INPE-503-LAFE

DPD

APROXIMAÇÃO DE FUNÇÕES CONTÍNUAS POR FRAÇÕES RACIONAIS, NO SENTIDO DE CHEBYSHEV

por

Mucio Roberto Dias

Agosto de 1974

cc.:30

SERVIÇO PÚBLICO FEDERAL
CONSELHO NACIONAL DE PESQUISAS
INSTITUTO DE PESQUISAS ESPACIAIS
São José dos Campos - Estado de S. Paulo - Brasil

APROXIMAÇÃO DE FUNÇÕES CONTÍNUAS POR FRAÇÕES RACIONAIS, NO SENTIDO DE CHEBYSHEV

Este trabalho foi apresentado, como parte dos requisitos necessários à obtenção do grau de Mestre em Computação Aplicada, pelo Engenheiro Mucio Roberto Dias deste Instituto, tendo como orientador o Dr. Plinio Tissi. A presente publicação foi autorizada pelo abaixo assinado.

F de Mendonça
Fernando de Mendonça
Diretor Geral

RESUMO

Este trabalho tem por objetivo a apresentação de um algoritmo que gera a melhor aproximação, no sentido de Chebyshev, de funções contínuas por frações racionais. Serão vistos os detalhes da implementação deste algoritmo, assim como alguns exemplos de resultados obtidos serão mostrados.

INDICE

SÍMBOLOS USADOS

f	Função a ser aproximada.
$R_{mn}(x)$	Fração racional cujo polinômio numerador tem grau menor ou igual a m e cujo polinômio denominador tem grau menor ou igual a n.
r_{mn}	$\max\limits_{x \in \|a,b\|} \|f(x) - R_{mn}(x)\|$
$R^*_{mn}(x)$	Fração racional escolhida dentre as $R_{mn}(x)$ cujo r_{mn} é mínimo.
r^*_{mn}	Mínimo de r_{mn}.
$A_m(x)$	Polinômio numerador de $R_{mn}(x)$.
$B_n(x)$	Polinômio denominador de $R_{mn}(x)$.
a_j	Coeficiente de x^j em $A_m(x)$.
b_j	Coeficiente de x^j em $B_n(x)$.
$A^*_m(x)$	Polinômio numerador de $R^*_{mn}(x)$.
$B^*_n(x)$	Polinômio denominador de $R^*_{mn}(x)$.
$E^*(x)$	$f(x) - R^*_{mn}(x)$, erro de $R^*_{mn}(x)$.

ν	Diferença entre graus dos polinômios $A_m(x)$ e $A^*_m(x)$.
μ	Diferença entre graus dos polinômios $B_n(x)$ e $B^*_n(x)$.
$[a,b]$	Intervalo no qual a aproximação é desejada (a,b reais e $a < b$)
$R^{(k)}_{mn}(x)$	Aproximação disponível após a k-ésima iteração.
$E^{(k)}(x)$	$f(x) - R^{(k)}_{mn}(x)$, erro após a k-ésima iteração.
$\underset{\sim}{x}^{(k)}$	Vetor de extremantes após a k-ésima iteração.
$a^{(k)}_j$	Coeficiente de x^j no numerador de $R^{(k)}_{mn}$.
$b^{(k)}_j$	Coeficiente de x^j no denominador de $R^{(k)}_{mn}$.
$x^{(k)}_i$	Elemento de $\underset{\sim}{x}^{(k)}$.

CAPÍTULO I

INTRODUÇÃO

1.1 - ESTABELECIMENTO DO PROBLEMA.

Dados uma função f real contínua num intervalo $[a,b]$ e dois inteiros não negativos m e n, deseja-se obter uma fração racional da forma

$$R_{mn}(x) = \frac{A_m(x)}{B_n(x)} = \frac{\sum_{j=0}^{m} a_j x^j}{\sum_{j=0}^{n} b_j x^j} \tag{1.1}$$

que minimiza

$$r_{mn} = \max_{x \in [a,b]} [f(x) - R_{mn}(x)] \tag{1.2}$$

Tal fração racional, anotada por R^*_{mn} e, em geral, tendo a forma

$$R^*_{mn}(x) = \frac{A^*_m(x)}{B^*_n(x)} = \frac{\sum_{j=0}^{m-\nu} a^*_{j+\nu} x^j}{\sum_{j=0}^{n-\mu} b^*_{j+\mu} x^j} \tag{1.3}$$

onde $\quad 0 \leq \nu \leq m \quad$ e $\quad 0 \leq \mu \leq n \quad$ e $\quad a^*_m, b^*_n \neq 0$, é conhecida como melhor aproximação, no sentido de Chebyshev, de $f(x)$ em $[a,b]$.

Anota-se por r^*_{mn} ao mínimo de r_{mn}, ou seja:

$$r^*_{mn} = \max_{x \in [a,b]} [f(x) - R^*_{mn}(x)] \tag{1.4}$$

Como citado por Ralston [12] e Frazer e Hart [6], a aproximação por frações racionais ($n \neq 0$) conduz a erros máximos menores, com o mesmo ou menor esforço desprendido nos cálculos, que a aproximação polinomial. Esta é, sem dúvida, uma das razões que tornam importante tal tipo de aproximação.

1.2 - TEOREMAS BÁSICOS

A existência e unicidade da melhor aproximação são dadas pelo teorema abaixo:

Teorema 1:- Entre todas as funções $R_{mn}(x)$, existe sòmente uma para a qual r_{mn} é mínimo (considerando-se idênticas duas frações racionais que são iguais, quando postas sob a forma irredutível).

Por outro lado, a completa caracterização da melhor aproximação tem-se de:

Teorema 2 (de Chebyshev):- Se a fração $R^*_{mn}(x) = \frac{A^*_m(x)}{B^*_n(x)}$ é irredutível, então, se r^*_{mn} não é zero, o número de pontos consecutivos, pontos de $[a,b]$, nos quais $E^*(x) = f(x) - R^*_{mn}(x)$ toma seu máximo valor r^*_{mn} com sinais alternados, não é menor que $N+2-d$ onde $d = \min(\mu,\nu)$ e $N = m+n$.

As provas dos dois teoremas estão em Achieser [1].

A quantidade $d = \min(\mu,\nu)$ é chamada de defeito de $R_{mn}(x)$. Se $d > 0$, o caso em questão é dito degenerado ou, que nele há degenerescência.

No que se segue, supõe-se $d = 0$, condição esta que não é muito restritiva, de acôrdo com Ralston [12]. Esse autor apresenta ainda um teorema que permite a identificação de casos degenerados, cuja aplicação prática, entretanto, não é possível em muitos casos, segundo Watson [15].

1.3 - COMENTÁRIOS SÔBRE OS ALGORITMOS EXISTENTES.

Os algoritmos existentes consistem, na sua maioria, em extensões do segundo algoritmo de Remez, idealizado para aproximação polinomial de funções contínuas, no sentido de Chebyshev. Tal extensão conduz, por imposição própria do algoritmo, a sistemas de equações algébricas não lineares. Alguns autores como Ralston [12], Frazer e Hart [6] e Vlach [14], propõem métodos iterativos para soluções de tais sistemas; outros, como

Collatz [5] evitam a solução direta daqueles sistemas fazendo com que o problema recaia numa busca de autovalores. A prova da convergência de tais algoritmos só tem sido conseguida em alguns casos, sob a hipótese de uma aproximação inicial suficientemente boa. Uma de tais provas foi dada por Ralston [11]. Entretanto, o próprio Ralston [12] identificou uma classe de problemas, chamados "quase-degenerados", para as quais uma aproximação inicial extremamente boa é requerida para a obtenção de bons resultados. Com isso, estes casos "quase-degenerados" apresentam-se como uma limitação na aplicabilidade de tais algoritmos.

Outros algoritmos têm sido apresentados, como o de Ishizaki e Watanabe [8] e o de Watson [15], os quais caracterizam-se por uma discretização do problema. Aliás, tais algoritmos são apresentados para efetuar aproximações não lineares em geral, não se tratando especificamente do caso de aproximação por frações racionais. Nestes casos, toma-se um número grande de pontos do intervalo, no qual se deseja a aproximação, e uma equação de êrro, função dos coeficientes da função aproximante, é associada a cada ponto. Então, parte-se para a minimização do êrro máximo associado a estes pontos. Esta minimização é conseguida por programação linear, depois de uma linearização aproximada de cada uma das equações de êrro citadas acima. Entretanto, embora bons resultados tenham sido conseguidos com estes métodos, estes trazem em si uma dificuldade básica: é que, para algumas funções, um número muito grande de divisões no intervalo é requerido, sem que se tenham meios de fixar "a priori" o número mínimo de tais divisões, que conduza a uma aproximação satisfatória.

O algoritmo apresentado neste trabalho, como se verá no capítulo seguinte, trata o problema sem discretizá-lo e se baseia, essencialmente, no teorema de Chebyshev, citado anteriormente. Entretanto, foram mantidos alguns aspectos, julgados positivos, dos algoritmos de Ishizaki e Watanabe e o de Watson. Assim é que, também no algoritmo apresentado aqui, utilizou-se programação linear, através de linearizações semelhantes às encontradas nos algoritmos acima.

CAPÍTULO II

DESCRIÇÃO DO ALGORITMO

2.1 - APROXIMAÇÃO INICIAL.

O algoritmo parte de uma aproximação inicial

$$R_{mn}^{(o)}(x) = \frac{\sum\limits_{j=0}^{m} a_j^{(o)} x^j}{\sum\limits_{j=1}^{n} b_j^{(o)} x^j + 1} \qquad (2.1)$$

e de um conjunto de N+2 extremantes, de erros com sinais alternados (abcissas de máximos ou mínimos locais), da função:

$$E^{(o)}(x) = f(x) - R_{mn}^{(o)}(x) \qquad (2.2)$$

Êste conjunto de extremantes iniciais será anotado por:

$$\underset{\sim}{x}^{(o)} = (x_1^{(o)}, x_2^{(o)}, \ldots, x_{N+2}^{(o)}) \qquad (2.3)$$

Por sugestão de Werner et al.[16], $R_{mn}^{(o)}$ é tomada como a fração racional que produz N+1 zeros em $E^{(o)}(x)$, zeros estes coincidentes com os zeros do polinômio de Chebyshev de ordem N+1, depois de uma transformação linear para o intervalo $[a,b]$, de tal modo que os zeros de $E^{(o)}(x)$

devem ser:

$$y_i = \frac{a+b}{2} + \frac{b-a}{2} \cdot \cos\left[(2i-1)\ /2(N+1)\right] \tag{2.4}$$

$$i = 1, 2, \ldots, N+1$$

Deste modo, $R_{mn}^{(o)}(x)$ pode ser obtido do sistema de equações lineares abaixo:

$$\sum_{j=0}^{m} a_j^{(o)} y_i^j \ - \ f(y_i) \sum_{j=1}^{n} b_j^{(o)} y_i^j \ = \ f(y_i) \tag{2.5}$$

$$i = 1, 2, \ldots, N+1$$

Determinada $R_{mn}^{(o)}(x)$, passa-se a buscar os extremantes internos $x_2^{(o)}, x_3^{(o)}, \ldots, x_{N+1}^{(o)}$ como máximos e mínimos de $E^{(o)}(x)$, e $x_1^{(o)} = a$ e $x_{N+2}^{(o)} = b$, ainda como sugestão de Werner et al.[16], tais que: $E^{(o)}(x_i^{(o)})$ e $E^{(o)}(x_{i+1}^{(o)})$ tenham sinais contrários para $i = 1, 2, \ldots, N+1$, e

$$x_i < x_{i+1} \qquad (i = 1, 2, \ldots, N+1)$$

Esta aproximação inicial tem se mostrado satisfatória, como se verá nos resultados obtidos.

2.2 - ALGORITMO.

Como foi visto, a solução do problema $R_{mn}^{*}(x)$ é encontrada quando se tiver:

$$E^*(x_i^*) - (-1)^i\lambda = f(x_i^*) - R_{mn}^*(x_i^*) - (-1)^i\lambda = 0 \qquad (2.6)$$

onde $\lambda = \pm r_{mn}^*$

$i = 1, 2, \ldots, N+2$

e $x^* = (x_1^*, x_2^*, \ldots, x_{N+2}^*)$ é o vetor de extremante finais, com

$x_i^* < x_{i+1}^* \qquad (i = 1, 2, \ldots, N+1)$

Admitindo-se que a função f tenha derivada em $[a,b]$, não há dúvida de que a solução final se caracterizará, ainda, por:

$$\frac{\partial E^*(x_i^*)}{\partial x_i^*} = f'(x_i^*) - \frac{\partial R_{mn}^*(x_i^*)}{\partial x_i^*} = 0 \qquad (2.7)$$

$i = 2, 3, \ldots, N+1$

Obriga-se, ainda seguindo a sugestão de Werner et al. [16], que $x_1^* = a$ e $x_{N+2}^* = b$. Por fim, faz-se com que:

$$R_{mn}^*(x) = \frac{\sum_{j=0}^{m} a_j^* x^j}{\sum_{j=1}^{n} b_j^* x^j + 1}$$

pois, senão, a determinação dos coeficientes a_j^* $(j = 0, \ldots, m)$ e b_j^* $(j = 1, \ldots, n)$ só seria possível a menos de uma constante multiplicativa.

Para a solução do problema, propõe-se, neste trabalho, o esquema iterativo que se segue.

Sejam F e G as funções definidas como abaixo:

$$F(x, a_0, a_1, \ldots, a_m, b_1, b_2, \ldots, b_n) =$$

$$= f(x) - \frac{\sum_{j=0}^{m} a_j x^j}{\sum_{j=1}^{n} b_j x^j + 1} \tag{2.8}$$

e $$G(x, a_0, a_1, \ldots, a_m, b_1, b_2, \ldots, b_n) = \frac{\partial F}{\partial x} =$$

$$= f'(x) - \frac{\partial}{\partial x}\left[\frac{\sum_{j=0}^{m} a_j x^j}{\sum_{j=1}^{n} b_j x^j + 1}\right] \tag{2.9}$$

Admitindo-se conhecidos a aproximação depois da k-ésima iteração $R_{mn}^{(k)}(x)$ e o conjunto $\underset{\sim}{x}^{(k)}$ de extremantes correspondentes, a aproximação $R_{mn}^{(k+1)}(x)$ e o vetor $\underset{\sim}{x}^{(k+1)}$ são calculados por meio da determinação de Δx_i $(i = 2, \ldots, N+1)$ e Δa_i $(i = 0, \ldots, m)$ e Δb_i $(i = 1, 2, \ldots, n)$, incrementos sôbre os extremantes interiores de $\underset{\sim}{x}^{(k)}$ e sôbre os coeficientes do numerador e denominador de $R_{mn}^{(k)}$, que minimizam H sujeitos a:

$$\left|\left[F(x_i^{(k)} + \Delta x_i, a_0^{(k)} + \Delta a_0, \ldots, a_m^{(k)} + \Delta a_m, b_1^{(k)} + \Delta b_1, \ldots, b_n^{(k)} + \Delta b_n) - \right.\right.$$

$$\left.\left. - (-1)^i \lambda\right]\right| \leq H \tag{2.10}$$

$$i = 1, 2, \ldots, N+2$$

$$\text{e} \qquad \Delta x_1 = \Delta x_{N+2} = 0$$

e

$$\left|\left[G(x_i^{(k)} + \Delta x_i,\ a_o^{(k)} + \Delta a_o,\ \ldots,\ a_m^{(k)} + \Delta a_m,\ b_1^{(k)} + \Delta b_1,\ \ldots,\ b_n^{(k)} + \Delta b_n)\right]\right| \leq \leq \varepsilon H \qquad (2.11)$$

onde ε é uma constante determinada

$$(0 < \varepsilon << 1)$$

$$i = 2, 3, \ldots, N+1$$

Admitindo-se:

$$F(x_i^{(k)} + \Delta x_i,\ a_o^{(k)} + \Delta a_o,\ \ldots,\ a_m^{(k)} + \Delta a_m,\ b_1^{(k)} + \Delta b_1,\ \ldots,\ b_n^{(k)} + \Delta b_n) \approx$$

$$\approx F(x_i^{(k)},\ a_o^{(k)},\ \ldots,\ a_m^{(k)},\ b_1^{(k)},\ \ldots,\ b_n^{(k)}) + \frac{\partial F}{\partial x_i^{(k)}} \Delta x_i + \frac{\partial F}{\partial a_o^{(k)}} \Delta a_o + \ldots$$

$$\ldots + \frac{\partial F}{\partial a_m^{(k)}} \Delta a_m + \frac{\partial F}{\partial b_1^{(k)}} \Delta b_1 + \ldots + \frac{\partial F}{\partial b_n^{(k)}} \Delta b_n \qquad (2.12)$$

e

$$G(x_i^{(k)} + \Delta x_i,\ a_o^{(k)} + \Delta a_o,\ \ldots,\ a_m^{(k)} + \Delta a_m,\ b_1^{(k)} + \Delta b_1,\ \ldots,\ b_n^{(k)} + \Delta b_n) =$$

$$G(x_i^{(k)},\ a_o^{(k)},\ \ldots,\ a_m^{(k)},\ b_1^{(k)},\ \ldots,\ b_n^{(k)}) + \frac{\partial G}{\partial x_i^{(k)}} \Delta x_i + \frac{\partial G}{\partial a_o^{(k)}} \Delta a_o + \ldots.$$

$$\ldots + \frac{\partial G}{\partial a_m^{(k)}} \Delta a_m + \frac{\partial G}{\partial b_1^{(k)}} \Delta b_1 + \ldots + \frac{\partial G}{\partial b_n^{(k)}} \Delta b_n \qquad (2.13)$$

o problema anterior, que recai em programação linear, pode ser posto como: deseja-se determinar x_i $(i = 2, \ldots, N+1)$, a_i $(i = 0, \ldots, m)$ e b_i $(i = 1, \ldots, n)$ que minimizam H sujeitos a:

$$\Big|\Big[F(x_i^{(k)}, a_o^{(k)}, \ldots, a_m^{(k)}, b_1^{(k)}, \ldots, b_n^{(k)}) + \frac{\partial F}{\partial x_i^{(k)}} \Delta x_i + \frac{\partial F}{\partial a_o^{(k)}} \Delta a_o + \ldots$$

$$\ldots + \frac{\partial F}{\partial a_m^{(k)}} \Delta a_m + \frac{\partial F}{\partial b_1^{(k)}} \Delta b_1 + \ldots + \frac{\partial F}{\partial b_n^{(k)}} \Delta b_n - (-1)^i \lambda\Big]\Big| \leq H \quad (2.14)$$

$$i = 1, 2, \ldots, N+2 \quad \text{e} \quad \Delta x_1 = \Delta_{N+2} = 0$$

e

$$\Big|\Big[G(x_k^{(k)}, a_o^{(k)}, \ldots, a_m^{(k)}, b_1^{(k)}, \ldots, b_n^{(k)}) + \frac{\partial G}{\partial x_i^{(k)}} \Delta x_i + \frac{\partial G}{\partial a_o^{(k)}} \Delta a_o + \ldots$$

$$\ldots + \frac{\partial G}{\partial a_m^{(k)}} \Delta a_m + \frac{\partial G}{\partial b_1^{(k)}} \Delta b_1 + \ldots + \frac{\partial G}{\partial b_n^{(k)}} \Delta b_n\Big]\Big| \leq \varepsilon \; H \qquad (2.15)$$

$$i = 2, 3, \ldots, N+1$$

Pareceu conveniente, ainda, exigir, para que o processo não se deteriorasse pela superposição de futuros extremantes, que:

$$x_i^{(k)} + \Delta x_i + \delta \leq x_{i+1}^{(k)} + \Delta x_{i+1} \tag{2.16}$$

com $\quad i = 1, 2, \ldots, N+1$

$$\Delta x_1 = \Delta x_{N+2} = 0$$

para algum $\quad \delta > 0.$

De posse da solução anterior, fazemos, então

$$a_i^{(k+1)} = a_i^{(k)} + \Delta a_i \qquad i = 0, \ldots, m \tag{2.17}$$

$$b_i^{(k+1)} = b_i^{(k)} + \Delta b_i \qquad i = 1, \ldots, n \tag{2.18}$$

$$x_i^{(k+1)} = x_i^{(k)} + \Delta x_i \qquad i = 1, \ldots, N+2 \tag{2.19}$$

Neste ponto, mostrou-se necessária uma pequena correção sobre $x_i^{(k+1)}$ $(i = 2, \ldots, N+1)$, para que estes coincidissem com os verdadeiros extremantes de $E^{(k+1)}(x) = f(x) - R_{mn}^{(k+1)}(x)$.

O processo deve ser repetido até a j-ésima iteração, na qual os erros associados aos extremantes satisfaçam ao teorema de Chebyshev, ou seja:

$$E^{(j)}(x_i^{(j)}) = -E^{(j)}(x_{i+1}^{(j)}) \quad (i = 1, 2, \ldots, N+1) \tag{2.20}$$

$$E^{(j)}(x_i^{(j)}) = \max_{x \in [a,b]} [E^{(j)}(x)]$$

É evidente que, no caso prático, pequenas tolerâncias são dadas às condições acima.

CAPÍTULO III

DETALHES DA IMPLEMENTAÇÃO

3.1 - GENERALIDADES.

As subrotinas, cujas descrições são feitas a seguir e cujas listagens compõem o apêndice I, foram escritas em Fortran IV, nível H, e usou-se o computador B-6700 para a obtenção dos resultados que serão mostrados no próximo capítulo. A escolha da linguagem Fortran, ao invés de outras mais apropriadas para o computador em questão, por exemplo Algol, se deveu ao fato de que, assim, as subrotinas se tornam mais facilmente implementáveis em outros computadores, especialmente nos de menor porte.

3.2 - DESCRIÇÃO DAS SUBROTINAS

Todo o algoritmo descrito no capítulo anterior está contido na subrotina RAPROX que, por sua vez, chama pelas seguintes subrotinas e funções externas: RESEL, STURM, NEWMAX, ERRO, IGUAL, REGMAG, VALNUM, MPDIV, FUNCAO, DERRO, DERIV, DELMAX, DSINAL. A seguir, encontra-se descrito o procedimento contido em cada uma delas, fazendo-se, na descrição, uso dos próprios nomes das variáveis encontradas nas listagens do apêndice I, para facilitar ao leitor. Assim, tem-se:

SUBROTINA RAPROX

Parâmetros:

M - grau do polinômio numerador

N - grau do polinômio denominador

AO - extremo inferior do intervalo onde se deseja a aproximação

BO - extremo superior do intervalo onde se deseja a aproximação

IERRO - denota a ocorrência ou não de anormalidades durante a execução

ISAIDA - especifica tipo de saída desejada.

Os coeficientes dos polinômios numerador e denominador serão armazenados nos vetores A e B, respectivamente, que poderão ser transferidos da subrotina RAPROX, via COMMON de "label" COEF, como poderá ser visto.

Todos os procedimentos realizados por esta subrotina, na ordem de execução, são dados abaixo:

- gera-se a aproximação inicial através do sistema de equações lineares descrito no ítem 2.1 do capítulo anterior. Caso o sistema de equações possua uma matriz singular ou a aproximação gerada possua polos no intervalo, interrompe-se a execução do programa. De posse da aproximação inicial, passa-se à busca dos extremantes, pela chamada da subrotina

NEWMAX que será, posteriormente, discutida.

- Montam-se as matrizes S e TS do sistema de inequações correspondentes a (2.14), (2.15) e (2.16). Para isso, todas as derivadas parciais da função F em (2.14) são calculadas pela função externa DERRO, assim como todas as derivadas de G em (2.15) são calculadas por DERIV. Como se verá no apêndice I, o ϵ que aparece em (2.13) é feito 1.0×10^{-6}, assim como o δ em (2.14) é feito igual ao comprimento do intervalo dividido por dez vezes o número total de extremantes.

- Com as matrizes S e TS montadas, passa-se à execução do Simplex, que, na presente implementação, é representado pelas subrotinas IGUAL e REGMAG.

- De posse dos novos coeficientes da nova aproximação e dos novos extremantes, assegura-se a não ocorrência de polos no intervalo através da subrotina STURM e, a seguir, se uma das condições abaixo forem satisfeitas, o processo é dado por terminado. Estas condições são:

  1) O valor absoluto da máxima diferença que ocorreu nos coeficientes da etapa anterior à presente etapa foi inferior a 10^{-8}. Neste caso, o processo está convergindo tão lentamente que achou-se conveniente sua interrupção.

  2) Os êrros nos extremantes satisfazem ao teorema de Chebyshev, com uma diferença entre o maior e o menor, dos elementos do vetor de valores absolutos dos erros, inferior a 10% do menor elemento daquele vetor.

3) O processo não convergiu em 20 iterações. Neste caso, também, achou-se conveniente a interrupção do processo.

Caso nenhuma das condições anteriores tenha sido satisfeita, recomeça-se o algoritmo. A seguir, são dadas todas as opções dos parâmetros ISAIDA e IERRO.

IERRO = 1 - Resultado satisfatório foi obtido.

IERRO = 2 - Convergência lenta. Variação máxima, em valor absoluto, nos coeficientes foi inferior a 10^{-8}.

IERRO = 3 - Sistema de equações que determinaria a aproximação inicial resultou com matriz singular.

IERRO = 4 - O processo não convergiu em 20 iterações.

IERRO = 5 - Simplex acusou solução inviável.

IERRO = 6 - Aproximação possui polos no intervalo.

ISAIDA = 0 - Nenhum resultado é escrito.

ISAIDA = 1 - É escrito o valor de IERRO e, se IERRO = 1, são escritos os coeficientes da melhor aproximação, assim como o número de iterações requeridas.

ISAIDA = 2 - Além de produzir as saídas correspondentes a ISAIDA = 1, os coeficientes, assim como os extremantes e respectivos erros em cada iteração do algoritmo, são também escritos.

ISAIDA = 3 - Além dos resultados produzidos por ISAIDA = 2, também as matrizes S e TS de cada iteração são escritas.

SUBROTINA RESEL

Parâmetros:

NTOT - ordem da matriz do sistema

A - matriz dos coeficientes do sistema

C - vetor dos termos independentes

D - vetor que, na saída, contém a solução do sistema

IERRO - denota ocorrência ou não de anormalidades na execução

MM - número de linhas dimensionadas para A no programa principal

Esta subrotina resolve sistema de equações lineares por inversão da matriz dos coeficientes. No processo de inversão é utilizado o método de redução de Gauss com pivoteamento do maior elemento da coluna que se estiver trabalhando. Caso a matriz A seja não singular, esta matriz contém, na saída, sua inversa. Caso contrário, a variável IERRO é feita igual a 3 e o processo é interrompido.

SUBROTINA STURM

Parâmetros:

XX - vetor que contém os coeficientes do polinômio, em ordem decrescente de grau de seus termos.

IGRAU - grau do polinômio

AO - extremo inferior do intervalo

BO - extremo superior do intervalo

N - número de zeros no intervalo

Esta subrotina determina o número de raízes, contadas uma só vez caso a multiplicidade seja superior a 1, que um polinômio dado tem num intervalo dado. O método utilizado é o das sequências de Sturm descrito por Burnside e Panton [2] que, resumidamente, pode ser posto como: dados um polinômio $p(x)$ e um intervalo $[a,b]$, o número de raízes de $p(x)$, em $[a,b]$ coincide com a diferença entre o número de mudanças de sinal quando a e b são substituídos na sequência de polinômios abaixo, dita sequência de Sturm:

$$p(x),\ p_1(x),\ -p_2(x),\ \ldots,\ -p_r(x),$$

onde $p_1(x)$ é a derivada de $p(x)$ e $p_2(x), \ldots, p_r(x)$ são todos os restos não nulos obtidos no processo de determinação do máximo divisor comum entre $p(x)$ e $p_1(x)$.

Tomou-se o cuidado de não alterar os valores contidos em nenhum dos parâmetros de entrada.

SUBROTINA MPDIV

Parâmetros:

X - vetor que contém os coeficientes do primeiro polinômio, em ordem decrescente de grau de seus termos.

IGRAUX - grau do primeiro polinômio

Y - vetor que contém os coeficientes do segundo polinômio, em ordem decrescente de grau de seus termos.

IGRAUY - grau do segundo polinômio

Esta subrotina encontra o resto da divisão dos polinômios cujos coeficientes estão em X, pelo polinômio cujos coeficientes estão em Y. Os coeficientes do resto, assim como seu grau, são armazenados na saída, em ordem decrescente de grau, em X e IGRAUX, respectivamente. Esta subrotina é unicamente utilizada pela subrotina STURM, no processo de geração das sequências de STURM.

SUBROTINA NEWMAX

Parâmetros:

XLIM - comprimento do subintervalo que contém o extremante, quando determinado "na força bruta"

DEL - incremento para o cálculo da inclinação média

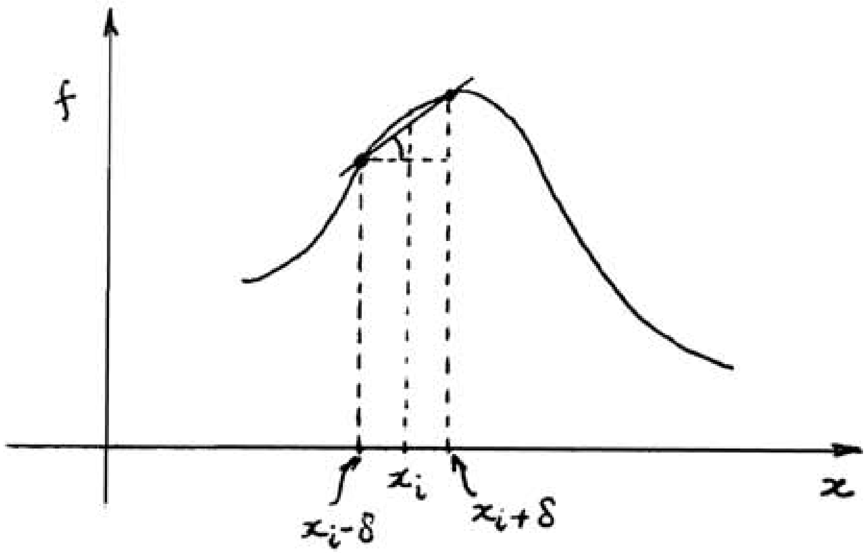
NITER - número máximo de iterações para a determinação do extremante

NTOT - número de extremantes

Esta subrotina determina os verdadeiros extremantes, a partir dos supostos extremantes obtidos em cada passo do algoritmo.

O esquema seguido para a determinação dos extremantes é o descrito por Wilde [17] como sendo devido a Kesten. Este esquema provém de uma modificação sôbre o método de Kiefer-Wolfowitz para a determinação de extremos de funções. Este método pode ser descrito, resumidamente, como abaixo:

Admita que se desejasse encontrar a abcissa de máximo da função f, da figura abaixo.



O método consiste em gerar uma sequência de abcissas x_0, x_1, x_2, ..., cujo limite é o ponto desejado. A partir, então, de um x_0 dado a sequência é gerada, por recorrência, por meio da seguinte expressão:

$$x_{i+1} = x_i + a_j \mathrm{sgn} \left[\frac{f(x_i+\delta) - f(x_i-\delta)}{2\delta} \right]$$

onde:- sgn é a função sinal, usualmente definida

$$\text{como:}\quad \operatorname{sgn}(u) = \begin{cases} 1 & \text{se} \quad u > 0 \\ 0 & \text{se} \quad u = 0 \\ -1 & \text{se} \quad u < 0 \end{cases}$$

- a_n é qualquer sequência tal que $a_{n+1} < a_n$;
- δ constante, tomado, aqui, como 10^{-8} do intervalo em que se deseja a aproximação.

Como um detalhe adicional, tem-se que o índice j, tomado a partir de 1, é incrementado de 1 sòmente se houve mudança de sinal da inclinação média, da etapa anterior à presente etapa. Neste trabalho, tomou-se $a_n = 1/100n$.

A determinação do extremante é dada por satisfatória se a inclinação média, em valor absoluto, foi inferior a 10^{-10}.

Espera-se que, em NITER iterações, seja possível conseguir a determinação do extremante pois, se tal não acontecer, usar-se-á o método da "procura bruta", assim denominado e sugerido por Golub e Smith [7]. Este método consiste em dividir o intervalo em consideração num número NDIV de intervalos iguais (neste trabalho NDIV foi tomado igual a 100) e examinar a ordenada em cada um dos extremos de cada intervalo. Os pontos à direita e à esquerda do ponto de ordenada máxima definem um novo intervalo sôbre o qual o processo é repetido. Esta subdivisão continua até que os comprimentos dos subintervalos fiquem menores que XLIM, tomado aqui co mo 10^{-8} vezes o comprimento do intervalo no qual a aproximação é desejada.

Os vetores A e B de coeficientes da aproximação, assim como o vetor X de extremantes, no qual esta subrotina produz modificações, são transferidos para a subrotina, via COMMON, de "label" COEF.

SUBROTINA FUNCAO

Parâmetros:

X - argumento

E - valor da função

Esta subrotina deve ser fornecida pelo usuário, já que, para qualquer X do intervalo onde se deseja a aproximação, o valor E correspondente da função a ser aproximada deve ser possível de ser obtido por esta subrotina. As variáveis X e E devem ser do tipo dupla precisão, para haver coerência com o restante das subrotinas.

SUBROTINA ERRO

Parâmetros:

X - ponto do intervalo onde se deseja calcular o erro

E - erro no ponto X.

Admitindo que f seja a função a ser aproximada e R_{mn} seja a aproximação no instante em que esta subrotina for chamada, o valor de E, de saída da subrotina, é:

$$E = f(x) - R_{mn}(x)$$

SUBROTINA VALNUM

Parâmetros:

A - vetor que contém coeficientes do polinômio em, ordem decrescente de grau

N - grau do polinômio

X - ponto onde se deseja avaliar o valor do polinômio

Y - valor do polinômio em X

Esta subrotina avalia polinômio cujos coeficientes encontram-se armazenados no vetor A, em ordem decrescente de grau, no ponto X, deixando o resultado em Y.

FUNÇÃO DELMAX

Parâmetros:

A - vetor do qual se deseja encontrar seu elemento, mínimo ou máximo, em valor absoluto

N - dimensão de A

KK - indica se se deseja o máximo ou mínimo dos elementos de A, em valor absoluto

Se KK = 1, DELMAX retorna com o menor, em valor absoluto, dos elementos do vetor A. Caso contrário (KK $\neq$ 1), DELMAX retorna com o maior, em valor absoluto, dos elementos de A.

FUNÇÃO DSINAL

Parâmetro:

X - argumento

Esta função faz com que DSINAL retorne com +1, 0 ou -1 conforme X seja positivo, nulo ou negativo, respectivamente.

FUNÇÃO DERRO

Parâmetros:

I - índice de linha na matriz S

J - índice de coluna na matriz S

Esta função fornece todos os coeficientes que dependem de cálculos de derivadas das primeiras N+2 linhas da matriz S. As N+2 primeiras inequações, cujos coeficientes estão em S, provêm de (2.14), de tal modo que a j-ésima inequação é:

$$0 \Delta x_1 + 0 \Delta x_2 + \ldots + \frac{\partial F}{\partial x_j^{(k)}} \Delta x_j + \ldots + 0 \Delta x_{N+2} + \frac{\partial F}{\partial a_o^{(k)}} \Delta a_o + \ldots + \frac{\partial F}{\partial a_m^{(k)}} \Delta a_m +$$

$$+ \frac{\partial F}{\partial b_1^{(k)}} \Delta b_1 + \ldots + \frac{\partial F}{\partial b_n^{(k)}} \Delta b_n - (-1)^j \lambda - H \leq - F(x_j^{(k)}, a_0^{(k)}, \ldots\ldots$$

$$\ldots, a_m^{(k)}, b_1^{(k)}, \ldots, b_n^{(k)})$$

$$j = 1, 2, \ldots, N+2$$

A função DERRO (I,J) retorna com o J-ésimo coeficiente da I-ésima inequação do tipo acima. As derivadas são calculadas, tomando a inclinação da função F correspondente a dois pontos obtidos do ponto onde se deseja a derivada, quando a este se deu um incremento, para a direita para a esquerda, cujo valor inicial foi tomado igual a 10^{-3} do intervalo de aproximação. Repete-se o cálculo das inclinações, tomando-se, em cada vez, o incremento igual à metade do anterior, até que duas inclinações consecutivas difiram por um valor inferior a 0,01% da última inclinação calculada.

FUNÇÃO DERIV

Parâmetros:

I - Índice de linha na matriz S

J - Índice de coluna na matriz S

Esta função fornece todos os coeficientes, que dependem de cálculos de derivadas, das N linhas da matriz S que seguem suas N+2 primeiras.

Estas N linhas são formadas pelos coeficientes das inequações seguintes:

$$0\Delta x_1 + \ldots + \frac{\partial G}{\partial x_j^{(k)}} \Delta x_j + \ldots + 0\Delta x_{N+2} + \frac{\partial G}{\partial a_o^{(k)}} \Delta a_o + \ldots + \frac{\partial G}{\partial a_m^{(k)}} \Delta a_m +$$

$$+ \frac{\partial G}{\partial b_1^{(k)}} \Delta b_1 + \ldots + \frac{\partial G}{\partial b_n^{(k)}} \Delta b_n + 0\lambda - \epsilon H \leq - G(x_j^{(k)},\ a_o^{(k)}, \ldots, a_m^{(k)},$$

$$b_1^{(k)}, \ldots, b_n^{(k)})$$

$$j = 2, 3, \ldots, N+1$$

As inequações acima provêm de (2.15). A função DERIV (I,J) retorna com o J-ésimo coeficiente da (I-N-2)-ésima inequação do tipo acima. A determinação das derivadas se faz como na função DERRO. Cumpre ressaltar, apenas, que as determinações de $\frac{\partial G}{\partial a_j^{(k)}}$ e $\frac{\partial G}{\partial b_j^{(k)}}$ são feitas, lembrando as definições de F e G em (2.8) e (2.9), a partir de:

$$\frac{\partial G}{\partial a_j^{(k)}} = \frac{\partial}{\partial a_j^{(k)}} \frac{\partial F}{\partial x_j^{(k)}} = \frac{\partial}{\partial x_j^{(k)}} \frac{\partial F}{\partial a_j^{(k)}} = \frac{\partial}{\partial x_j^{(k)}} \left[\frac{(x_j^{(k)})^j}{\sum\limits_{j=1}^{n} b_j^{(k)} (x_j^{(k)})^j + 1} \right]$$

e

$$\frac{\partial G}{\partial b_j^{(k)}} = \frac{\partial}{\partial b_j^{(k)}} \frac{\partial F}{\partial x_j^{(k)}} = \frac{\partial}{\partial x_j^{(k)}} \frac{\partial F}{\partial b_j^{(k)}} = \frac{\partial}{\partial x_j^{(k)}} \left[\frac{-(x_j^{(k)})^j \sum\limits_{j=0}^{m} a_j^{(k)} (x_j^{(k)})^j}{\left[\sum\limits_{j=1}^{n} b_j^{(k)} (x_j^{(k)})^j\right]^2} \right]$$

o que é possível, já que as condições de igualdade de derivadas cruzadas são satisfeitas pela função F.

SUBROTINAS IGUAL E REGMAG

Estas subrotinas e outras chamadas por estas, encarregam-se, exclusivamente, da parte de programação linear. Foram desenvolvidas pelo grupo de Análise de Sistemas do INPE, onde têm sido, amplamente, usadas por seus pesquisadores. As listagens destas subrotinas, onde são encontrados os esclarecimentos que permitem seu uso, encontram-se, também, no apêndice I.

CAPÍTULO IV

CONCLUSÃO E EXEMPLOS

4.1 - CONCLUSÃO.

Embora não se tenha conseguido a prova da convergência do método exposto, alguns dos bons resultados obtidos são mostrados a seguir, para atribuir alguma validade ao método. Cumpre ressaltar que, quando se tomou graus maiores dos que aqueles utilizados nos exemplos que se seguirão, não se conseguiu sucesso na aproximação. Acredita-se que os insucessos sejam devidos a imprecisões numéricas, já que erros muito pequenos ocorrem quando se tenta aproximações com graus relativamente altos. Assim, êste problema deve ser defrontado por qualquer método que tente efetuar tal tipo de aproximação. Entretanto, como se verá, o método está funcionando satisfatoriamente para graus baixos, com a única restrição de que a função tenha derivada segunda contínua no intervalo.

4.2 - EXEMPLOS.

Para ilustração, foram escolhidas as seguintes cinco funções a serem aproximadas: seno no intervalo $[0,\pi/2]$, arcotangente no intervalo $[0,1]$, logaritmo natural no intervalo $[1,2]$, exponencial no inter valo [0,1], raiz quadrada no intervalo [1/2,1]. Estas funções foram escolhidas, arbitrariamente, dentre as funções intrinsecas, do tipo dupla precisão, fornecidas pelo Fortran do B-6700. Cada uma das funções acima foi aproximada por duas combinações de graus. Em cada exemplo, será encontrado, além do gráfico da função erro correspondente à melhor aproximação, a listagem dos coeficientes dos polinômios, assim como os extremantes e respectivos erros em cada iteração.

Função Logaritmo Natural

Intervalo: $[1,2]$

Grau do Numerador: 4

Grau do Denominador: 2

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

-.18215658D-01 .43251250D+00 .53789624D+01 -.25702383D+01 -.32230208D+01

COEFICIENTES DO DENOMINADOR

.28607367D+01 .55516407D+01 .10000000D+01

EXTREMANTE	ERRO
.1000000000D+01	-.1713448671D-07
.1046912031D+01	.1491733345D-07
.1180421766D+01	-.1033189085D-07
.1377711755D+01	.6371746292D-08
.1601260467D+01	-.3932884994D-08
.1805911568D+01	.2649730937D-08
.1948716853D+01	-.2058128052D-08
.2000000000D+01	.1881933657D-08

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

-.20084213D-01 .46568092D+00 .55788917D+01 -.27764527D+01 -.32460357D+01

COEFICIENTES DO DENOMINADOR

.30062348D+01 .56918074D+01 .10000000D+01

EXTREMANTE	ERRO
.1000000000D+01	-.5042465272D-08
.1033992937D+01	.5858501497D-08
.1141420543D+01	-.6987715465D-08
.1320674323D+01	.6592708910D-08
.1546981911D+01	-.6315089309D-08
.1771635100D+01	.5298258632D-08
.1938456503D+01	-.4938493871D-08
.2000000000D+01	.4482975879D-08

****

ITERACAO 2

COEFICIENTES DO NUMERADOR

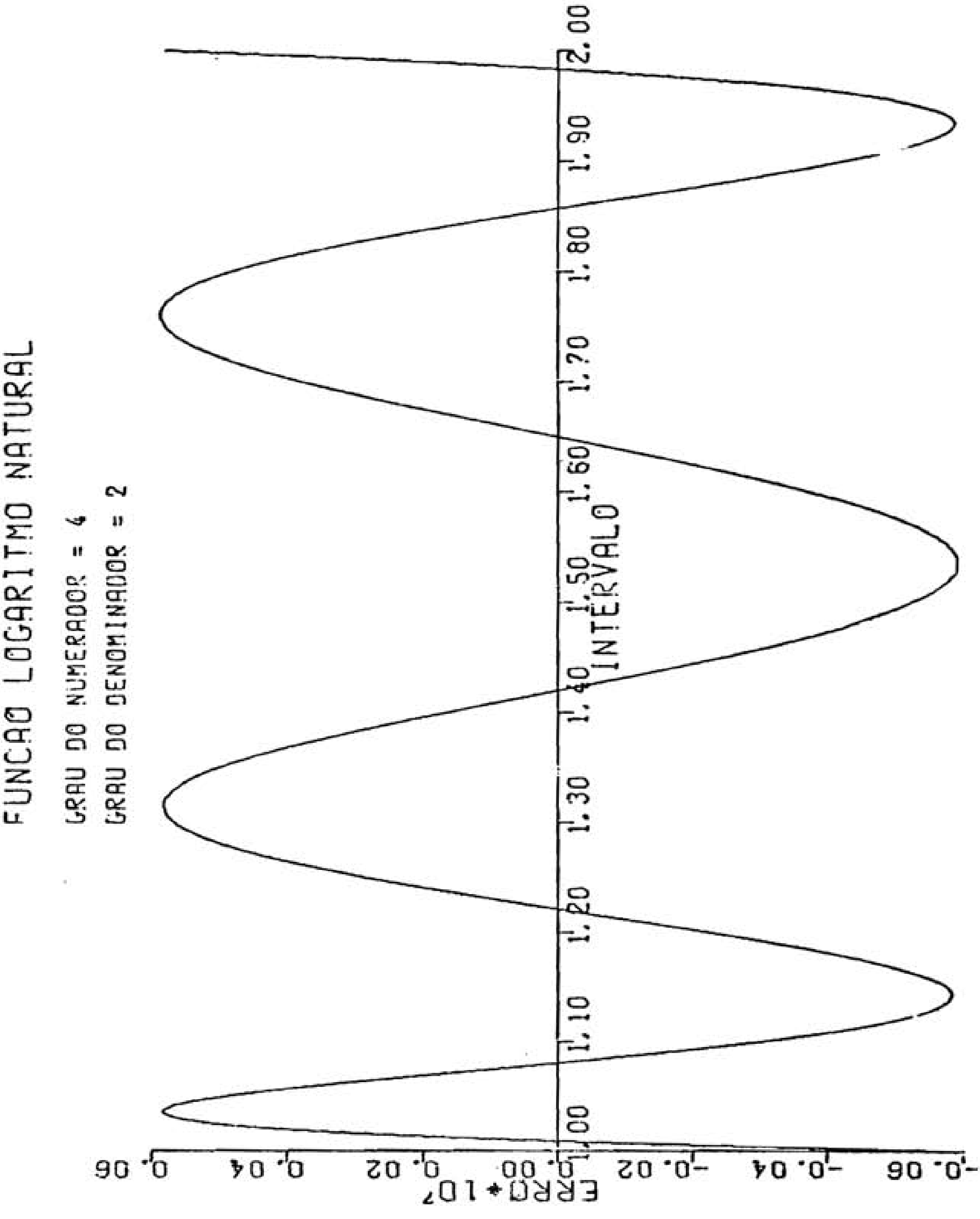
-.20325025D-01 .46973621D+00 .56009051D+01 -.27997440D+01 -.32505718D+01

COEFICIENTES DO DENOMINADOR

,302300120+01 ,570697800+01 ,100000000+01

EXTREMANTE	ERRO
,1000000000D+01	-,5815469626D-08
,1035958089D+01	,5839930448D-08
,1143020815D+01	-,5826595017D-08
,1315161536D+01	,5840893317D-08
,1534312859D+01	-,5911584894D-08
,1760543517D+01	,5888229659D-08
,1934354271D+01	-,5658527667D-08
,2000000000D+01	,5798440204D-08

****


FUNCAO LOGARITMO NATURAL
GRAU DO NUMERADOR = 4
GRAU DO DENOMINADOR = 2
1.00
1.10
1.20
1.30
1.40
1.50
1.60
1.70
1.80
1.90
2.00
INTERVALO
-0.06
-0.04
-0.02
0.00
0.02
0.04
0.06
ERRO*10⁷

Função Logaritmo Natural

Intervalo: $[1,2]$

Grau do Numerador: 2

Grau do Denominador: 1

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.23364630D+00 .19060211D+01 -.21394470D+01

COEFICIENTES DO DENOMINADOR

.13806136D+01 .10000000D+01

EXTREMANTE	ERRO
.10000000000D+01	-.92612054630D-04
.11350550000D+01	.73836138900D-04
.14403868810D+01	-.45437966560D-04
.18449173080D+01	.29969932040D-04
.20000000000D+01	-.25637818310D-04

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

.24656914D+00 .19202919D+01 -.21667461D+01

COEFICIENTES DO DENOMINADOR

.14167365D+01 .10000000D+01

EXTREMANTE	ERRO
.10000000000D+01	-.47535188100D-04
.11082026330D+01	.50641914590D-04
.14220013620D+01	-.52165622320D-04
.18140092930D+01	.48587844970D-04
.20000000000D+01	-.46393515690D-04

****

ITERACAO 2

COEFICIENTES DO NUMERADOR

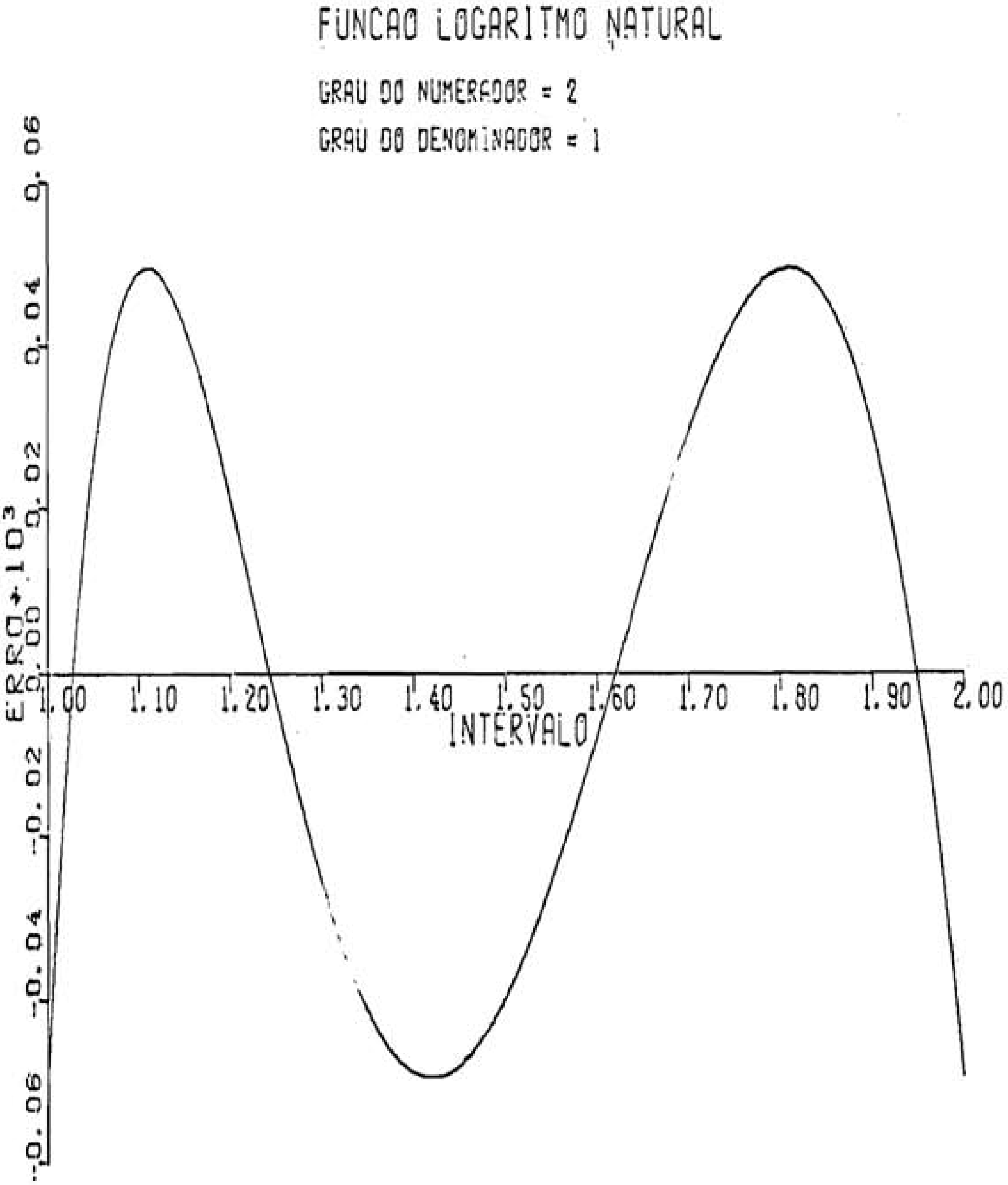
.24702613D+00 .19204998D+01 -.21670059D+01

COEFICIENTES DO DENOMINADOR

.14198720D+01 .10000000D+01

EXTREMANTE	ERRO
.1000000000D+01	-.4957870420D-04
.1109346245D+01	.4958700133D-04
.1419628741D+01	-.4958881477D-04
.1810108787D+01	.4961092040D-04
.2000000000D+01	-.4957780857D-04

****


FUNCAO LOGARITMO NATURAL
GRAU DO NUMERADOR = 2
GRAU DO DENOMINADOR = 1
ERRO*10³
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
1.00
1.10
1.20
1.30
1.40
1.50
1.60
1.70
1.80
1.90
2.00
INTERVALO

Função Raiz Quadrada

Intervalo $[0.5,1]$

Grau do Numerador: 2

Grau do Denominador: 1

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.399033960D+00 .176151110D+01 .211803340D+00

COEFICIENTES DO DENOMINADOR

.137243200D+01 .100000000D+01

EXTREMANTE	ERRO
.50000000000D+00	-.19339921830D-04
.56869595300D+00	.16173284120D-04
.74210375990D+00	-.10987435140D-04
.92326046300D+00	.78470044290D-05
.10000000000D+01	-.69067021000D-05

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

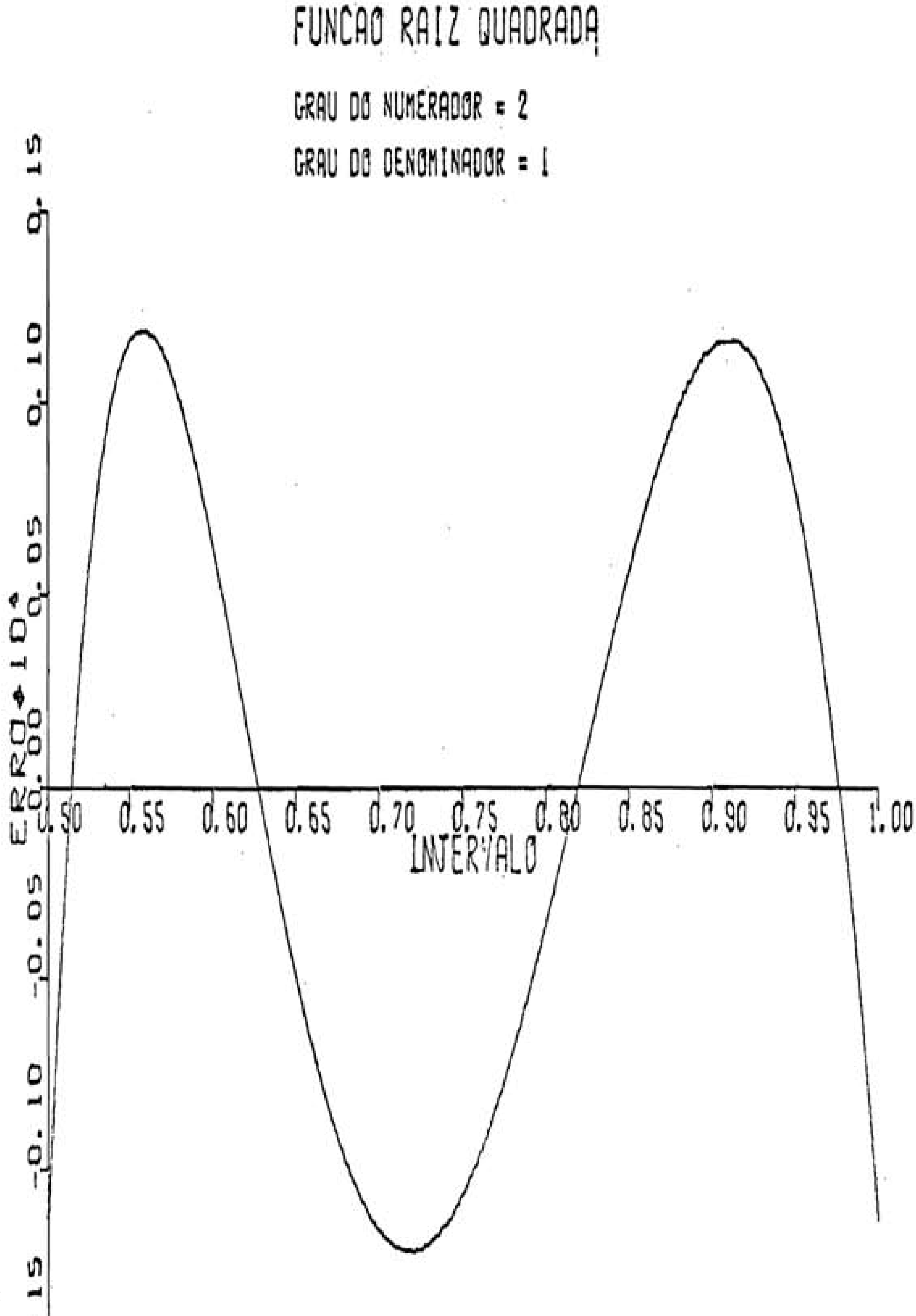
.412354620D+00 .178121850D+01 .209468970D+00

COEFICIENTES DO DENOMINADOR

.140301500D+01 .100000000D+01

EXTREMANTE	ERRO
.50000000000D+00	-.11374782710D-04
.55751211300D+00	.11861747680D-04
.71836004450D+00	-.12140544070D-04
.91090993470D+00	.11602030150D-04
.10000000000D+01	-.11246754190D-04

****


FUNCAO RAIZ QUADRADA
GRAU DO NUMERADOR = 2
GRAU DO DENOMINADOR = 1
ERRO*10
0.15
0.10
0.05
0.00
-0.05
-0.10
-0.15
0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00
INTERVALO

Função Raiz Quadrada

Intervalo: [0.5,1]

Grau do Numerador: 4

Grau do Denominador: 1

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.462991980D-01 -.321459380D+00 .177308970D+01 .257743670D+01 .154244990D+00

COEFICIENTES DO DENOMINADOR

.322961110D+01 .100000000D+01

EXTREMANTE	ERRO
.50000000000D+00	-.11801575940D-06
.53204986650D+00	.10519460150D-06
.62107894260D+00	-.78510394400D-07
.74536225260D+00	.54954974180D-07
.87166947130D+00	-.40329524550D-07
.96553336800D+00	.32391999400D-07
.10000000000D+01	-.30991191040D-07

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

.494584270D-01 -.337143340D+00 .182490230D+01 .260242210D+01 .152761550D+00

COEFICIENTES DO DENOMINADOR

.324240080D+01 .100000000D+01

EXTREMANTE	ERRO
.50000000000D+00	-.57136979050D-07
.52598941370D+00	.60579765310D-07
.60355489000D+00	-.65411927230D-07
.72294875480D+00	.65109730790D-07
.85565255990D+00	-.61273093540D-07
.96025993580D+00	.57406371690D-07
.10000000000D+01	-.56053083630D-07

****

ITERACAO 2

COEFICIENTES DO NUMERADOR

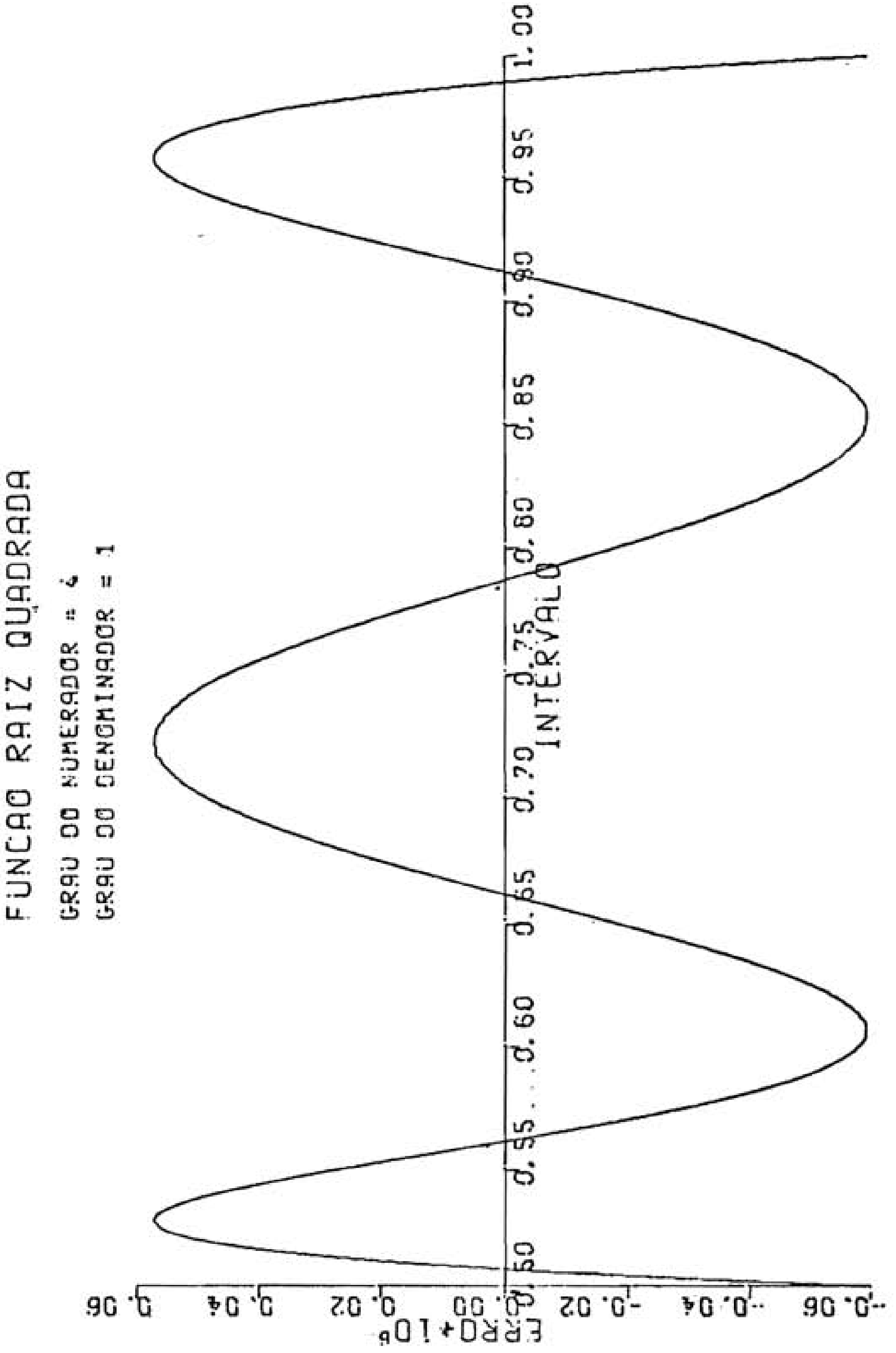
.496004890D-01 -.337779820D+00 .182677520D+01 .260321470D+01 .152720940D+00

COEFICIENTES DO DENOMINADOR

.329453120+01 .100000000+01

EXTREMANTE	ERRO
.5000000000D+00	-.6105254465D-07
.5266404346D+00	.6109870441D-07
.6041070970D+00	-.6106537092D-07
.7215267076D+00	.6109010796D-07
.8532122724D+00	-.6116922796D-07
.9591496424D+00	.6111079440D-07
.1000000000D+01	-.6105268989D-07

****


FUNCAO RAIZ QUADRADA
GRAU DO NUMERADOR = 4
GRAU DO DENOMINADOR = 1
INTERVALO
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00
-0.06
-0.04
-0.02
0.00
0.02
0.04
0.06

Função Exponencial

Intervalo: $[0,1]$

Grau do Numerador: 2

Grau do Denominador: 1

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.234842100+00 .712346190+00 .100012530+01

COEFICIENTES DO DENOMINADOR

-.283692120+00 .100000000+01

EXTREMANTE	ERRO
.12242273070-21	-.12529080590-03
.15189348000+00	.13851667330-03
.51140262320+00	-.17816488790-03
.85955306900+00	.23254269110-03
.10000000000+01	-.26090064180-03

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

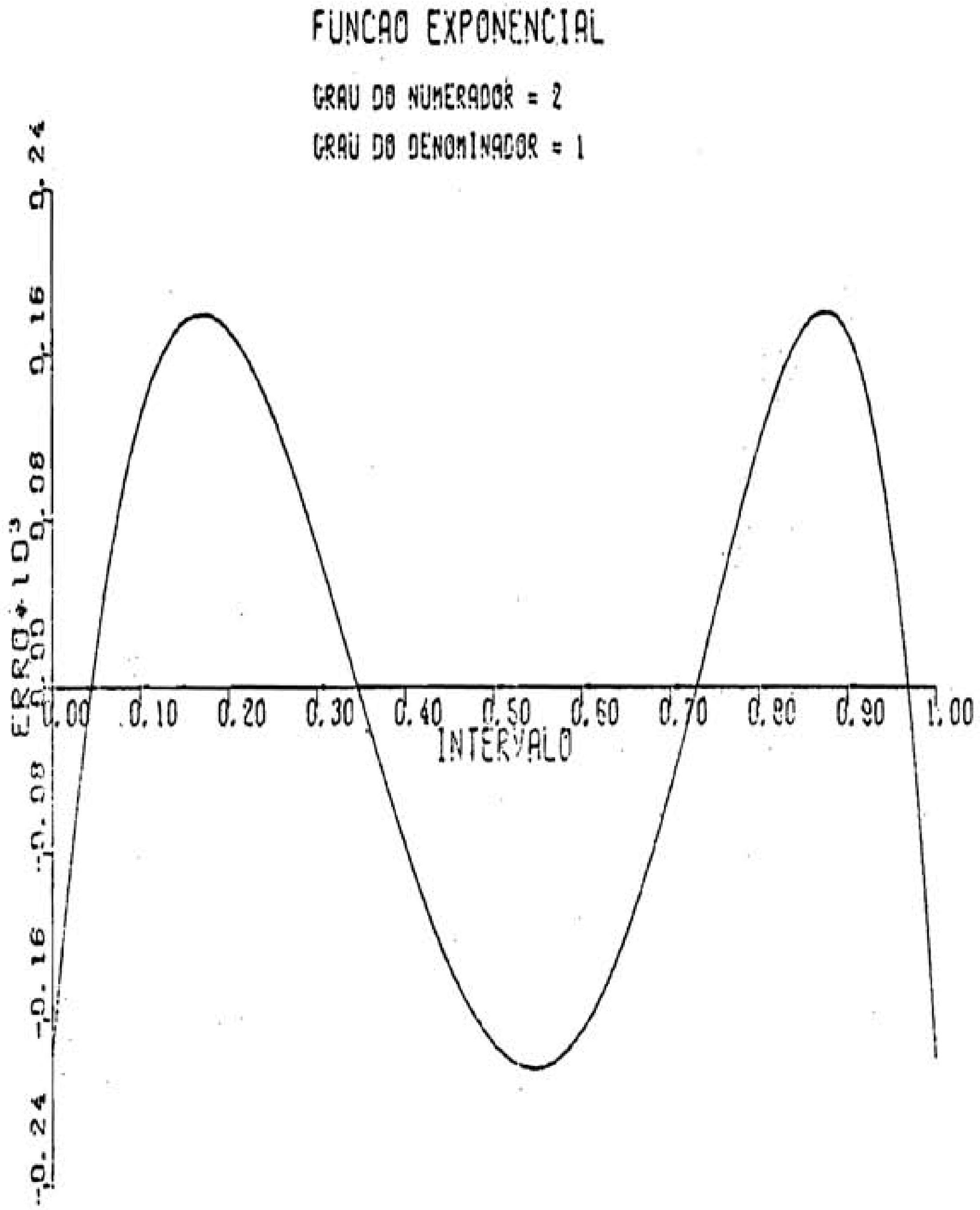
.238692300+00 .713274040+00 .100017680+01

COEFICIENTES DO DENOMINADOR

-.281893430+00 .100000000+01

EXTREMANTE	ERRO
.12242273070-21	-.17684106900-03
.16967489440+00	.17979236390-03
.54526676760+00	-.18317536720-03
.87564545620+00	.18065012820-03
.10000000000+01	-.17705493610-03

****


FUNCAO EXPONENCIAL
GRAU DO NUMERADOR = 2
GRAU DO DENOMINADOR = 1
ERRO * 10³
0.24
0.16
0.08
0.00
-0.08
-0.16
-0.24
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
INTERVALO

Função Exponencial

Intervalo $[0,1]$

Grau do Numerador: 2

Grau do Denominador: 2

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.10755996D+00 .54115857D+00 .10000027D+01

COEFICIENTES DO DENOMINADOR

.65238417D-01 -.45870610D+00 .10000000D+01

EXTREMANTE	ERRO
.4632211430D-22	-.2732913137D-05
.9385542842D-01	.2996157275D-05
.3544318647D+00	-.3825887514D-05
.6636142671D+00	.5212622163D-05
.9080361396D+00	-.6730341724D-05
.1000000000D+01	.7428807818D-05

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

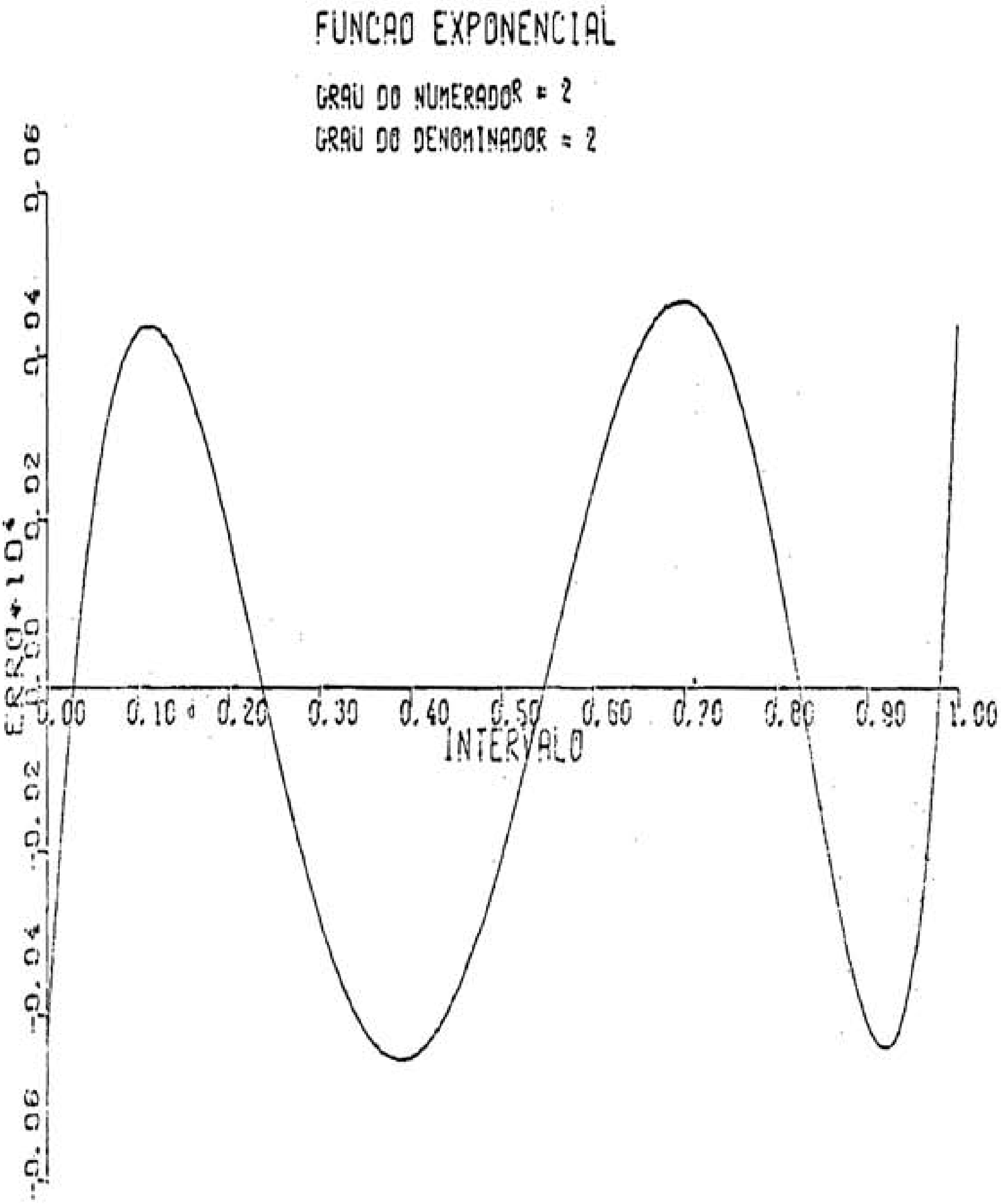
.10899704D+00 .54307455D+00 .10000043D+01

COEFICIENTES DO DENOMINADOR

.64509691D-01 -.45674397D+00 .10000000D+01

EXTREMANTE	ERRO
.4632211430D-22	-.4299091738D-05
.1129239935D+00	.4397167049D-05
.3702909151D+00	-.4594840273D-05
.6985532555D+00	.4636283321D-05
.9212210126D+00	-.4451745157D-05
.1000000000D+01	.4301693953D-05

****


FUNCAO EXPONENCIAL
GRAU DO NUMERADOR = 2
GRAU DO DENOMINADOR = 2
ERRO*10⁴
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
INTERVALO

Função Seno

Intervalo: $[0,\pi/2]$

Grau do Numerador: 3

Grau do Denominador: 3

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

-.14557198D+00 .12528950D+00 .99992334D+00 .12311026D-05

COEFICIENTES DO DENOMINADOR

.15695961D-01 .23988184D-01 .12451624D+00 .10000000D+01

EXTREMANTE	ERRO
.2977850205D-22	-.1231102599D-05
.7774167750D-01	.1230032433D-05
.2953735259D+00	-.1222137272D-05
.6095598240D+00	.1196908103D-05
.9584191850D+00	-.1148806703D-05
.1273545621D+01	.1087960526D-05
.1492451147D+01	-.1037205345D-05
.1570796327D+01	.1017600417D-05

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

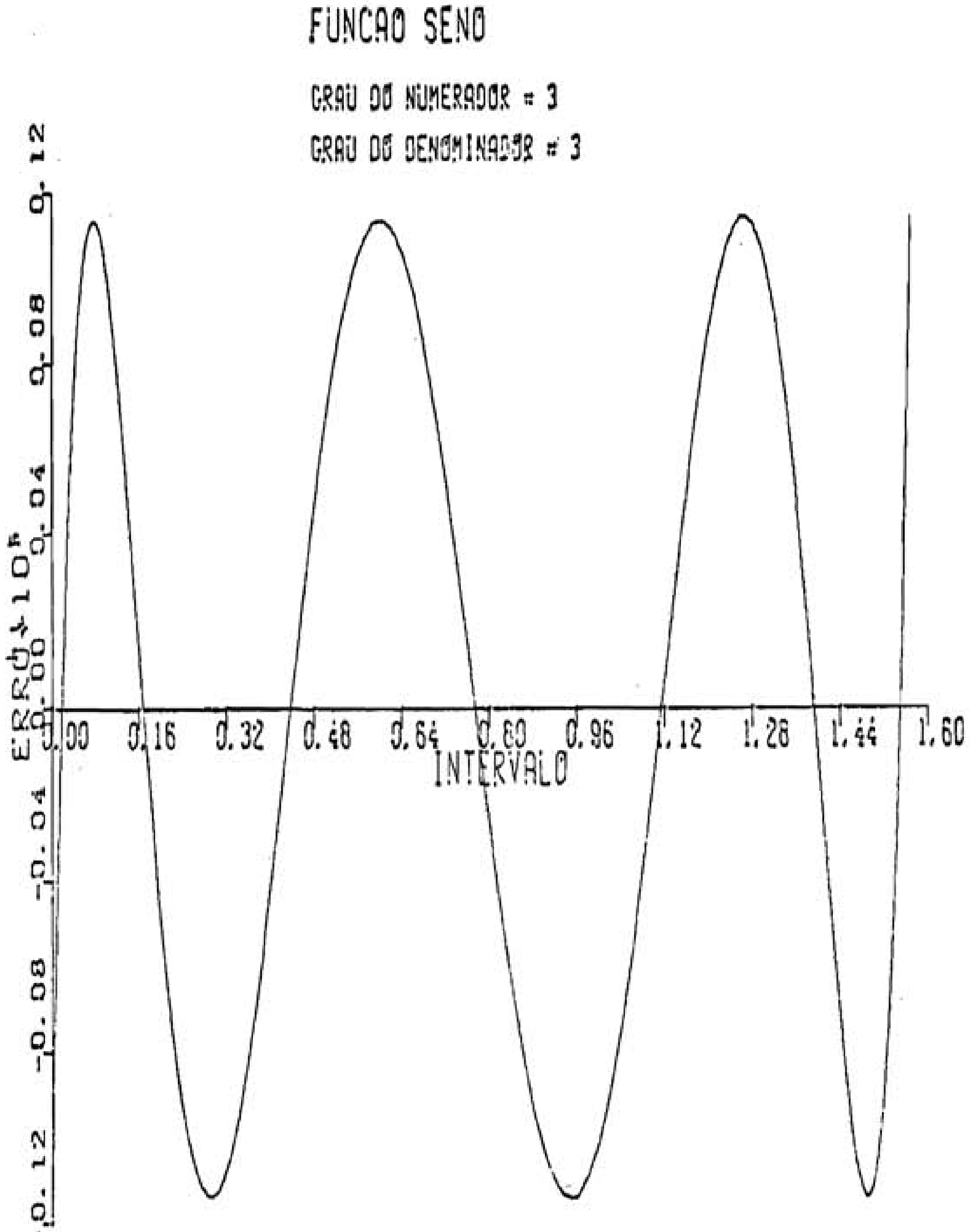
-.14459441D+00 .12306744D+00 -.99992832D+00 .11351766D-05

COEFICIENTES DO DENOMINADOR

.15489898D-01 .24564031D-01 .12233480D+00 .10000000D+01

EXTREMANTE	ERRO
.2977850205D-22	-.1135176602D-05
.7666750034D-01	.1135443320D-05
.2912935866D+00	-.1136377010D-05
.6018268085D+00	.1138139558D-05
.9490195541D+00	-.1138438533D-05
.1266513624D+01	.1140077921D-05
.1490024135D+01	-.1134238499D-05
.1570796327D+01	.1136708765D-05

****


FUNCAO SENO
GRAU DO NUMERADOR = 3
GRAU DO DENOMINADOR = 3
ERRO * 10
INTERVALO
0.00
0.16
0.32
0.48
0.64
0.80
0.96
1.12
1.28
1.44
1.60
0.12
0.08
0.04
0.00
-0.04
-0.08
-0.12

Função Seno

Intervalo: $[0,\pi/2]$

Grau do Numerador: 4

Grau do Denominador: 2

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.18247209D-01 -.12820653D+00 -.98132462D-01 .99997463D+00 .40614137D-06

COEFICIENTES DO DENOMINADOR

.39461983D-01 -.98392202D-01 .10000000D+01

EXTREMANTE	ERRO
.2977850205D-22	-.4061413750D-06
.7783039063D-01	.4070534335D-06
.2956263415D+00	-.4074894235D-06
.6099134839D+00	.4029915567D-06
.9587672007D+00	-.3907594752D-06
.1273776111D+01	.3735156393D-06
.1492523777D+01	-.3584439714D-06
.1570796327D+01	.3525119871D-06

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

.18196400D-01 -.12811148D+00 -.97980815D-01 .99997552D+00 .38933275D-06

COEFICIENTES DO DENOMINADOR

.39537109D-01 -.98233490D-01 .10000000D+01

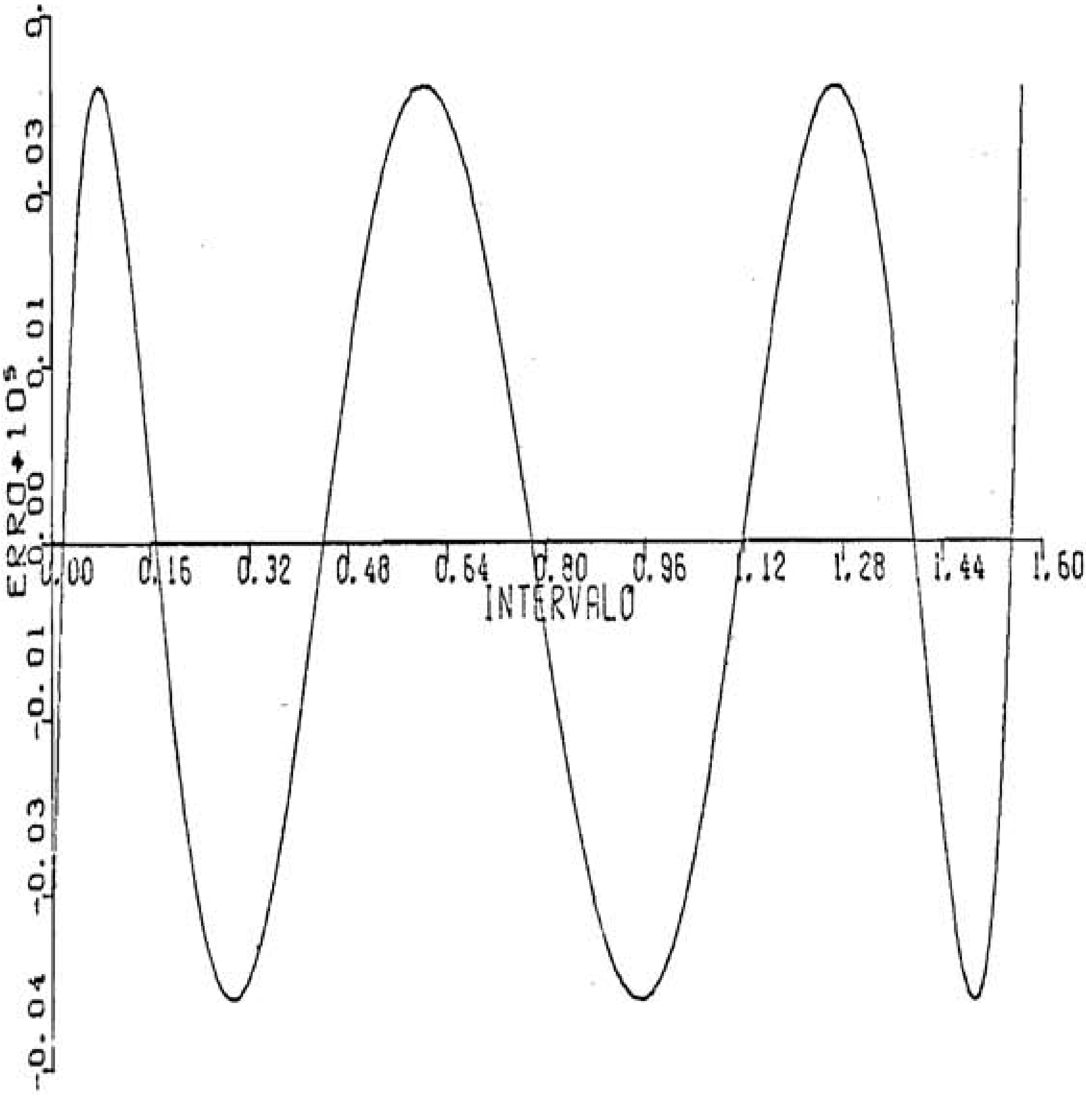
EXTREMANTE	ERRO
.2977850205D-22	-.3893327492D-06
.7721270124D-01	.3893545485D-06
.2929784286D+00	-.3895154363D-06
.6043275589D+00	.3898351985D-06
.9515283201D+00	-.3903115320D-06
.1268060407D+01	.3894758596D-06
.1490548886D+01	-.3906219453D-06
.1570796327D+01	.3880328291D-06

****

FUNCAO SENO

GRAU DO NUMERADOR = 4

GRAU DO DENOMINADOR = 2



Função Arcotangente

Intervalo: [0,1]

Grau do Numerador: 2

Grau do Denominador: 3

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.13087792D+00 .99886341D+00 .15735509D-04

COEFICIENTES DO DENOMINADOR

-.70753927D-01 .39178443D+00 .11741226D+00 .10000000D+01

EXTREMANTE	ERRO
.3970466940D-22	-.1573550944D-04
.6610100000D-01	.1528858986D-04
.2452524903D+00	-.1339921606D-04
.4908607122D+00	.1009266521D-04
.7419040003D+00	-.7022771513D-05
.9301592225D+00	.5262850019D-05
.1000000000D+01	-.4698204640D-05

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

.85626174D-01 .99429959D+00 .93815643D-05

COEFICIENTES DO DENOMINADOR

-.77462221D-01 .38363941D+00 .75076207D-01 .10000000D+01

EXTREMANTE	ERRO
.3970466940D-22	-.9331564306D-05
.5762135959D-01	.9615995251D-05
.2176712947D+00	-.1021034939D-04
.4590329689D+00	.1066604436D-04
.7073671152D+00	-.1045917214D-04
.9176261558D+00	.9872182769D-05
.1000000000D+01	-.9571544255D-05

****

ITERACAO 2

COEFICIENTES DO NUMERADOR

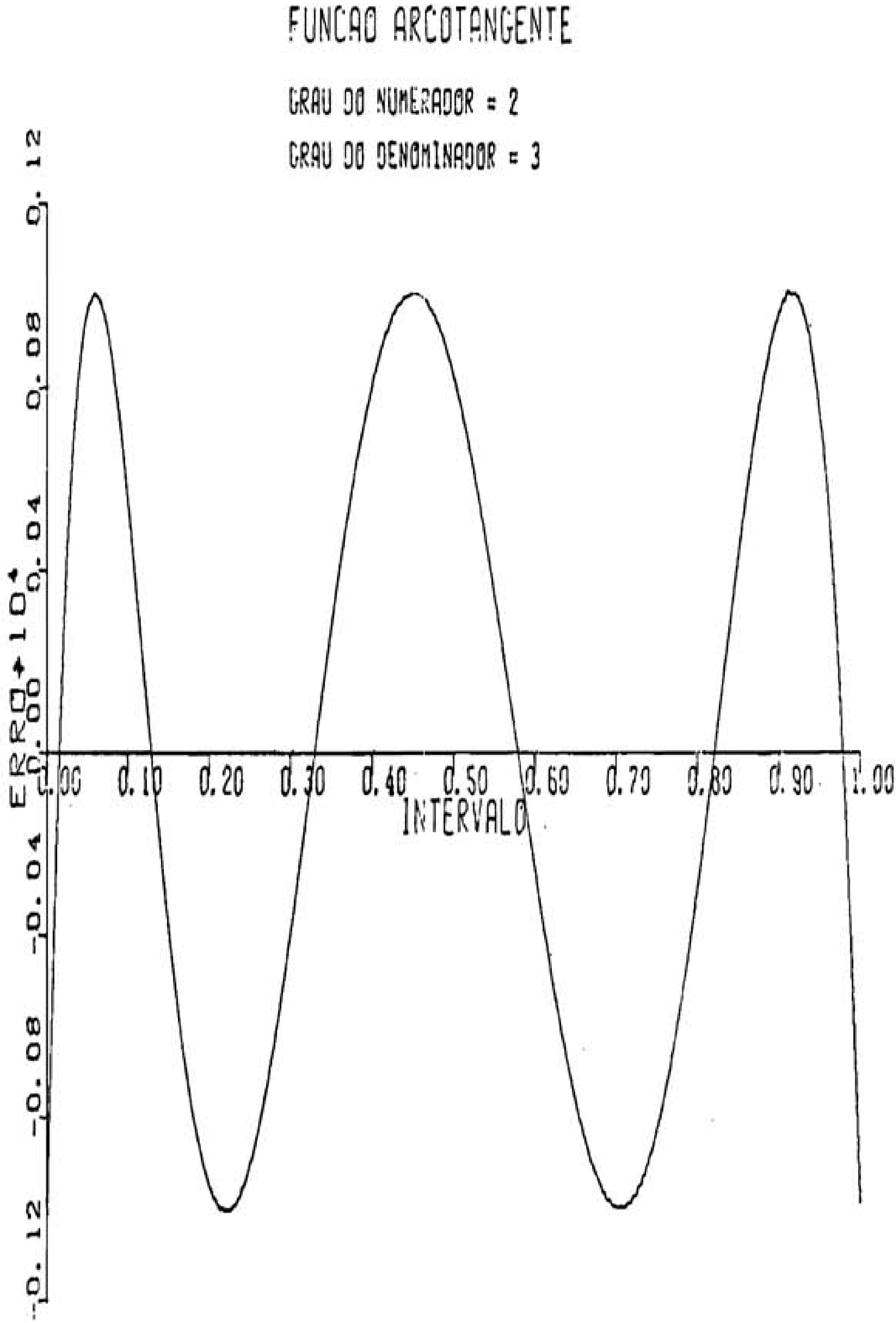
.86961517D-01 .99918168D+00 .10031409D-04

COEFICIENTES DO DENOMINADOR

-.77353389D-01 .38404421D+00 .76224759D-01 .10000000D+01

EXTREMANTE	ERRO
.3970446940D-22	-.1003160970D-04
.5897888094D-01	.1003873942D-04
.2207589414D+00	-.1004197833D-04
.4511805422D+00	.1003318247D-04
.7054825069D+00	-.1003512150D-04
.9160960495D+00	.1003610225D-04
.1000000000D+01	-.1003137923D-04

****


FUNCAO ARCOTANGENTE
GRAU DO NUMERADOR = 2
GRAU DO DENOMINADOR = 3
ERRO * 10⁴
0.12
0.08
0.04
0.00
-0.04
-0.08
-0.12
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
INTERVALO

Função Arcotangente

Intervalo: $[0,1]$

Grau do Numerador: 2

Grau do Denominador: 2

APROXIMACAO INICIAL

COEFICIENTES DO NUMERADOR

.63544890D+00 .99395061D+00 .11962745D-03

COEFICIENTES DO DENOMINADOR

.49055887D+00 .58432497D+00 .10000000D+01

EXTREMANTE	ERRO
.4632211430D-22	-.1196274453D-03
.9098724022D-01	.1064198859D-03
.3300667802D+00	-.7245648080D-04
.6363430193D+00	.4034031445D-04
.8971709975D+00	-.2403345472D-04
.1000000000D+01	.1967197476D-04

****

ITERACAO 1

COEFICIENTES DO NUMERADOR

.57388681D+00 .99655520D+00 .48512160D-04

COEFICIENTES DO DENOMINADOR

.46197511D+00 .53776182D+00 .10000000D+01

EXTREMANTE	ERRO
.4632211430D-22	-.4851215977D-04
.7023967519D-01	.5339113427D-04
.2730408871D+00	-.6096779180D-04
.5708724045D+00	.6351974145D-04
.8681059170D+00	-.5341261267D-04
.1000000000D+01	.4959754165D-04

****

ITERACAO 2

COEFICIENTES DO NUMERADOR

.57424917D+00 .99639365D+00 .55344172D-04

COEFICIENTES DO DENOMINADOR

.46242734D+00 .53758860D+00 .10000000D+01

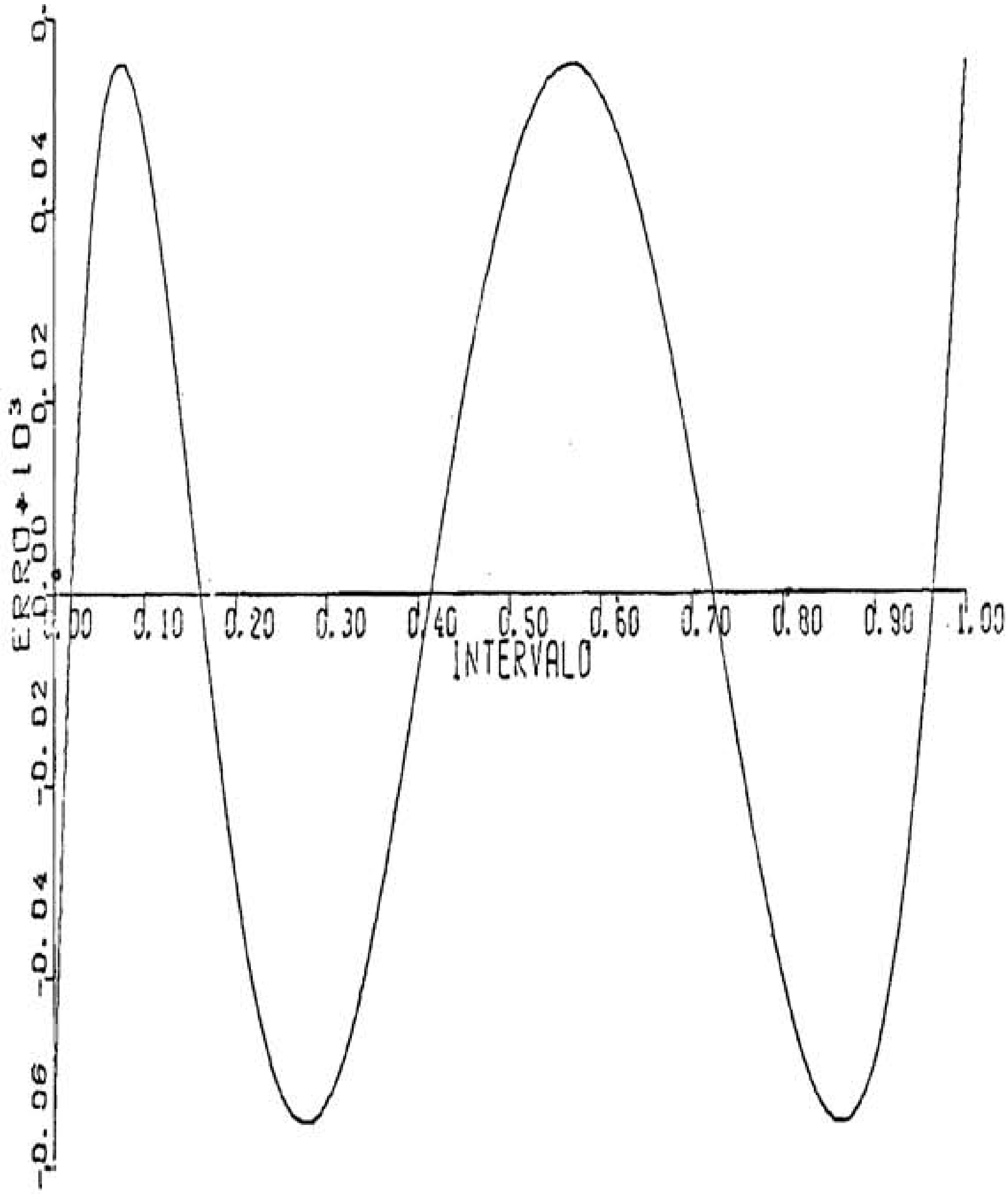
EXTREMANTE	ERRO
.4632211430D-22	-.5534417196D-04
.7347266578D-01	.5548559381D-04
.2764233762D+00	-.5539232787D-04
.5669100961D+00	.5538801425D-04
.8631210697D+00	-.5545469120D-04
.1000000000D+01	.5534336347D-04

****

FUNCAO ARCOTANGENTE

GRAU DO NUMERADOR = 2

GRAU DO DENOMINADOR = 2



BIBLIOGRAFIA

1. ACHIESER, N.I. - Theory of Approximation.
   Frederick Ungar Co., New York (1956).

2. BURNSIDE, W.S. e PANTON, A.W. - The Theory of Equations with an Introduction to the Theory of Binary Algebraic Forms. Volume II. Dover, New York, (1960).

3. CHENEY, E.W. - Introduction to Approximation Theory.
   McGraw-Hill, New York (1966).

4. CODY, W.J., FRASER, W. e HART, J.F. - Rational Chebyshev Approximation Using Linear Equations.
   Numerische Mathematik, 12, (1968), pag. 242 - 251.

5. COLLATZ, L. - Functional Analysis and Numerical Mathematics.
   Academic Press, New York (1966).

6. FRASER, W. e HART, J.F. - On the Computation of Rational Approximation to Continuous Functions.
   Communications of the ACM 7, 5 (julho de 1962), pag. 401 - 403.

7. GOLUB, G.H. e SMITH, L.B. - Chebyshev Approximation of Continuous Functions by a Chebyshev System of Functions. Collected Algorithms from CACM, Algorithm 414, (1971).

8. ISHIZAKI, Y. e WATANABE, H. - An Iterative Chebyshev Approximation Method for Network Design.
   IEEE Transactions on Circuit Theory, 15, (1968), pag. 326 - 336.

9. LORENTZ, G.G. - Approximation of Functions.
Holt, Rinehart and Winston, New York (1966).

10. MEINARDUS, G. - Approximation of Functions: Theory and Numerical Methods. Springer- Verlag, New York (1967).

11. RALSTON, A. - Rational Chebyshev Approximation by Remes' Algorithms.
Numerische Mathematik, 7, (1965).

12. RALSTON, A. - Rational Chebyshev Approximation.
Em Mathematical Methods for Digital Computers, Vol. II. A. Ralston e H.S. Wilf (editores), Wiley, New York (1967).

13. SNYDER, M.A. - Chebyshev Methods in Numerical Approximation.
Prentice-Hall, Englewood Cliffs, N.J. (1966).

14. Vlach, J. - Computerized Approximation and Synthesis of Linear Networks.
Wiley, New York (1966).

15. WATSON, G.A. - On an Algorithm for Nonlinear Minimax Approximation.
Communications of the ACM 3, 13 (março de 1970), pag. 160 - 162.

16. WERNER, H., STOER, J. e BOMMAS, W. - Rational Chebyshev Approximation.
Numerische Mathematik, 10, (1967), pag. 289 - 306.

17. WILDE, D.J. - Optimum Seeking Methods.
Englewood Cliffs, N.J. (1964).

APÊNDICE I

```
      SUBROUTINE RAPROX (M,N,A0,B0,IERRO,ISAIDA)
      DOUBLE PRECISION AA(20,20),TI(20),S(70,30),TS(70),DELTA(30),
     1 Y(60),DD(60),HG(30),HF(30),XT,D,E
      DOUBLE PRECISION A(10),B(10),X(20)
      COMMON /COEF/A,B,X
      COMMON //MM,NN,AA0,BB0
      DOUBLE PRECISION Z,DE
      DE=1.0E-8*(B0-A0)
      NTOT=M+N+2
      NI=2*NTOT-1
      NE=5*(NTOT-1)
      MM=M
      NN=N
      AA0=A0
      BB0=B0
      DEL=(B0-A0)/(10.0*FLOAT(NTOT))
C
C
C          INICIALIZACAO DO ARRAY DE EXTREMANTES
C
      C1=(A0+B0)/2.0
      C2=(B0-A0)/2.0
  499 XT=FLOAT(NTOT-1)
      DO 500 I=1,NTOT
      XI=FLOAT(NTOT-I)
  500 X(I)=C1+C2*DCOS(XI*3.1415926536/XT)
C
C          INICIALIZACAO DOS COEFICIENTES
C
      XT=2.0*FLOAT(NTOT-1)
      DO 520 I=1,NTOT-1
      XI=FLOAT(2*I-1)
  520 Y(I)=C1+C2*DCOS(XI*3.1415926536/XT)
      DO 530 I=1,NTOT-1
      CALL FUNCAO (Y(I),TI(I))
      AA(I,M+1)=1.0
      IF (M.EQ.0) GO TO 545
      DO 540 J=1,M
  540 AA(I,M+1-J)=AA(I,M+2-J)*Y(I)
  545 IF (N.EQ.0) GO TO 530
      AA(I,M+N+1)=-TI(I)*Y(I)
      IF (N.EQ.1) GO TO 530
      DO 535 J=1,N-1
  535 AA(I,M+N+1-J)=AA(I,M+N+2-J)*Y(I)
  530 CONTINUE
      CALL RESEL (NTOT-1,AA,TI,DD,IERRO,20)
      IF (IERRO.EQ.3) GO TO 950
      DO 550 I=1,M+1
  550 A(I)=DD(I)
      DO 551 I=1,N
  551 B(I)=DD(I+M+1)
      B(N+1)=1.0
      IF (ISAIDA.LT.2) GO TO 2610
      WRITE (6,119)
  119 FORMAT (57X,19HAPROXIMACAO INICIAL)
      WRITE (6,120) (A(I),I=1,M+1)
      WRITE (6,121) (B(I),I=1,N+1)
 2610 CALL STURM (B,N,A0,B0,KK)
      IF (KK.GT.0) IERRO=6
      CALL NEWMAX (1.0E-8,DE,100,NTOT)
C
```

[redacted]

```
C   PREPARACAO PARA O SIMPLEX
C
      IF (ISAIDA.LT.2) GO TO 2620
      WRITE (6,118)
  118 FORMAT (//,49X,10HEXTREMANTE,20X,4HERRO,/)
      DO 560 I=1,NTOT
      CALL ERRO (X(I),HF(I))
  560 WRITE (6,6) X(I),HF(I)
      WRITE (6,116)
C
C
 2620 ITERA=0
   25 ITERA=ITERA+1
      IF (ITERA.GT.20) IERRO=4
      IF (IERRO.EQ.4) GO TO 950
      XE=-1.0
      DO 30 I=1,NI-1
      XE=-XE
      S(I,NI)=1.0
      IF (I.GE.NTOT+1) GO TO 27
      CALL ERRO (X(I),E)
      TS(I)=-E
      GO TO 26
   27 TS(I)=-DERRO(I-NTOT+1,I-NTOT)
      S(I,NI)=0.1E-5
      S(I,NI-1)=0.0
      GO TO 29
   26 S(I,NI-1)=XE
   29 DO 30 J=1,NI-2
      IF (I.GE.NTOT+1) GO TO 28
      S(I,J)=DERRO(I,J)
      GO TO 30
   28 S(I,J)=DERIV(I,J)
   30 CONTINUE
      DO 33 I=NI,2*NI-2
      S(I,NI)=S(I-NI+1,NI)
      TS(I)=-TS(I-NI+1)
      DO 33 J=1,NI-1
   33 S(I,J)=-S(I-NI+1,J)
      DO 34 I=2*NI-1,NE
      IJ=I-2*NI+2
      TS(I)=X(IJ)-X(IJ+1)+DEL
      DO 39 J=1,NI
   39 S(I,J)=0.0
      IF (IJ.EQ.1) GO TO 37
      IF (IJ.EQ.NTOT-1) GO TO 38
      S(I,IJ-1)=-1.0
      S(I,IJ)=1.0
      GO TO 34
   37 S(I,1)=1.0
      GO TO 34
   38 S(I,IJ-1)=-1.0
   34 CONTINUE
      IF (ISAIDA.LT.3) GO TO 2520
      WRITE (6,2510) ((S(I,J),J=1,NI),I=1,NE)
 2510 FORMAT (//,30X,48HMATRIZ DOS COEFICIENTES DO SISTEMA DE INEQUACOES
     1,//,(1X,9E14.6))
      WRITE (6,2515) (TS(I),I=1,NE)
 2515 FORMAT (//,30X,54HVETOR DE TERMOS INDEPENDENTES DO SISTEMA DE INEQ
     1UACOES,//,(1X,9E14.6))
 2520 CONTINUE
```

```
C
C   MATRIZES UTILIZADAS PELO SIMPLEX
C
      DIMENSION INDVB(65),LOCVAR(100),INDTIP(60)
      DOUBLE PRECISION CC(100),AT(70,70),P(70),XX(100),
     1 ATR(70,100),C(40)
C
C
      DO 35 I=1,NI
   35 C(I)=0.0
      C(NI)=1.0
      CALL IGUAL (S,ATR,C,TS,INDTIP,CC,70,70,NE,NI,II,NVAR,1,0)
      CALL REGMAG(ATR,TS,CC,AT,INDVB,LOCVAR,P,1,0,1E-15,Z,ITER,
     1KTIPO,70,70,100,0,3,II,NVAR,SS)
      IF (KTIPO.EQ.2) GO TO 940
      DO 50 I=1,NVAR
   50 XX(I)=0.0
      DO 60 I=1,II
      IF (INDVB(I).GT.NVAR) GO TO 60
      XX(INDVB(I))=AT(I,II+1)
   60 CONTINUE
      IJ=0
      DO 70 I=1,NI
      IJ=IJ+1
      IF (INDTIP(I).EQ.1) DELTA(I)=XX(IJ)
      IF (INDTIP(I).EQ.-1) DELTA(I)=-XX(IJ)
      IF (INDTIP(I).NE.0) GO TO 70
      DELTA(I)=XX(IJ)-XX(IJ+1)
      IJ=IJ+1
   70 CONTINUE
      E=DELTA(NI-1)
C
C   O VETOR DELTA CONTEM A SOLUCAO OBTIDA DO SIMPLEX
C
      DO 79 I=1,N
   79 B(I)=B(I)+DELTA(NTOT+M-1+I)
      DO 81 I=1,M+1
   81 A(I)=A(I)+DELTA (I+M+N)
      IF (ISAIDA.LT.2) GO TO 2530
      WRITE (6,117) ITERA
  117 FORMAT (//,57X,8HITERACAO,I3)
      WRITE (6,120) (A(I),I=1,M+1)
      WRITE (6,121) (B(I),I=1,N+1)
 2530 DO 82 I=2,NTOT-1
   82 X(I)=X(I)+DELTA(I-1)
      CALL STURM (B,N,A0,B0,KK)
      IF (KK.GT.0) IERRO=6
      IF (IERRO.EQ.6) GO TO 950
      CALL NEWMAX (1.0E-8,DE,100,NTOT)
      JJ=0
      IF (ISAIDA.LT.2) GO TO 2550
      WRITE (6,118)
 2550 DO 80 I=1,NTOT
      CALL ERRO (X(I),HF(I))
      HG(I)=DERRO(I,I-1)
      IF (ISAIDA.LT.2) GO TO 80
      WRITE (6,6) X(I),HF(I)
   80 CONTINUE
    6 FORMAT (45X,D17.10,10X,D17.10)
      IF (ISAIDA.LT.2) GO TO 2540
      WRITE (6,116)
```

```
 116 FORMAT (/,63X,10H**********)
2540 DO 85 I=1,NTOT-1
     IF (HF(I)*HF(I+1).LE.0.0) JJ=JJ+1
  85 CONTINUE
     IF (JJ.LT.NTOT-1) GO TO 25
     D=DELMAX(HF,NTOT,1)
     E=DELMAX(HF,NTOT,0)
     FAT=1.1
     IF (D*FAT.GE.E) GO TO 910
     D=DELMAX(DELTA,NI-2,0)
     IF (D.LE.1.0E-8) GO TO 920
     GO TO 25
 940 IERRO=5
 950 IF (ISAIDA.LT.1) GO TO 1000
     WRITE (6,110) IERRO
     GO TO 1000
 910 IERRO=1
     IF (ISAIDA.LT.1) GO TO 1000
     WRITE (6,110) IERRO
 110 FORMAT (1H1,9X,6HIERRO=,I2)
     GO TO 1010
 920 IERRO=2
     IF (ISAIDA.LT.1) GO TO 1000
     WRITE(6,110) IERRO
1010 WRITE (6,120) (A(I),I=1,M+1)
     WRITE (6,121) (B(I),I=1,N+1)
 121 FORMAT (//,30X,27HCOEFICIENTES DO DENOMINADOR,(//,20X,4D25.18))
 120 FORMAT (//,30X,25HCOEFICIENTES DO NUMERADOR,(//,20X,4D25.18))
     WRITE (6,130) ITERA
 130 FORMAT (/,9X,20HNUMERO DE ITERACOES=,I3)
1000 RETURN
     END
```

```
      SUBROUTINE NEWMAX (XLIM,DEL,NITER,NTOT)
      DOUBLE PRECISION U,V,W,ERROV,ERROW,AUX,DEL,ERROU,ERROX,UANT
      DOUBLE PRECISION ANT,XAST,ETMT,XXX,DX,EMAX
      DOUBLE PRECISION A(10),B(10),X(20)
      COMMON /COEF/A,B,X
      NDIV=100
      ALFA=0.8
      DO 350 I=2,NTOT-1
      J=0
      AUX=0.0
      ICOUNT=0
      CALL ERRO (X(I),ERROX)
    5 U=X(I)
   10 V=U+DEL
      W=U-DEL
      ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.NITER) GO TO 500
      CALL ERRO(V,ERROV)
      CALL ERRO(W,ERROW)
      IF (AUX*ANT.LE.0.0) J=J+1
      ANT=AUX
      AUX=(ERROV-ERROW)/(2.0*DEL)
      IF (DABS(AUX).LT.1.0D-10) GO TO 350
      UANT=U
      IF (ERROX.LT.0.0) AUX=-AUX
   12 U=UANT+DSINAL(AUX)/(FLOAT(J)*100)
      IF (U.GE.X(I+1).OR.U.LE.X(I-1)) GO TO 15
      CALL ERRO (U,ERROU)
      IF (DSINAL(ERROX).NE.DSINAL(ERROU)) GO TO 15
      GO TO 10
   15 J=J+1
      GO TO 12
  350 X(I)=U
      RETURN
  500 U=X(I)
      V=U+ALFA*(X(I+1)-U)
      W=U+ALFA*(X(I-1)-U)
      CALL ERRO (U,ERROU)
      SERROU=DSINAL(ERROU)
  280 DX=(V-W)/FLOAT(NDIV)
      EMAX=0.0
      XXX=W-DX
      DO 288 J=0,NDIV
      XXX=XXX+DX
      CALL ERRO (XXX,ETMT)
      IF (DSINAL(ETMT).NE.SERROU) GO TO 288
      IF (DABS(ETMT).LE.EMAX) GO TO 288
      EMAX=DABS(ETMT)
      XAST=XXX
      V=XAST+DX
      W=XAST-DX
  288 CONTINUE
      IF (DX.GE.XLIM) GO TO 280
      U=XAST
      GO TO 350
      RETURN
      END
```

```
      SUBROUTINE STURM (XX,IGRAU,A0,B0,N)
      DOUBLE PRECISION X(10),Y(10),P(10),Q(10),XX(1),AUX
      IGRAUX=IGRAU
      DO 2 I=1,IGRAUX+1
2     X(I)=XX(I)
      J=IGRAUX
      DO 10 I=1,IGRAUX
      Y(I)=J*X(I)
10    J=J-1
      CALL VALNUM (X,IGRAUX,DBLE(A0),P(1))
      CALL VALNUM (X,IGRAUX,DBLE(B0),Q(1))
      CALL VALNUM (Y,IGRAUX-1,DBLE(A0),P(2))
      CALL VALNUM (Y,IGRAUX-1,DBLE(B0),Q(2))
      K=2
      IF (IGRAUX.EQ.1) GO TO 100
      IGRAUY=IGRAUX-1
15    CALL MPDIV (X,IGRAUX,Y,IGRAUY)
      K=K+1
      DO 18 I=1,IGRAUX+1
 18   X(I)=-X(I)
      CALL VALNUM (X,IGRAUX,DBLE(A0),P(K))
      CALL VALNUM (X,IGRAUX,DBLE(B0),Q(K))
      IF (IGRAUX.EQ.0) GO TO 100
      DO 20 I=1,IGRAUY+1
      AUX=X(I)
      X(I)=Y(I)
20    Y(I)=AUX
      NN=IGRAUX
      IGRAUX=IGRAUY
      IGRAUY=NN
      GO TO 15
100   IA=0
      IB=0
      DO 110 I=1,K-1
      IF (P(I)*P(I+1).LE.0.0) IA=IA+1
      IF(Q(I)*Q(I+1).LE.0.0) IB=IB+1
110   CONTINUE
      N=IA-IB
      RETURN
      END
```

```
      SUBROUTINE MPDIV (X,IGRAUX,Y,IGRAUY)
      DOUBLE PRECISION X(1),Y(1),C
      I=IGRAUX-IGRAUY+1
      IF (I.LE.0) RETURN
 10   II=II+1
      C=X(II)/Y(1)
      X(II)=0.0
      DO 20 K=1,IGRAUY
      J=II+K
 20   X(J)=X(J)-C*Y(K+1)
      I=I-1
      IF (I.GT.0) GO TO 10
      DO 30 I=1,IGRAUY
 30   X(I)=X(IGRAUX+1-IGRAUY+I)
      IGRAUX=IGRAUY-1
      J=IGRAUX
      DO 100 I=1,IGRAUX
      IF (X(I).NE.0.0) GO TO 40
      J=J-1
100   CONTINUE
 40   IF (J.EQ.IGRAUX) RETURN
      DO 110 I=1,J+1
110   X(I)=X(IGRAUX-J+I)
      IGRAUX=J
      RETURN
      END
```

```
      SUBROUTINE VALNUM (A,N,X,Y)
      DOUBLE PRECISION A(N),X,Y
      Y=0.0
      DO 5 I=1,N+1
    5 Y=Y*X+A(I)
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION DELMAX(A,N,KK)
      DOUBLE PRECISION A(N)
      DELMAX=DABS(A(1))
      DO 20 I=1,N
      IF (KK.EQ.1) GO TO 30
      IF (DABS(A(I)).GT.DELMAX) GO TO 10
      GO TO 20
   30 IF (DABS(A(I)).LT.DELMAX) GO TO 10
      GO TO 20
   10 DELMAX=DABS(A(I))
   20 CONTINUE
      RETURN
      END
```

```
SUBROUTINE ERRO (X,E)
DOUBLE PRECISION A(10),B(10),XX(20)
COMMON /COFF/A,B,XX
COMMON //M,N,A0,B0
DOUBLE PRECISION X,E,YN,YD
CALL VALNUM (A,M,X,YN)
CALL VALNUM (B,N,X,YD)
CALL FUNCAO (X,E)
E=E-YN/YD
RETURN
END
```

```
      DOUBLE PRECISION FUNCTION DERRO(I,J)
      DOUBLE PRECISION A(10),B(10),X(20)
      DOUBLE PRECISION XX,F1,F2
      COMMON /COEF/A,B,X
      COMMON //M,N,A0,B0
      DOUBLE PRECISION D,D1
      K=0
      XINC=(B0-A0)/1000.0
      IF (J.GE.M+N+1) GO TO 100
      IF (J.NE.I-1) GO TO 110
      KK=1
 30   XX=X(I)+XINC
      CALL ERRO (XX,F1)
      XX=X(I)-XINC
      CALL ERRO (XX,F2)
 50   K=K+1
      D=(F1-F2)/(2.0*XINC)
      IF (K.EQ.2) GO TO 60
 40   D1=D
      XINC=XINC/2.0
      IF (KK.EQ.1) GO TO 30
      IF (KK.EQ.2) GO TO 130
      IF (KK.EQ.3) GO TO 230
 60   IF (D1.EQ.0.0) GO TO 500
      D1=DABS((D-D1)/D1)
      IF (D1.LE.0.0001) GO TO 500
      K=1
      GO TO 40
 500  DERRO=D
      RETURN
 110  DERRO=0.0
      RETURN
 100  IF (J.GE.2*M+N+2) GO TO 200
      JJ=J-M-N
 130  KK=2
      A(JJ)=A(JJ)+XINC
      CALL ERRO (X(I),F1)
      A(JJ)=A(JJ)-2.0*XINC
      CALL ERRO (X(I),F2)
      A(JJ)=A(JJ)+XINC
      GO TO 50
 200  JJ=J-2*M-N-1
 230  KK=3
      B(JJ)=B(JJ)+XINC
      CALL ERRO (X(I),F1)
      B(JJ)=B(JJ)-2.0*XINC
      CALL ERRO (X(I),F2)
      B(JJ)=B(JJ)+XINC
      GO TO 50
      END
```

```
      DOUBLE PRECISION FUNCTION DERIV(I,J)
      DOUBLE PRECISION A(10),B(10),X(20)
      COMMON /COEF/A,B,X
      COMMON //M,N,A0,B0
      DOUBLE PRECISION XX,F1,F2,F3
      DOUBLE PRECISION D,D1
      K=0
      NTOT=M+N+2
      XINC=(B0-A0)/1000.0
      IF (J.GE.M+N+1) GO TO 100
      IF (J.NE.I-NTOT) GO TO 110
 30   KK=1
      XX=X(I-NTOT+1)
      CALL ERRO (XX,F2)
      XX=XX-XINC
      CALL ERRO (XX,F1)
      F1=(F2-F1)/XINC
      XX=XX+2.0*XINC
      CALL ERRO (XX,F3)
      F2=(F3-F2)/XINC
 50   K=K+1
      D=(F2-F1)/XINC
      IF (K.EQ.2) GO TO 60
 40   D1=D
      XINC=XINC/2.0
      IF (KK.EQ.1) GO TO 30
      IF (KK.EQ.2) GO TO 130
      IF (KK.EQ.3) GO TO 230
 60   IF (D1.EQ.0.0) GO TO 500
      D1=DABS((D-D1)/D1)
      IF (D1.LE.0.0001) GO TO 500
      K=1
      GO TO 40
500   DERIV=D
      RETURN
 110  DERIV=0.0
      RETURN
 100  IF (J.GE.2*M+N+2) GO TO 230
 130  KK=2
      XX=X(I-NTOT+1)-XINC
      CALL VALNUM(B,N,XX,F1)
      F1=-XX**(2*M+N+1-J)/F1
      XX=XX+2.0*XINC
      CALL VALNUM (B,N,XX,F2)
      F2=-XX**(2*M+N+1-J)/F2
      GO TO 50
 230  KK=3
200   XX=X(I-NTOT+1)-XINC
      CALL VALNUM (A,M,XX,F1)
      CALL VALNUM (B,N,XX,F2)
      F1=F1*XX**(2*(M+N+1)-J)/(F2**2)
      XX=X(I-NTOT+1)+XINC
      CALL VALNUM (A,M,XX,F2)
      CALL VALNUM (B,N,XX,F3)
      F2=F2*XX**(2*(M+N+1)-J)/(F3**2)
      GO TO 50
      END
```

```
      FUNCTION DSINAL(X)
      DOUBLE PRECISION X
      IF (X) 10,20,30
   10 DSINAL=-1.0
      GO TO 40
   20 DSINAL=0.0
      GO TO 40
   30 DSINAL=1.0
   40 RETURN
      END
```

```
      SUBROUTINE RESEL  (NTOT,A,C,D,IERRO,MM)
C
C
      DOUBLE PRECISION TESTE,AMAX
      DOUBLE PRECISION A(MM,1) ,C(NTOT),D(NTOT),ID(70,70)
C
C ESTA SUBROTINA RESOLVE SISTEMA DE EQUACOES LINEARES POR INVERSAO
C   DA MATRIZ DOS COEFICIENTES PELO METODO DE GAUSS-JORDAN FAZENDO
C   PIVOTEMENTO COM O MAIOR ELEMENTO DA COLUNA EM CONSIDERACAO
C
C   DESCRICAO DOS PARAMETROS -
C        NTOT- ORDEM DA MATRIZ DO SISTEMA
C        A- MATRIZ DOS COEFICIENTES DO SISTEMA
C        C- VETOR DOS TERMOS INDEPENDENTES
C        D- VETOR QUE NA SAIDA CONTERA A SOLUCAO DO SISTEMA
C        IERRO- DENOTA OCORRENCIA OU NAO DE ANORMALIDADES DURANTE A
C               EXECUCAO
C        MM- NUMERO DE LINHAS DIMENSIONADAS PARA A NO PROGRAMA PRINCIPAL
C
C
C GERA MATRIZ IDENTIDADE
C
      DO 10 I=1,NTOT
      DO 5  J=1,NTOT
    5 ID(I,J)=0.
   10 ID(I,I)=1.
C
C INVERSAO DA MATRIZ A
C
      DO 60 I=1,NTOT
      AMAX=DABS(A(I,I))
      IND=I
      DO 20 J=I,NTOT
      TESTE=DABS(A(J,I))-AMAX
      IF (TESTE)  20,20,15
   15 AMAX=DABS(A(J,I))
      IND=J
   20 CONTINUE
      DO 30 K=1,NTOT
      AMAX=A(I,K)
      A(I,K)=A(IND,K)
      A(IND,K)=AMAX
      AMAX=ID(I,K)
      ID(I,K)=ID(IND,K)
      ID(IND,K)=AMAX
   30 CONTINUE
      IF (A(I,I)) 32,35,32
   35 IERRO=3
      RETURN
   32 DO 45 K=1,NTOT
      IF (K-I) 40,45,40
   40 A(I,K)= A(I,K)/A(I,I)
   45 ID(I,K)=ID(I,K)/A(I,I)
      A(I,I)=1.
      DO 60 J=1,NTOT
      AMAX=A(J,I)
      IF (J-I) 50,60,50
   50 DO 60 K=1,NTOT
      A(J,K)=A(J,K)-AMAX*A(I,K)
      ID(J,K)=ID(J,K)-AMAX*ID(I,K)
   60 CONTINUE
```

```
C
C GERA VETOR SOLUCAO D
C
      DO 70 I=1,NTOT
      D(I)=0.
      DO 70 J=1,NTOT
      A(I,J)=TD(I,J)
   70 D(I)=D(I)+TD(I,J)*C(J)
      RETURN
      END
```

```
      SUBROUTINE IGUAL (A,ATR,C,B,INDTIP,CC,MM1,MM2,M,N,II,NVAR,KPROB,L)
      DIMENSION INDTIP(1)
      DOUBLE PRECISION A(MM1,1),ATR(MM2,1),C(1),B(1),CC(1),COEF,BB
C
C     FINALIDADE -
C                   DADO UM CONJUTO DE DESIGUALDADES LINEARES NAS VARIA -
C                   VEIS "X(I)" . DETERMINAR QUAIS VARIAVEIS SAO RESTRI -
C                   TAS OU NAO E A SEGUIR MONTAR A MATRIZ DOS COEFICIEN -
C                   TES DAS RESTRICOES COMO IGUALDADES
C
C     ESCLARECIMENTOS -
C
C                          SE UMA VAR. E ".LE. 0" , FAZEMOS "X=-X"
C                          ADICIONAMOR AS VAR. DE FULGA CONVENIETEMENTE
C                          O N. TOTAL DE RESTRICOES PROVAVELMENTE DIMINUI -
C           RA JA QUE AS RESTRICOES CORRESPONDENTES AS VAR. RESTRITAS
C           NAO APARECERAO EM "ATR"
C                          SE TIVER ALGUMA RESTRICAO COM IGUALDADE,DEVEMOS
C           TRANSFORMA-LA EM DUAS DESIGUALDADES NA MONTAGEM DE "A" ,AN -
C           TES DE CHAMAR A SUBROTINA "IGUALD".
C                          ADIMITE-SE QUE PARA PROBLEMA DE MAXIMO AS RESTRI
C           COES ESTEGEM NA FORMA DE  ".LE."  DA MESMA FORMA PARA MINIMO
C           EM  ".GE."
C
C     PAREMETROS
C
C      A       - MATRIZ DAS RESTRICOES * DIM(M,N)
C      ATR     - MATRIZ DAS RESTRICOES COM IGUALDADES * DIM(II,NVAR)
C      C       - VETOR COM COEFICIENTES DA FUNCAO OBJ * DIM(N)
C      B       - VETOR COM TERMOS CONTANTES  * DIM(M)
C      INDTIP  - ESPECIFICA O TIPO DA VARIAVEL = -1  .LE. 0
C                                              =  0  IRRESTRITA
C                                              =  1  .GE. 0
C      CC      - VETOR COM COEF. TRANSF. DA F. OBJ. * DIM(NVAR)
C      MM1     - N. DE LINHAS DIMENSIONADAS PARA MATRIZ ASSOCIADA COM "A"
C      MM2     - N. DE LINHAS DIMENSIONADAS PARA MAT. ASSOCIADA A "ATR"
C      M       - N. DE RESTRICOES
C      N       - N. DE VARIAVEIS
       II      - N. DE RESTRICOES APOS A TRANSFORMACAO
C      NVAR    - N. DE VARIAVEIS  APOS A TRANSFORMACAO
C      KPROB   - TIPO DE PROLEMA   =1 - MINIMO
C                                  =2 - MAXIMO
C
C     *** IMPORTANTE ***
C             ESTA SUBROTINA CHAMA A SURROTINA
C     ** MARINT **
C
      DO 10 J=1,N
   10 INDTIP(J)=0
      II=1
      JJ=M
      DO 70 I=1,M
      ICONT=0
      DO 20 J=1,N
      IF(A (II,J).EQ.0.)GO TO 20
      COEF=A(II,J)
      ICONT=ICONT+1
      ID=J
   20 CONTINUE
      IF(ICONT.GT.1) GO TO 60
      IF(B(II).NE.0.) GO TO 60
```

```
      IF((KPROB.EQ.1.AND.COEF.LT.0.).OR.(KPROB.EQ.2.AND.COEF.GT.0))GO
     * TO 40
      INDTIP(ID)=1
      GO TO 50
   40 INDTIP(ID)=-1
   50 BB=B(II)
      B(II)=B(JJ)
      B(JJ)=BB
      CALL MARINT(A,N,II,JJ,MM1)
      JJ=JJ-1
      GO TO 70
   60 II=II+1
   70 CONTINUE
      II=II-1
      JJ=0
      DO 100 J=1,N
      JJ=JJ+1
      IF(INDTIP(J).GE.0) K=1
      IF(INDTIP(J).EQ.-1) K=-1
      DO 80 I=1,II
   80 ATR(I,JJ)=K*A(I,J)
      CC(JJ)=K*C(J)
      IF(INDTIP(J).NE.0) GO TO 100
      JJ=JJ+1
      DO 90 I=1,II
   90 ATR(I,JJ)=-A(I,J)
      CC(JJ)=-C(J)
  100 CONTINUE
      IF(KPROB.EQ.2)XK=1
      IF(KPROB.EQ.1)XK=-1
      NVAR=JJ
      IF (L.EQ.1) RETURN
      DO 110 J=1,II
      DO 110 I=1,II
      ATR(I,JJ+J)=0.
      IF(I.EQ.J) ATR(I,JJ+J)=XK
  110 CONTINUE
      NVAR=JJ+II
      RETURN
      END
```

```
C
      SUBROUTINE MARINT(A,N,LA,LB,MM)
C
      DOUBLE PRECISION A(MM,1),SAVE
C
C     OBJETIVO
C
C                TROCAR DUAS LINHAS DE UMA MATRIZ
C
C     DESCRICAO DOS PARAMETROS
C
C         A  NOME DA MATRIZ
C         N  NUMERO DE COLUNAS EM "A"
C        LA  LINHA A SER TROCADA COM LINHA "LB"
C        LB  LINHA A SER TROCADA COM LINHA "LA"
C        MM N. DE LINHAS DIMENS. NO P.P. PARA A MAT. ASSOCIADA COM "A"
C
      DO 10 J=1,N
      SAVE=A(LA,J)
      A(LA,J)=A(LB,J)
   10 A(LB,J)=SAVE
      RETURN
      END
```

```
C
C          SUBROTINA  R E G M A G
C
      SUBROUTINE REGMAG(A,B,C,AA,INDVB,LOCVAR,P,KPROB,ERRO,Z,ITER,
     *KTIPO,MM1,MM2,KATVAL,KVIAVE,KSAIDA,M,N,S)
      INTEGER R,S
C
      DIMENSION INDVB(1),LOCVAR(1),ME(65),MF(65),FINLAO(4)
      DOUBLE PRECISION A(MM1,1),AA(MM2,1),B(1),C(1),P(1),XMIN,Z
C
C     FINALIDADE -
C
C                      MAXIMIZAR OU MINIZAR
C                                                 C * X
C                      SUJEITO AS RESTRICOES
C                                                 X .GE. ZERO
C                                               E  A * X .EQ. B
C
C     OBSERVACAO -
C
C            ADIMITE-SE QUE O PROBLEMA JA ESTEJE NA FORMA GERAL,OU SEJA,
C        AS RESTRICOES EM FORMA DE IGUALDADE.
C           CASO CONTRARIO TRANSFORME-O PRIMEIRAMENTE PARA A FORMA
C        ACIMA MENCIONADO ANTES DE CHAMAR ESTA SUBROTINA.
C
C     REFERENCIA -
C                      METODO - O SIMPLEX USANDO MULTIPLICADORES . GEORGE B.
C        DANTZIG
C
C     DESCRICAO DOS PARAMETROS
C
C          A       - MATRIZ DAS RESTRICOES - DIM(M,N)   - VEJA **
C          B       - VETOR COM TERMOS CONTANTES -DIM(M)   - VEJA **
C          C       - VETOR OBJETIVO -DIM(M)   - VEJA **
C          AA      - MATRIZ DE DIM(M+2,M+1) , CONTEM NO FINAL DO CICLO K -
C                    O INVERSO DA MATRIZ FORMADA PELOS VETORES COLUNAS DE
C                    DE "A" CORESPONDENDO AS VAR. QUE ESTAO NA BASE NAS
C                    "M" PRIMEIRAS LINHAS E COLUNAS
C                    MULTIPLICADOR "-PI" E "-Z" - LINHA M+1
C                    MULTIPLICADOR "-SIGMA" E "-W" - LINHA M+2
C          INDVB   - VETOR INDICADOR DAS VARIAVEIS BASICAS - DIM(M)
C          LOCVAR  - VETOR COM A LOCALIZACAO DAS VARIAVEIS - DIM(N)
C                             = 0 - NA BASE
C                             = 1 - FORA DA BASE
C          ME      - VETOR DE TRABALHO - DIM(M)
C          MF      - VETOR DE TRABALHO - DIM(M)
C           P      - VETOR DE TRABALHO - DIM(M)
C          KPROB   - TIPO DE OTIMZACAO
C                             =1  PROBLEMA DE MINIMO
C                             =2  PROBLEMA DE MAXIMO
C          ERRO    - PARAMETRO DE ARREDONDAMENTO
C          Z       - VALOR DA FUNCAO OBJETIVO
C          ITER    - NUMERO DE ITERACOES
C          KTIPO   - VIABILIDADE DO PROBLEMA
C                              =0 VIAVEL LIMITADO
C                              =1 ILIMITADO
C                              =2 INVIAVEL
C          MM1     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "A" - DIM. NO PP.
C          MM2     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "AA" - DIM. NO PP
C          KATVAL  - INDICA DE QUANTAS EM QUANTAS ITERACOES SAO EFETUADAS
C                    ATUALIZACOES DA INVERSA DA MATRIZ BASICA
```

```
C           KVIAVE - INDICADA SE DESEJAMOS APENAS FASE 1 OU AMBAS
C                                =0 DESEJA  A  SOLUCAO BASICA OTIMA
C                                =1 DESEJA UMA SOLUCAO BASICA VIAVEL
C           KSAIDA - OPCAO DE SAIDAS
C                                =0 TABLEAU COMPLETO A CADA ITERACAO
C                                =1 SOLUCAO VIAVEL E CUSTOS DOS FATORES POR
C                                   ITERACAO
C                                =2 SOLUCAO OTIMA
C                                =3 APENAS TRANSFERENCIA POR MEIO DOS PARA-
C                                   METROS FORMAIS
C           TRAB   - VETOR DE TRABALHO - DIM(M**2)
C           M      - N. DE RESTRICOES DO PROBLEMA
C           N      - N. DE VARIAVEIS DO PROBLEMA
C           S      - INDICA A ULTIMA VARIAVEL QUE ENTROU OU QUE E CANDIDA-
C                    TA A ENTRAR NA BASE
C
C  ** OBSERVACOES -
C
C             DEVIDOS OS VETORES E MATRIZES ABAIXOS DESCRIMINADOS TEREM
C             OUTRAS FUNCOES DENTRO DA SUBROTINA ALEM DAS ACIMAS ESPECIFI-
C             CADAS,TERAMOS AS DIMENSOES PARA
C
C           A - DIM (M+2,N+1)
C           B - DIM (M+2)
C           C - DIM (N+1)
C
C      *** IMPORTANTE ***
C               ESTA SUBROTINA CHAMA AS SUBROTINAS :
C      ** MONTCD ** ** MINIMO ** ** MONTBP ** ** LINHAR ** **PIVOMA **
C      ** RESEL **
C                                      E AS FUNCOES
C      ** VALK ** ** VALKI ** ** VALKF **
C
      IF(KSAIDA.EQ.3) GO TO 33
      KI=1
      IF(KPROB.EQ.2) KI=3
      KF=KI+1
      WRITE(6,2010)(FINLAO(I),I=KI,KF)
 2010 FORMAT(2X,2A6,/2X,9(1H-),//12X, 5HC * X//2X,10HSUJEITOS A
     * /2X,10(1H-),/12X,9HA * X = B,/
     *16X,1HE,/15X,11HX .GE. ZERO,////)
      K=VALK(N, 8 )
      DO 10 J=1,K
      DO 10 I=1,M
      KI=VALKI(J, 8)
      KF=VALKF(KI,J,K,N, 8)
      WRITE(6,2050)((A(I,L),L=KI,KF)
      IF(I.EQ.M) WRITE(6,2020)
 2020 FORMAT(//)
   10 CONTINUE
      WRITE(6,2030)
 2030 FORMAT(///2X, 7HVETOR C,/2X,7(1H-))
      WRITE(6,2050)(C(I),I=1,N)
      WRITE(6,2040)
 2040 FORMAT(///2X, 7HVETOR B,/2X,7(1H-))
      WRITE(6,2050)(B(I),I=1,M)
   33 IF(KPROB.EQ.1) GO TO 38
      IF(KSAIDA.LE.2) WRITE(6,2043)
 2043 FORMAT(//2X,20HPROBLEMA EQUIVALENTE,/2X,20(1H-),//2X,9HMINIMIZAR,/
     1 2X,9(1H-),//12X, 7H- C * X,//2X,10HSUJEITOS A,/2X,10(1H-),/12X,
     2 9HA * X = B,/16X,1HE,/15X,11HX .GE. ZERO,//5X,
```

```
    3 31HMAXIMO C * X = - MINIMO - C * Y,/5X,31(1H-),)
      DO 35 I=1,N
   35 C(I)=-C(I)
   38 N1=N+1
      M2=M+2
      M1=M+1
      MN=M+N
      IF(KSAIDA.LE.2) WRITE(6,2045)
 2045 FORMAT(1H1,4X,8HMATRIZ A,/5X,8(1H-))
      DO 40 J=1,N1
      A(M2,J)=0.
   40 A(M1,J)=C(J)
      A(M1,N1)=0.
      DO 60 I=1,M
      A(I,N1)=B(I)
      IF(B(I).GE. 0.)GO TO 55
      DO 50 J=1,N1
   50 A(I,J)=-A(I,J)
   55 DO 60 J=1,N1
   60 A(M2,J)=A(M2,J)-A(I,J)
      IF(KSAIDA.EQ.3) GO TO 68
      K=VALK(N1,8)
      DO 64 J=1,K
      DO 64 I=1,M2
      KI=VALKI(J,8)
      KF=VALKF(KI,J,K,N1,8)
      WRITE(6,2050) (A(I,L),L=KI,KF)
      IF(I.EQ.M2) WRITE(6,2020)
   64 CONTINUE
 2050 FORMAT(5X,8E14.6)
   68 DO 70 I=1,M2
      DO 70 J=1,M
      IF(I.EQ.J)GO TO 75
      AA(I,J)=0.
      GO TO 70
   75 AA(I,J)=1.
   70 CONTINUE
      DO 80 I=1,M
      J=N+I
      INDVB(I)=J
   80 AA(I,M1)=A(I,N1)
      AA(M1,M1)=0.
      AA(M2,M1)=A(M2,N1)
      DO 90 I=1,M
   90 LOCVAR(I)=1
      ITER=0
      KTIPO=0
      ICONTI=0
      M1M2=M2
      IF (KSAIDA.EQ.3) GO TO 100
      WRITE(6,2052)
 2052 FORMAT(///5X,9HMATRIZ AA,/5X,9(1H-))
      K=VALK(M1,8)
      DO 92 J=1,K
      DO 92 I=1,M2
      KI=VALKI(J,8)
      KF=VALKF(KI,J,K,M1,8)
      WRITE(6,2050)(AA(I,L),L=KI,KF)
      IF(I.EQ.M2) WRITE(6,2020)
   92 CONTINUE
 2060 FORMAT(5X,8F14.6)
```

```
      WRITE(6,2065)(INDVB(I),I=1,M)
 2065 FORMAT(///5X,11HVETOR INDVB,/5X,11(1H-),/(5X,40I3))
      WRITE(6,2070)(LOCVAR(I),I=1,N)
 2070 FORMAT(///5X,12HVETOR LOCVAR,/5X,12(1H-),/(5X,40I3))
      IF(KSAIDA.EQ.1) WRITE(6,2100)
 2100 FORMAT(1H1)
  100 CALL MONTCD(A,AA,LOCVAR,C,M,N,M1M2,MM1,MM2,ERRO)
      IF(KSAIDA.LE.1) WRITE(6,2130) (C(L),L=1,N)
 2130 FORMAT(//// 5X, 6HCUSTOS,/5X,6(1H-),/(5X,8E14.6))
      CALL MINIMO(C,N ,XMIN,IND)
      S=IND
      IF(XMIN.GE.0.)GO TO 270
      CALL MONTBP(A,AA,B,M,M1,M2,M1M2,MM1,MM2,S,ERRO)
      IF(KSAIDA.LE.1) WRITE(6,2140) (B(L),L=1,M)
 2140 FORMAT(//11X,1H-,/5X,7HVETOR P,/5X,7(1H-),/(5X,8F14.6))
      CALL LINHAR(AA,B,M,MM2  ,R,S,MF,MF,ERRO,KTIPO,KSAIDA)
      IF(KTIPO.EQ.1) RETURN
      ITER=ITER+1
      ICONTI=ICONTI+1
      CALL PIVOMA(A,AA,B,INDVB,M,N,M1,M1M2,MM1,MM2,ICONTI,KATVAL,
     *ERRO,KSAIDA,ITER,R,S,LOCVAR)
      GO TO 100
  270 IF(M1M2.EQ.M1) GO TO 340
      IF(AA(M2,M1).EQ. 0.)GO TO 280
      KTIPO=2
      IF(KSAIDA.LT.3) WRITE(6,2175)
 2175 FORMAT(///5X,17HPROBLEMA INVIAVEL,/5X,8(1H-),1X,8(1H-))
      RETURN
  280 M1M2=M1
      IF(KSAIDA.LT.3)WRITE(6,2180)
 2180 FORMAT(///5X,35HFINAL DA FASE 1 ** INICIO DA FASE 2,/5X,35(1H-)/)
      IF(KVIAVE.EQ.1) RETURN
      GO TO 100
  340 Z=-AA(M1,M1)
      IF(KPROB.EQ.2) Z=-Z
      IF(KSAIDA.EQ.3) GO TO 999
      WRITE(6,2260) Z ,ITER
 2260 FORMAT((1H1,5X,120(1H*),///5X,19HSOLUCAO OTIMA   Z =,E14.8,/5X,
     *33(1H-),//5X,17HN. DE ITERACOES =,I3,/5X,20(1H-)///))
      WRITE(6,2270)(INDVB(I),AA(I,M1),I=1,M)
 2270 FORMAT(45X,33HVARIAVEIS NA BASE     -     VALORES,/45X,37(1H-),/(
     *52X,I3,11X,1H-,1X,E14.8))
      WRITE(6,2280)
 2280 FORMAT(//5X,120(1H*))
  999 RETURN
      END
```

```
C
      SUBROUTINE MONTCD(A,AA,KB,C,M,N,MM,MM1,MM2,ERRO)
C
      DIMENSION KB(1)
      DOUBLE PRECISION A(MM1,1),AA(MM2,1),C(1)
C
C     FINALIDADE -
C
C                 MONTA O VETOR C COM O CUSTO DOS FATORES RELATIVOS PA-
C           RA A PRESENTE ITERACAO, E FUNCAO DAS VAR. BASICAS.
C
C     PARAMETROS -
C
C          A       - MATRIZ DAS RESTRICOES - DIM(M,N)
C          AA      - AREA DE TRABALHO COMPREENDENDOO INVERSO DA BASE,
C                    "-PI","-SIGMA","-Z" E "-W"
C          KB      - VETOR COM A LOCALIZACAO DAS VARIAVEIS * DIM(N)
C                            = 0 NA BASE
C                            = 1 FORA DA BASE
C          C       - VETOR COM O CUSTO DOS FATORES CALCULADO * DIM(N)
C          M       - N. DE RESTRICOES DO PROBLEMA
C          N       - N. DE VARIAVEIS DO PROBLEMA
C          MM      - =M1 SE ESTAMOS NA FASE 1
C                    =M2 SE ESTAMOS NA FASE 2
C          MM1     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "A" - DIM. NO PP.
C          MM2     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "AA" - DIM. NO PP
C          ERRO    - PARAMETRO DE ARREDONDAMENTO
C
      DO 30 J=1,N
      IF(KB(J).EQ.0) GO TO 20
      C(J)=A(MM,J)
      DO 10 I=1,M
   10 C(J)=C(J)+AA(MM,I)*A(I,J)
      GO TO 30
   20 C(J)=0.
   30 CONTINUE
      DO 40 I=1,N
      IF (DABS(C(I)).LE.ERRO) C(I)=0.
   40 CONTINUE
      RETURN
      END
```

```
C
      SUBROUTINE MINIMO(TRANSF,N,XMIN,IND)
C
      DOUBLE PRECISION TRANSF(1),XMIN
C
C    FINALIDADE -
C
C                 DETERMINA A COORDENADA MINIMA DE UM VETOR E SUA POSICAO
C
C     PARAMETROS
C
C          N      - N. DE COORDENADAS
C          TRANSF - VETOR DE ENTRADA
C          XMIN   - VALOR MINIMO
C          IND    - POSICAO DESTE VALOR NO VETOR
C
      XMIN=.1E+60
      DO 10 I=1,N
      IF(XMIN.LE.TRANSF(I))GO TO 10
      XMIN=TRANSF(I)
      IND=I
   10 CONTINUE
      RETURN
      END
```

```
C
      SUBROUTINE MONTBP(A,AA,B,M,M1,M2,M1M2,MM1,MM2,IS,ERRO)
C
      DOUBLE PRECISION A(MM1,1),AA(MM2,1),B(1)
C
C     FINALIDADE -
C
C                  ESCREVE A VARIAVEL CANDIDATA A ENTRAR NA BASE
C              EM TERMOS DAS VAR. BASICAS , OU SEJA , CALCULA O VETOR
C              P PARA A ITERACAO PRESENTE
C
C     PARAMETROS
C
C      A       - MATRIZ DAS RESTRICOES * DIM(M,N)
C      AA      - AREA DE TRABALHO.CONTEM O INVERSO DA MATRIS DOS COEF.
C                DAS VAR. BASICAS * DIM(M+2,M+1)
C      B       - VETOR ONDE OS VALORES CALCULADOS SAO GUARDADOS
C      M       - N.DE RESTRICOES DO PROBLEMA
C      M1      - = M + 1
C      M2      - = M + 2
C      M1M2    - =M1 SE ESTAMOS NA FASE 1
C                =M2 SE ESTAMOS NA FASE 2
C      MM1     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "A" - DIM. NO PP.
C      MM2     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "AA" - DIM. NO PP
C      IS      - INDICADOR DA VAR. CANDIDATA A ENTRAR NA BASE
C      ERRO    - PARAMETRO DE ARREDONDAMENTO
C
      B(M1)=A(M1,IS)
      DO 30  I=1,M
   30 B(M1)=B(M1)+AA(M1,I)*A(I,IS)
      IF(M1M2.EQ.M1) GO TO 50
      B(M2)=A(M2,IS)
      DO 40  I=1,M
   40 B(M2)=B(M2)+AA(M2,I)*A(I,IS)
   50 DO 60  I=1,M
      B(I)=0.
      DO 60  J=1,M
   60 B(I)=B(I)+AA(I,J)*A(J,IS)
   80 DO 90 I=1,M1M2
      IF (DABS(B(I)).LE.ERRO) B(I)=0.
   90 CONTINUE
      RETURN
      END
```

```
C
      SUBROUTINE LINHAR(AA,P,M,   M2,IR,IS,IND,NIND,ERRO,KTIPO,KSAIDA)
C
      DOUBLE PRECISION AA(M2,1),P(1),XMIN,QBP
      DIMENSION IND(1),NIND(1)
C
C    FINALIDADE -
C
C       DETERMINA A POSICAO DA VARIAVEL QUE VAI SAIR DA BASE E FAZ O
C       PIVOTAMENTO
C       NOS CASOS DEGENERADOS APLICA-SE A REGRA DO LEXOGRAFICO
C
C      PARAMETROS
C
C           AA     - AREA DE TRABALHO COMPREENDENDOO INVERSO DA BASE,
C                    "-PI","-SIGMA","-Z" E "-W"
C           P      - VETOR COLUNA DA VARIAVEL QUE VAI ENTRAR NA BASE
C           M      - N. DE VAR. DA BASE
C           M2     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "A" - DIM. NO PP .
C           IS     - POSICAO DA VAR. QUE VAI ENTRAR NA BASE
C           IR     - POSICAO DA VAR. QUE VAI SAIR DA BASE
C           IND    - VETOR DE TRABALHO - DIM(M)
C           NIND   - VETOR DE TRABALHO - DIM(M)
C           ERRO   - PARAMETRO PARA ARREDONDAMENTOS
C           KTIPO  - VEJA SUBROTINA REGMAG
C           KSAIDA - VEJA SUBROTINA REGMAG
C
      M1=M+1
      XMIN=.1E+60
      NDEG=0
      NPNEG=0
      DO 10 I=1,M
      IF(P(I).GT. 0.) GO TO 20
      NPNEG=NPNEG+1
      GO TO 10
   20 QBP=AA(I,M1)/P(I)
      IF(QBP-XMIN) 30,40,10
   30 XMIN=QBP
      NDEG=1
      IND(NDEG)=I
      GO TO 10
   40 NDEG=NDEG+1
      IND(NDEG)=I
   10 CONTINUE
      IF(NPNEG.LT.M) GO TO 50
      KTIPO=1
      IF(KSAIDA.LT.3)WRITE(6,2000)
 2000 FORMAT(///5X,18HPROBLEMA ILIMITADO./5X,8(1H-),1X,9(1H-),)
      RETURN
   50 IF(NDEG.GT.1)GO TO 60
      IR=IND(1)
      RETURN
   60 DO 120 J=1,M
      IF(J.EQ.1) GO TO 80
      NDEG=NLEX
      DO 70 K=1,NDEG
   70 IND(K)=NIND(K)
   80 XMIN=.1E+60
      NLEX=0
      DO 110 I=1,NDEG
      QBP=AA(IND(I),J)/P(IND(I))
```

```
      IF(QBP-XMIN)90,100,110
   90 XMIN=QBP
      NLEX=1
      NIND(NLEX)=IND(I)
      GO TO 110
  100 NLEX=NLEX+1
      NIND(NLEX)=IND(I)
  110 CONTINUE
      IF(NLEX.EQ.1) GO TO 130
  120 CONTINUE
  130 IR=NIND(1)
      RETURN
      END
```

```
C
      SUBROUTINE PIVOMA(A,AA,B,INDVB,M,N,M1,M1M2,MM1,MM2,ICONTI,KATVAL,
     *ERRO,KSAIDA,ITER,R,S,LOCVAR)
      INTEGER R,S
C
      DOUBLE PRECISION A(MM1,1),AA(MM2,1),B(1),Z
      DIMENSION INDVB(1),LOCVAR(1)
      DOUBLE PRECISION D
C
C     FINALIDADE -
C
C                     PIVOTA SOBRE O ELEMENTO A(R,S). A INVERSA DA MATRIZ
C               DOS COEF. DAS VAR. BASICAS,O VETOR PI E/OU SIGMA E O VETOR
C               DOS TERMOS CONSTANTES SAO ATUALIZADOS DE KATVAL EM KATVAL
C               ITERACOES CONFORME PODEMOS OBSERVAR NO DANTZIG,CAP 5, PARA-
C               GRAFO 8.
C
C     PARAMETROS
C
C      A       - MATRIZ DAS RESTRICOES - DIM(M,N)
C      AA      - AREA DE TRABALHO COMPREENDENDO INVERSO DA BASE,
C                "-PI","-SIGMA","-Z" E "-W"
C      B       - VETOR COM OS VALORES CALCULADOS NA SUB. MONTBP
C      INDVB   - VETOR INDICADOR DAS VARIAVEIS BASICAS - DIM(M)
C      M       - N. DE RESTRICOES DO PROBLEMA
C      N       - N. DE VARIAVEIS DO PROBLEMA
C      M1      - = M + 1
C      M1M2    - =M1 SE ESTAMOS NA FASE 1
C                =M2 SE ESTAMOS NA FASE 2
C      MM1     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "A" - DIM. NO PP.
C      MM2     - N. DE LINHAS DA MATRIZ ASSOC. A MAT "AA" - DIM. NO PP.
C      ICONTI  - ACUMULADOR COM O N. DE ITERACOES JA EFETUADAS APOS A
C                ULTIMA ATUALIZACAO DA AREA DE TRABALHO * MAT AA
C      KATVAL  - INDICA DE QUANTAS EM QUANTAS ITERACOES SAO EFETUADAS
C                ATUALIZACOES DA INVERSA DA MATRIZ BASICA
C      KSAIDA  - OPCAO DE SAIDAS  *  VEJA SUB. REGMAG
C      ITER    - NUMERO DE ITERACOES
C      R       - INDICADOR DA VARIAVEL CONDIDATA A ENTRAR NA BASE
C      S       - INDICADOR DA VARIAVEL QUE VAI SAIR DA BASE
C      ME      - VETOR DE TRABALHO * DIM(M)
C      MF      - VETOR DE TRABALHO * DIM(M)
C      LOCVAR  - VETOR COM A LOCALIZACAO DAS VARIAVEIS - DIM(N)
C                               = 0 - NA BASE
C                               = 1 - FORA DA BASE
C      TRAB    - AREA DE TRABALHO * VETOR COM DIM(M**2)
C
C     *** IMPORTANTE ***
C              ESTA SUBROTINA CHAMA AS SUBROTINAS :
C     *** RESEL ***
C
      N1=N+1
      M2=M+2
      LOCVAR(S)=0
      IF(INDVB(R).LE.N) LOCVAR(INDVB(R))=1
      INDVB(R)=S
      IF(ICONTI.EQ.KATVAL) GO TO 40
      DO 10  J=1,M1
   10 AA(R,J)=AA(R,J)/B(R)
      DO 30  I=1,M1M2
      IF(I.EQ.R) GO TO 30
      DO 20  J=1,M1
```

```
   20 AA(I,J)=AA(I,J)-B(I)*AA(R,J)
   30 CONTINUE
      GO TO 80
   40 ICONTI=0
      DO 50  K=1,M
      IF(INDVB(K).GT.N) GO TO 44
      DO 42  I=1,M
   42 AA(I,K)=A(I,INDVB(K))
      GO TO 50
   44 DO 46 I=1,M
   46 AA(I,K)=0.
      L=INDVB(K)-N
      AA(L,K)=1.
   50 CONTINUE
      CALL RESEL (M,AA,B,B,IERRO,MM2)
      DO 52 I=1,M
      AA(I,M1)=0.
      DO 52 J=1,M
   52 AA(I,M1)=AA(I,M1)+AA(I,J)*A(J,N1)
      AA(M1,M1)=0.
      DO 54  J=1,M
      AA(M1,J)=0.
      DO 53 I=1,M
      IF(INDVB(I).GT.N) GO TO 53
      AA(M1,J)=AA(M1,J)-A(M1,INDVB(I))*AA(I,J)
   53 CONTINUE
   54 AA(M1,M1)=AA(M1,M1)+AA(M1,J)*A(J,N1)
      IF(M1M2.EQ.M1) GO TO 80
      AA(M2,M1)=A(M2,N1)
      DO 57  J=1,M
      AA(M2,J)=0.
      DO 56 I=1,M
      IF(INDVB(I).GT.N) GO TO 56
      AA(M2,J)=AA(M2,J)-A(M2,INDVB(I))*AA(I,J)
   56 CONTINUE
   57 AA(M2,M1)=AA(M2,M1)+AA(M2,J)*A(J,N1)
  80  DO 90 I=1,M1M2
      DO 90 J=1,M1
      IF (DABS(AA(I,J)).LE.ERRO) AA(I,J)=0.0
  90  CONTINUE
      IF(KSAIDA.GE.2) RETURN
      IF(KSAIDA.EQ.0) GO TO 110
      WRITE(6,2100)ITER
 2100 FORMAT(////5X,10HITERACAO =,I3,/5X,13(1H-))
      K=VALK(M,8 )
      DO 100 J=1,K
      KI=VALKI(J, 8)
      KF=VALKF(KI,J,K,M,8)
      WRITE(6,2110)(INDVB(L),L=KI,KF)
 2110 FORMAT(2X,13HVAR. BASICAS ,8(5X,I3,4X))
  100 WRITE(6,2120)(AA(L,M1),L=KI,KF)
 2120 FORMAT(2X,13HVALORES V. B.,8E14.6,/)
      Z=-AA(M1M2,M1)
      WRITE(6,2140) Z
 2140 FORMAT(//5X,17HFUNCAO OBJETIVO =,E14.6,/5X,31(1H-),///)
      RETURN
  110 K=VALK(M1,8.)
      WRITE(6,2145) ITER
 2145 FORMAT(1H1,20X,41HT     A     B     L     E     1     U     N.     ,I3,/
     *21X,41H-     -     -     -     -     -     -     --     ,3H---,//)
      DO 120 J=1,K
```

```
      DO 120 I=1,M
      KI=VALKI(J, 8)
      KF=VALKF(KI,J,K,M1,8)
      WRITE(6,2160)INDVB(I),(AA(I,L),L=KI,KF)
 2160 FORMAT(3X,I2,2X,8E14.6)
      IF(I.LT.M) GO TO 120
      WRITE(6,2170) (AA(M1,L),L=KI,KF)
 2170 FORMAT(/7X,8E14.6,)
      IF(M1M2.GT.M1) WRITE(6,2180)(AA(M1M2,L),L=KI,KF)
 2180 FORMAT( 7X,8E14.6,//)
  120 CONTINUE
      RETURN
      END
```

```
C
      INTEGER FUNCTION VALK(N, L)
C
C     FINALIDADE -
C
C                 DETERMINAR O N. DE LINHAS PARA INPRIMIR "N" VALOR5S 1
C
      K=N/L
      VALK=K+1
      IF(K*L.EQ.N) VALK=K
      RETURN
      END
```

```
C
      INTEGER FUNCTION VALKI(K,NL)
C
C     FINALIDADE -
C
C                   DETERMINAR O INDICE DA PRIMEIRA VARIAVEL NA "K-ESIMA"
C              LINHA DE IMPRESSAO
C
C     PARAMETROS
C
C         NL     - TAXA DE IMPRESSAO POR LINHA
C
      VALKI=(K-1)*NL+1
      RETURN
      END
```

```
C
      INTEGER FUNCTION VALKF(KI,J,K,N,NL)
C
C     FINALIDADE -
C
C                  DETERMINAR O INDICE DA ULTIMA VARIAVEL DA J-ESIMA
C           LINHA
C
C     PARAMETROS
C
C      K       - N. DE LINHAS NECESSARIAS PARA IMPRIMIR "N"  VALORES
C                A TAXA DE "NL" POR LINHA
C      KI      - INDICE DA PRIMEIRA VARIAVEL DA K-ESIMA LINHA
C
      VALKF=KI+NL-1
      IF(J.EQ.K) VALKF=N
      RETURN
      END
```