```
  120 IT=0
   12 CALL DIVQ(N,S,P,A,G)
C     DETERMINACAO DE G(Z),G(N-1) E G(N)
      CALL DIVQ(N-2,S,P,G,H)
C     DETERMINACAO DE H(Z),H(N-3) E H(N-2)
      IF(N-3)13,14,15
   13 STØP
   14 HNM3=1.
      GØ TØ 16
   15 HNM3=H(N-3)
   16 C=S*H(N-2)-P*HNM3
      D=H(N-2)**2-C*HNM3
      DS=(HNM3*G(N)-H(N-2)*G(N-1))/D
      DP=(H(N-2)*G(N)-C*G(N-1))/D
      PR1=ABSF(DS/S)
C     PR1-VARIACAO RELATIVA DE S  (EM MODULO)
      PR2=ABSF(DP/P)
C     PR2-VARIACAO RELATIVA DE P  (EM MODULO)
      IF(PR1-3.)17,17,18
   17 S=S+DS
      GØ TØ 19
   18 S=3.*S*SINAL(DS)
   19 IF(PR2-6.)20,20,21
   20 P=P+DP
      GØ TØ 22
   21 P=6.*P*SINAL(DP)
   22 IF(PR1-DELTA)23,221,221
  221 IF(ABSF(DS)-DELTA)23,300,300
   23 IF(PR2-DELTA)24,231,231
  231 IF(ABSF(DP)-DELTA)24,300,300
  300 IT=IT+1
      IF(IT-50)301,302,302
  301 IF(SENSE SWITCH 3)1,12
  302 PRINT 303,S,P
  303 FØRMAT(16H50 ITERACOES  S=E14.8,4H  P=,E14.8/
     1  24HLER SUGESTOES PARA S E P)
      GØ TØ 6
   24 CALL EQ2G(S,P)
      PUNCH 100,S,P
      IF(SENSE SWITCH 2)6,25
C     DETERMINADO UM FACTOR Z**2-S*Z+P
C     G(Z) TOMA O PAPEL DE F(Z)
   25 N=N-2
      DØ 26 I=1,N
   26 A(I)=G(I)
      GØ TØ 9
      END
```

```
C          SUBRØUTINE DIVQ(L,S,P,U,V)
C          V(1)=U(1)+S   V(2)=U(2)+S*V(1)-P
C          V(K)=U(K)+S*V(K-1)-P*V(K-2)
           DIMENSIØN U(50),V(50)
           T1=1.
           T0=0.
           DØ 1 K=1,L
           V(K)=U(K)+S*T1-P*T0
           T0=T1
     1     T1=V(K)
           RETURN
           END
```

```
           SUBRØUTINE EQ2G(S,P)
           D=S*S-4.*P
           X2=0.5*SQRTF(ABSF(D))
           X1=0.5*S
           IF(D)1,2,2
     2     X=X1+X2
           PRINT 3,X
     3     FØRMAT (E14.8)
           X=X1-X2
           PRINT 3,X
           RETURN
     1     PRINT 4,X1,X2
     4     FØRMAT(E14.8,3X,4H+(-),E14.8,3H I)
           RETURN
           END
```

```
           FUNCTIØN SINAL(X)
           IF(X)1,2,3
     1     SINAL=-1.
           RETURN
     2     SINAL=0.
           RETURN
     3     SINAL=1.
           RETURN
           END
```

Com $z^4-3z^3+4z^2-12z=z(z-3)(z^2+4)$, $\delta=10^{-8}$ e chaves 1 e 2 desligadas, obtivemos

```
 .00000000E+00
-.75246150E-08    +(-) .20000000E+01  I
 .30000000E+01
```

Bibliografia: [4], [9], [27], [35], [37].

BIBLIOGRAFIA

[1] ALT, Franz (editor) — *Advances in Computers*. New York. Academic Press. 1961. Vol. 2.

[2] ASHER, M. & FORSYTHE, G. E. — *SWAC experiments on the use of orthogonal polynomials for data-fitting*. J. Assoc. Computing Mach. 5 (1): p. 9-21. 1958.

[3] BODEWIG, E. — *Matrix Calculus*. 2nd. Ed. Amsterdam. North-Holland Publishing Company. 1959.

[4] BUCKINGHAM, R. A. — *Numerical Methods*. London. Sir Isaac Pitman & Sons, Ltd. 1962.

[5] CLENSHAW, C. W. — *A note on the summation of* CHEBYSHEV *series*. Math. Tables and Aids Computation. *9*: p. 118-120. 1955.

[6] ———————— CHEBYSHEV *series for mathematical functions*. London. Her Majesty's Stationery Office. 1962. National Physical Laboratory Mathematical Tables Series, Vol. 5.

[7] ———————— — *Curve fitting with a digital computer*. Computer J. *2*(4): p. 170-173. 1960.

[8] CLENSHAW, C. W., MILLER, G. F. & WOODGER, M. — *Handbook Series Special Functions Algorithms for Special Functions I*. Numerische Math. *4* (5): p. 403-419. 1963.

[9] DURAND, E. — *Solutions numériques des équations algébriques. Equations du type* $F(x)=0$. *Racines d'un polinôme*. Paris. Masson & Cie. Editeurs. 1960. Tome I.

[10] ———————— — *Solutions numériques des équations algébriques. Systèmes de plusieurs équations. Valeurs propres des matrices*. Paris. Masson & Cie. Editeurs. 1961. Tome II.

[11] FADDEEVA, V. N. — *Computational methods of Linear Algebra*. New York. Dover Publications, Inc. 1959.

[12] FORSYTHE, G. E. — *Generation and use of orthogonal polynomials for data-fitting with a digital computer*. J. Soc. Industr. Appl. Math. *5* (2): p. 74-88. June 1957.

[13] FOX, L. — *Numerical solution of ordinary and partial differential equations*. Oxford. Pergamon Press. 1962.

[14] FREITAS, A. César de — *Análise Numérica e Análise Funcional*. Lisboa. Instituto de Alta Cultura. 1963.

[15] ———————— — *Métodos Numéricos em Álgebra Linear* — I. Lisboa. Instituto de Alta Cultura. 1960.

[16] ———————— — *Cálculos com Números Aproximados*. Lisboa. Instituto de Alta Cultura. 1960.

[17] GEORGE, Richard — *Algorithm 120*. Commun. Assoc. Computing Mach. *5* (8): p. 437. 1962.

[18] ———————— — *Certifications of algorithm 120 and matrix inversion by* GAUSS-JORDAN. Commun. Assoc. Computing Mach. *6* (1): p. 40. 1963.

[19] GONÇALVES, J. Vicente — *Curso de Álgebra Superior.* Lisboa.

[20] HOLLINGDALE, S. H. & TOOTILL, G. C. — *Electronic Computers.* Pelican Book A524. 1965.

[21] I. B. M. — *IBM 1620 Data Processing System Reference Manual.* San Jose, Calif., IBM, (A26-4500).

[22] ——— — *IBM 1620 FORTRAN II Programming System Reference Manual.* San Jose, Calif., IBM, 1964. (C26-5876-2).

[23] ——— — *IBM 1620 FORTRAN II Operator's Guide.* San Jose, Calif., IBM, 1962 (C26-5662-2).

[24] LANCZOS, Cornelius — *Applied Analysis.* London. Sir Isaac Pitman & Sons Ltd. 1957.

[25] Mc CRACKEN, Daniel D. — *A guide to Fortran Programming.* New York. John Wiley & Sons, Inc. 1963.

[26] MINEUR, H. — *Techniques de Calcul Numérique.* Paris. Librairie Polytechnique Ch. Béranger. 1952.

[27] NATIONAL PHYSICAL LABORATORY — *Modern Computing Methods.* 2nd. Ed. London. Her Majesty's Stationery Office. 1961. «Notes on Applied Science», n.º 16.

[28] RALSTON, A. & WILF, H. S. (editores) — *Mathematical Methods for Digital Computers.* New York. John Wiley. 1962.

[29] SCARBOROUGH, J. B. — *Numerical Mathematical Analysis.* 5.ª ed. London. Oxford University Press. 1962.

[30] SCHWARZ, H. R. — *An introduction to Algol (procedure gjr, proposed by* H. RUTISHAUSER) Commun. Assoc. Computing Mach. *5* (2): p. 94. 1962.

[31] SHANTZ, P. W. — *Algorithm 220* GAUSS-SEIDEL. — Commun. Assoc. Computing. Mach. *6*(12): p. 739. Dez. 1963.

[32] USPENSKY, J. V. — *Theory of equations.* London. McGraw-Hill. 1948.

[33] VARGA, R. S. — *Matrix iterative analysis.* Englewood Cliffs, New Jersey. Prentice-Hall, Inc. 1962.

[34] WILF, Herbert S. — *Mathematics for the Physical Sciences.* New York. John Wiley and Sons, Inc. 1962.

[35] WILKINSON, J. H. — *The evaluation of the zeros of ill-conditioned polynomials.* Numerische Math. *1* (3): p. 150-180. 1959.

[36] ——— — *Handbook Series Linear Algebra. Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection.* Numerische Math. *4* (4): p. 362-367. 1962.

[37] ZAGUSKIN, V. L. — *Handbook of Numerical Methods for the Solution of Algebraic and Transcendental Equations.* Oxford. Pergamon Press. 1961.

[19] Gonçalves, J. Vicente — *Curso de Álgebra Superior.* Lisboa.

[20] Hollingdale, S. H. & Tootill, G. C. — *Electronic Computers.* Pelican Book A524. 1965.

[21] I. B. M. — *IBM 1620 Data Processing System Reference Manual.* San Jose, Calif., IBM, (A26-4500).

[22] ——— *IBM 1620 FORTRAN II Programming System Reference Manual.* San Jose, Calif., IBM, 1964. (C26-5876-2).

[23] ——— *IBM 1620 FORTRAN II Operator's Guide.* San Jose, Calif., IBM, 1962 (C26-5662-2).

[24] Lanczos, Cornelius — *Applied Analysis.* London. Sir Isaac Pitman & Sons Ltd. 1957.

[25] Mc Cracken, Daniel D. — *A guide to Fortran Programming.* New York. John Wiley & Sons, Inc. 1963.

[26] Mineur, H. — *Techniques de Calcul Numérique.* Paris. Librairie Polytechnique Ch. Béranger. 1952.

[27] National Physical Laboratory — *Modern Computing Methods.* 2nd. Ed. London. Her Majesty's Stationery Office. 1961. «Notes on Applied Science», n.º 16.

[28] Ralston, A. & Wilf, H. S. (editores) — *Mathematical Methods for Digital Computers.* New York. John Wiley. 1962.

[29] Scarborough, J. B. — *Numerical Mathematical Analysis.* 5.ª ed. London. Oxford University Press. 1962.

[30] Schwarz, H. R. — *An introduction to Algol (procedure gjr, proposed by* H. Rutishauser) Commun. Assoc. Computing Mach. *5* (2): p. 94. 1962.

[31] Shantz, P. W. — *Algorithm 220* Gauss-Seidel.—Commun. Assoc. Computing. Mach. *6*(12): p. 739. Dez. 1963.

[32] Uspensky, J. V. — *Theory of equations.* London. McGraw-Hill. 1948.

[33] Varga, R. S. — *Matrix iterative analysis.* Englewood Cliffs, New Jersey. Prentice-Hall, Inc. 1962.

[34] Wilf, Herbert S. — *Mathematics for the Physical Sciences.* New York. John Wiley and Sons, Inc. 1962.

[35] Wilkinson, J. H. — *The evaluation of the zeros of ill-conditioned polynomials.* Numerische Math. *1* (3): p. 150-180. 1959.

[36] ——— — *Handbook Series Linear Algebra. Calculation of the eigenvalues of a symmetric tridiagonal matrix by the method of bisection.* Numerische Math. *4* (4): p. 362--367. 1962.

[37] Zaguskin, V. L. — *Handbook of Numerical Methods for the Solution of Algebraic and Transcendental Equations.* Oxford. Pergamon Press. 1961.