DIRCEU DOUGLAS SALVETTI

ANÁLISE DE INTERVALOS

Dissertação de Mestrado apresentada à Comissão de Pós-Graduação da Escola Politécnica da Universidade de São Paulo.

Orientador: Prof. Dr. Ivan de Queiroz Barros

São Paulo
1970

DIRCEU DOUGLAS SALVETTI

ANALISE DE INTERVALOS

SÃO PAULO

1970

ANALISE DE INTERVALOS

Dissertação de Mestrado apresentada à Comissão de Pós-Graduação da Escola Politécnica da Universidade de São Paulo

Orientador: Prof.Dr. Ivan de Queiroz Barros

SÃO PAULO

1970

Agradecemos profundamente a todos aqueles que direta ou indiretamente contribuiram para a realização deste trabalho e, em particular, aos professores

Dr. Ivan de Queiroz Barros
pela excelente orientação

Alesio João de Caroli
pela leitura dos originais e inúmeras sugestões

Drs. Waldyr Muniz Oliva e Leo Borges Vieira
pelos constantes estímulos

PREFACIO

Na resolução de problemas matemáticos há três tipos de êrros que podem ocorrer nos cálculos:

(1) propagação dos êrros dos dados iniciais.

(2) acumulação dos êrros de arredondamento.

(3) êrro de truncamento.

Suponhamos, por exemplo, que o problema $y'' = ay^{-b}$, $y(0)=y_0$, $y(1) = y_1$, descreva algum processo físico e que os números $\tilde{a}$, $\tilde{b}$, $\tilde{y}_1$ e $\tilde{y}_1$ são apenas aproximações das quantidades a, b, y_0 e y_1, respectivamente. Neste caso o problema que se coloca é: qual a magnitude do êrro na solução em decorrência dos êrros cometidos na determinação experimental dos valores das quantidades a, b, y_0 e y_1 ?

O êrro de arredondamento é causado pela execução dos cálculos com números arredondados para um número finito de dígitos. Este êrro decorre da impossibilidade de se representar um número real qualquer com um número finito de algarismos, quando se usa a representação decimal do número real.

O êrro de truncamento é originado pelo truncamento de uma sequência infinita de operações aritméticas após um número finito de etapas.

Os êrros de arredondamento e de truncamento podem sempre, em princípio, ser tornados arbitràriamente pequenos pelo uso de cálculos adicionais, o que não ocorre com o primeiro tipo de êrro, sendo, porisso mesmo, o mais desfavorável.

Na prática, naturalmente, aparece a questão de eficiência e é importante formular esquemas computacionais onde ao prêço de um aumento do volume de cálculos se possa tornar tão pequeno quanto desejado o êrro de arredondamento e de truncamento.

Há muitas técnicas para análise do êrro em muitos tipos de cálculos. Entretanto, na prática, são raramente usadas, por serem difíceis, exigirem muito tempo e impossíveis de efetivarem-se com cálculos manuais, se o problema prático fôr de certa complexidade.

A análise de intervalos preocupa-se com técnicas, programáveis para um computador, que permitam simultâneamente o cálculo - de um valor aproximado da solução e uma análise rigorosa e completa do êrro do resultado encontrado.

A propagação dos êrros dos dados iniciais e o efeito acumulativo dos êrros de arredondamento em qualquer sequência finita de operações aritméticas, podem ambos ser limitados pela máquina durante a execução das operações, se estas forem executadas com aritmética de intervalos arredondada em vez de usar a aritmética real.

Este trabalho está dividido em duas partes. Na parte I estudamos estruturas topológica, de ordem e algébrica no conjunto dos intervalos (Cap.1) e as funções racionais de intervalos (Cap.2).

Na parte II apresentamos exemplos numéricos utilizando intervalos: alguns na área de problemas algébricos (Cap.3) e outros na área de problemas contínuos (Cap.4).

Os programas para resolução desses problemas, redigidos em FORTRAN e processados no computador B-3500 do Centro de Computação Eletrônica da Universidade de São Paulo, têm finalidade ilustrativa, não havendo preocupação de otimização.

Nas ilustrações numéricas, com exceção do método de Hansen para inversão de matriz, usamos ternos em vez de intervalos.

Os ternos são representados por $[\underline{x},\tilde{x},\overline{x}]$, onde $\underline{x}$ e $\overline{x}$ definem o intervalo $[\underline{x},\overline{x}]$ e $\tilde{x}$ é um número real entre $\underline{x}$ e $\overline{x}$. A aritmética de ternos arredondada utiliza a aritmética real para manipular com $\tilde{x}$ e a aritmética de intervalos arredondada para operar com $[\underline{x},\overline{x}]$. Uma das razões para uso de ternos é que se o resultado final, expresso por $[\underline{x},\overline{x}]$, carecer de significação, ainda resta a informação dada por $\tilde{x}$; além disso, muitos algoritmos são formulados de modo a usar $\tilde{x}$ e o correspondente intervalo $[\underline{x},\overline{x}]$ (§ 13 - Método de Newton Generalizado).

No § 14 (Cap.4) apresentamos um método de primeira ordem para resolução de problemas iniciais em sistema de equações diferenciais ordinárias, em que o segundo membro é dado por funções racionais. Êste método pode ser estendido para os métodos de ordem k, os quais utilizam a expansão em série de Taylor. No § 15 expomos um método e respectivo programa, redigido em FORTRAN, para geração automática dos coeficientes de Taylor, necessários para os métodos de ordem k.

No apêndice A relacionamos e descrevemos sucintamente as subrotinas utilizadas nos diversos exemplos numéricos. No apêndice B colocamos a listagem dos programas descritos no apêndice A e, finalmente, no apêndice C apresentamos alguns dos resultados numéricos obtidos do processamento dos programas apresentados.

D.D.Salvetti

I N D I C E

PARTE I

Capítulo 1 - Intervalos - Estruturas topológica, de ordem e algébrica

Capítulo 2 - Funções racionais

PARTE II

Capítulo 3 - Problemas algébricos

Capítulo 4 - Problemas de equações diferenciais ordinarias

APÊNDICES

BIBLIOGRAFIA

APÊNDICE A

APÊNDICE B - Listagens

APÊNDICE C - Resultados Numéricos

Capítulo 1

§ 1 - Conjunto de Intervalos $\mathcal{J}$

Definição

Seja R a reta real e $a, b \in R$ com $a \leqslant b$. Definimos intervalo $I = [a,b]$ por:

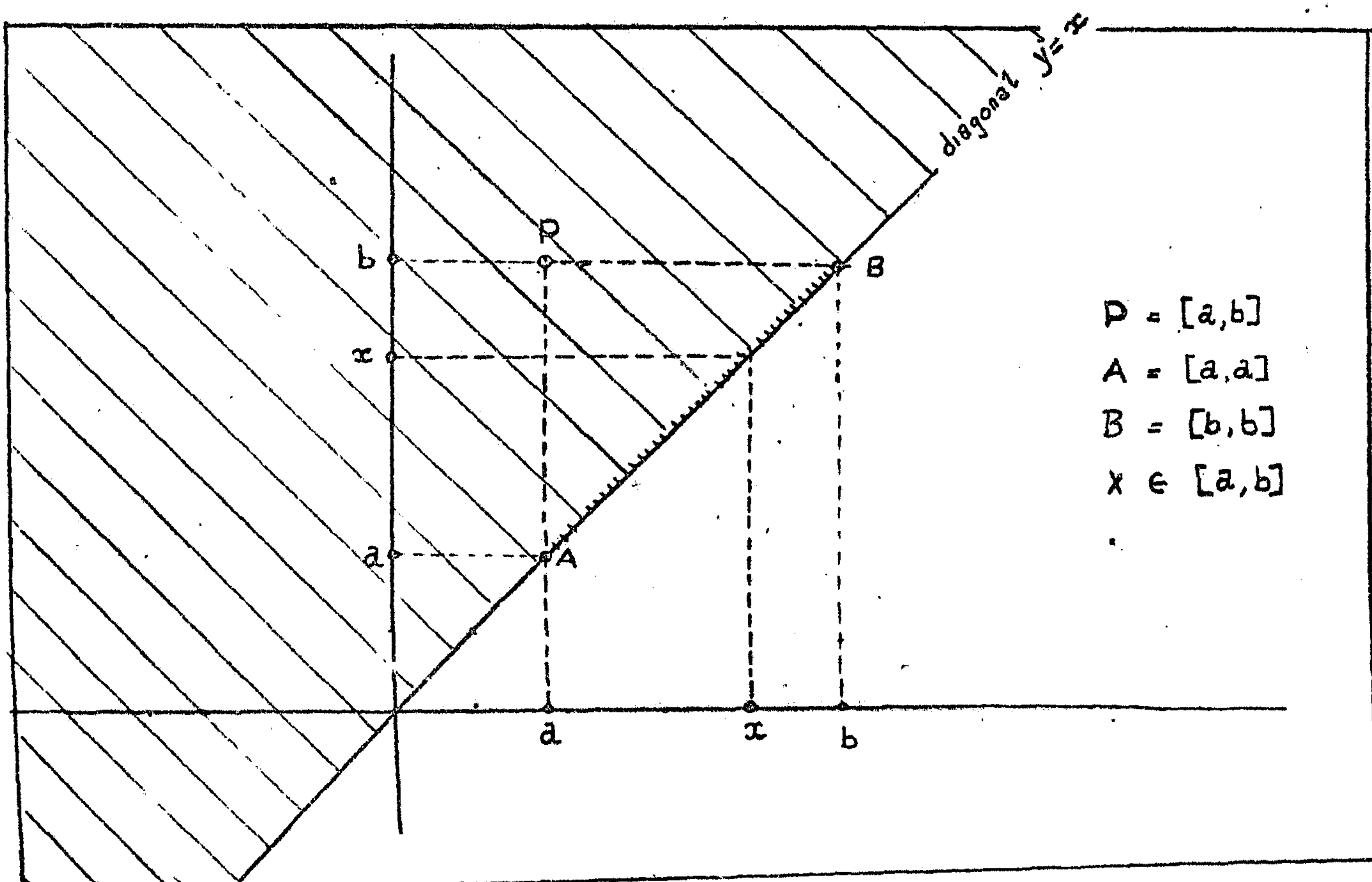
$$I = \{x \in R \mid a \leqslant x \leqslant b\}$$

Indicaremos por $\mathcal{J}$ o conjunto de todos intervalos (fechados) da reta.

Identificando-se o vetor $\begin{bmatrix} a \\ b \end{bmatrix} \in R^2$, $a < b$ com o intervalo $[a,b]$ e o vetor $\begin{bmatrix} a \\ a \end{bmatrix} = [a,a]$ com o número real a, temos

$$R \subset \mathcal{J} \subset R^2$$

Na figura abaixo a parte hachurada representa $\mathcal{J}$ como o semi-plano que contém a diagonal y=x. Tendo em vista a identificação feita, pontos acima da diagonal representam os intervalos $[a,b]$ com $a < b$ e pontos sôbre a diagonal os intervalos degenerados, em que a=b. Os números reais que pertencem ao intervalo $[a,b]$ são representados pelo segmento AB da diagonal.



Seja $I = [a,b]$ e $J = [c,d]$

1. Se $x \in I$, x é um número real entre a e b, inclusives.
2. $I \subset J \Leftrightarrow a \leq c \leq d \leq b$
3. $I \cap J = \emptyset \Rightarrow I \cup J$ não é um intervalo.
4. $I \cap J \neq \emptyset \Rightarrow I \cup J = [\min(a,c), \max(b,d)]$
5. $I = J \Leftrightarrow a = c$ e $b = d$.

§ 2 - Topologia - Aplicações Contínuas

Topologia em $\mathcal{J}$

Consideremos o espaço métrico (R^2,d), munido da distância

$$d(x,y) = \max_i(|x_i - y_i|)$$

sendo $x = (x_1,x_2)$ e $y = (y_1,y_2)$ pontos de R^2. A distância d define em R^2 a topologia habitual. Como $\mathcal{J} \subset R^2$ adotaremos como distância em $\mathcal{J}$ a distância induzida por d, isto é, se $I = [a,b]$ e $J = [c,d]$

$$d(I,J) = \max(|a-c|, |b-d|)$$

A topologia em $\mathcal{J}$ definida por esta distância é a topologia induzida pela topologia habitual de R^2.

Essa topologia em $\mathcal{J}$ induz em $R \subset \mathcal{J}$ a topologia habitual e a distância usual:

$$d(x,y) = d([x,x],[y,y]) = \max(|x-y|,|x-y|) = |x-y|$$

Topologia em $\mathcal{J} X \mathcal{J} X \mathcal{J} X \ldots X \mathcal{J}$

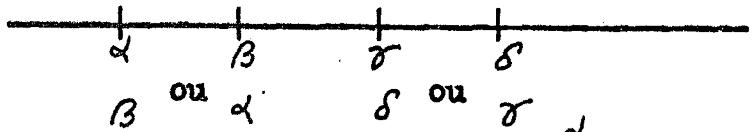
Pontos do produto cartesiano $\mathcal{J} X \mathcal{J} X \ldots X \mathcal{J}$ são vetores de intervalos $(X_1,X_2,\ldots,X_n)$. Indicando $X = (X_1,X_2,\ldots,X_n)$ e $Y = (Y_1,Y_2,\ldots,Y_n)$ dois vetores do produto cartesiano, definimos distância entre X e Y por

$$d(X,Y) = \max d(X_i,Y_i), \quad i=1,2,\ldots,n$$

Essa distância define em $\prod_{i=1}^{n} \mathcal{J}$ a topologia produto.

Lema 1

Sejam α, β, γ e δ números reais. Usaremos a notação $\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix} < \begin{smallmatrix}\gamma\\ \delta\end{smallmatrix}$ para indicar que simultâneamente $\alpha < \gamma$, $\beta < \gamma$, $\alpha < \delta$ e $\beta < \delta$



Se $\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix} < \begin{smallmatrix}\gamma\\ \delta\end{smallmatrix}$, então existe y tal que $\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix} < y < \begin{smallmatrix}\gamma\\ \delta\end{smallmatrix}$

Demonstração:

$$\left.\begin{array}{l}\left.\begin{array}{l}\alpha < \gamma\\ \beta < \gamma\end{array}\right\} \Rightarrow max(\alpha,\beta) < \gamma\\ \left.\begin{array}{l}\alpha < \delta\\ \beta < \delta\end{array}\right\} \Rightarrow max(\alpha,\beta) < \delta\end{array}\right\} \Rightarrow max(\alpha,\beta) < min(\gamma,\delta)$$

Portanto, basta tomar y entre $\max(\alpha,\beta)$ e $\min(\gamma,\delta)$.

Proposição 1

Seja $X = (X_1, X_2, \ldots, X_n)$ e $Y = (Y_1, Y_2, \ldots, Y_n)$, $X_i = [a_i, b_i]$ $Y_i = [c_i, d_i]$ e $\mathcal{E} > 0$, então $d(X,Y) < \mathcal{E}$ se e sòmente se valem simultâneamente:

(i) $\forall x \in X$, $\exists\ y \in Y$ tal que $d(x,y) < \mathcal{E}$

(ii) $\forall y \in Y$, $\exists\ x \in X$ tal que $d(x,y) < \mathcal{E}$

Demonstração:

Se $d(X,Y) < \mathcal{E}$, então $d(X_i, Y_i) < \mathcal{E}$. Portanto, $|a_i - c_i| < \mathcal{E}$ e $|b_i - d_i| < \mathcal{E}$, isto é,

$$a_i - \mathcal{E} < c_i < a_i + \mathcal{E} \tag{2.1}$$

$$b_i - \mathcal{E} < d_i < b_i + \mathcal{E} \tag{2.2}$$

Se $x \in X$, então $a_i \leq x_i \leq b_i$. (2.3)

De (2.1) e (2.3), $c_i < x_i + \mathcal{E}$. De (2.2) e (2.3) $x_i - \mathcal{E} < d_i$.

Temos então $\begin{smallmatrix}c_i\\ x_i - \mathcal{E}\end{smallmatrix} < \begin{smallmatrix}d_i\\ x_i + \mathcal{E}\end{smallmatrix}$ e pelo lema 1 existe y_i tal que $y_i \in Y_i = [c_i, d_i]$ e $|x_i - y_i| < \mathcal{E}$. Portanto, sendo $y = (y_1, y_2, \ldots, y_n)$, temos $y \in Y$ e $d(x,y) < \mathcal{E}$, o que completa a demonstração da implicação (i). Da propriedade $d(X,Y) = d(Y,X)$ e da demonstração anterior decorre a implicação (ii).

Provemos agora a recíproca, isto é, que (i) e (ii) implicam $d(X,Y)$

Tomando-se $x = (a_1, a_2, \ldots, a_n) \in X_1 x X_2 x \ldots x X_n$ existe pela (i) $y = (y_1, y_2, \ldots, y_n) \in Y_1 x Y_2 x \ldots x Y_n$ tal que $d(x,y) < \mathcal{E}$. Logo $|a_i - y_i| < \mathcal{E}$, donde $a_i - \mathcal{E} < y_i < a_i + \mathcal{E}$ .Portanto,

$$c_i < a_i + \mathcal{E} \tag{2.4}$$

Se $x = (b_1, b_2, \ldots, b_n) \in X_1 x X_2 x X_3 \ldots x X_n$ estabelecemos de modo semelhante a desigualdade

$$b_i - \mathcal{E} < d_i \tag{2.5}$$

Considerando-se $y = (c_1, c_2, \ldots, c_n)$ e $y' = (d_1, d_2, \ldots, d_n)$ pertencentes a $Y_1 x Y_2 x \ldots x Y_n$ repetimos o raciocínio feito, obtendo, através de (ii),

$$a_i < c_i + \mathcal{E} \tag{2.6}$$

$$d_i - \mathcal{E} < b_i \tag{2.7}$$

De (2.4) e (2.6), $|a_i - c_i| < \mathcal{E}$ .De (2.5) e (2.7), $|b_i - d_i| < \mathcal{E}$

Portanto, $d(X,Y) < \mathcal{E}$, q.e.d.

Observação (Moore, R.E. - Interval Analysis, pg.16)

Se n = 1 a proposição 1 assume o seguinte enunciado:

Se $I, J \in \mathcal{F}$, $\mathcal{E} > 0$, então $d(I,J) < \mathcal{E}$ ($\leq \mathcal{E}$)

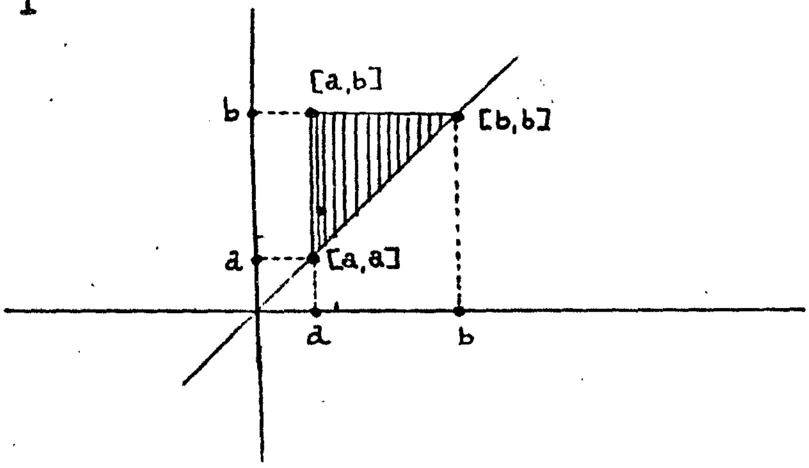
se e sòmente se as seguintes afirmações valem simultâneamente:

(i) $\forall x \in I, \exists\ y \in J$ tal que $|x - y| < \mathcal{E}$ ($\leq \mathcal{E}$)

(ii) $\forall y \in J, \exists\ x \in I$ tal que $|x - y| < \mathcal{E}$ ($\leq \mathcal{E}$)

Continuidade em $\mathcal{F}_I$

Seja $\mathcal{F}_I = \mathcal{F}_{[a,b]} = \{ [x,y] \mid a \leq x \leq y \leq b \}$, o conjunto dos intervalos contidos em $I = [a,b]$. O triângulo hachurado na figura abaixo representa $\mathcal{F}_I$.



A aplicação $F : \mathcal{J}_I \longrightarrow \mathcal{J}$ é contínua em $X \in \mathcal{J}_I$ se dado $\varepsilon > 0$ arbitrário, existe $\delta(\varepsilon)$ tal que para todo $Y \in \mathcal{J}_I$ que satisfaça $d(X,Y) \leq \delta(\varepsilon)$, ocorre $d(F(X),F(Y)) \leq \varepsilon$.

Continuidade em $\mathcal{J}_{I_1} x \mathcal{J}_{I_2} x \ldots x \mathcal{J}_{I_n}$

A aplicação $F : \prod_{i=1}^{n} \mathcal{J}_{I_i} \longrightarrow \mathcal{J}$ é contínua em $X=(X_1,X_2,\ldots,X_n)$ se dado $\varepsilon > 0$ arbitrário existe $\delta(\varepsilon)$ tal que para todo $Y=(Y_1,Y_2,\ldots,Y_n)$ pertencente a $\prod_{i=1}^{n} \mathcal{J}_{I_i}$ que satisfaça à condição $d(X,Y) \leq \delta(\varepsilon)$, ocorre $d(F(X),F(Y)) \leq \varepsilon$.

Recordemos que $d(F(X),F(Y)) = \max d(X_i,Y_i)$, $i=1,2,\ldots,n$, enquanto que $d(X,Y)$ é a distância entre dois intervalos. Observamos que indicamos com a mesma notação as distâncias em $\mathcal{J}$ ou em $\mathcal{J} x \mathcal{J} x \ldots x \mathcal{J}$.

§ 3 - Ordem - Aplicações Monotônicas

Relação de Ordem Parcial

A relação de inclusão $I \subset J$ define uma relação de ordem parcial em $\mathcal{J}$. No produto cartesiano $\mathcal{J} x \mathcal{J} x \ldots x \mathcal{J}$ uma relação de ordem parcial é definida da seguinte maneira

$$X = (X_1,X_2,\ldots,X_n) \prec Y = (Y_1,Y_2,\ldots,Y_n)$$

se e só se $X_i \subset Y_i$, $i = 1,2,\ldots,n$

A relação de ordem $\prec$,por abuso de linguagem, será referida daqui por diante pelo símbolo $\subset$.

Aplicação Monotônica

Seja $F : \mathcal{J}_{I_1} x \mathcal{J}_{I_2} x \ldots x \mathcal{J}_{I_n} \longrightarrow \mathcal{J}_{J_1} x \mathcal{J}_{J_2} x \ldots x \mathcal{J}_{J_m}$, X e Y elementos de $\mathcal{J}_{I_1} x \mathcal{J}_{I_2} x \ldots x \mathcal{J}_{I_n}$. A função de intervalos F é monotônica se $X \subset Y$ implica $F(X) \subset F(Y)$.

Composição de Aplicações Monotônicas

Proposição 1

Seja $F : \mathcal{F}_{I_1} \times \mathcal{F}_{I_2} \times \cdots \times \mathcal{F}_{I_m} \rightarrow \mathcal{F}_{J_1} \times \mathcal{F}_{J_2} \times \cdots \times \mathcal{F}_{J_m}$ e

$G : \mathcal{F}_{K_1} \times \mathcal{F}_{K_2} \times \cdots \times \mathcal{F}_{K_\ell} \rightarrow \mathcal{F}$ aplicações monotônicas.

Se $F(\mathcal{F}_{I_1} \times \mathcal{F}_{I_2} \times \cdots \times \mathcal{F}_{I_m}) \subset \mathcal{F}_{K_1} \times \mathcal{F}_{K_2} \times \cdots \times \mathcal{F}_{K_m}$, a composta $G \circ F$ é monotônica.

Demonstração:

Consideremos X e Y pertencentes a $\mathcal{F}_{I_1} \times \mathcal{F}_{I_2} \times \cdots \times \mathcal{F}_{I_m}$. Se $X \subset Y$, entã $F(X) \subset F(Y)$ e $F(X)$, $F(Y)$ pertencem a $\mathcal{F}_{K_1} \times \mathcal{F}_{K_2} \times \cdots \times \mathcal{F}_{K_\ell}$. Logo $G(F(X)) \subset G(F(Y))$. Portanto, para todo $X, Y \in \mathcal{F}_{I_1} \times \mathcal{F}_{I_2} \times \cdots \times \mathcal{F}_{I_m}$ que satisfaça à inclusão $X \subset Y$, temos $G \circ F(X) \subset G \circ F(Y)$

Corolário 1.1

A composição de um número finito de aplicações monotônicas é monotônica.

§ 4 - Restrição e extensão de aplicações

Definição

Seja $F: \mathcal{F}_I \longrightarrow \mathcal{F}$. A restrição aos reais de F é a aplicação $f: I \rightarrow \mathcal{F}$ definida por

$$f(x) = F([x,x]) = F(x)$$

Generalizando: seja $F: \mathcal{F}_{I_1} \times \mathcal{F}_{I_2} \times \ldots \times \mathcal{F}_{I_n} \longrightarrow \mathcal{F}^m$. Sua restrição aos reais é a aplicação $f: I_1 \times I_2 \times \ldots \times I_n \rightarrow \mathcal{F}^m$ definida por

$$f(x_1,x_2,\ldots,x_n) = F([x_1,x_1],[x_2,x_2],\ldots,[x_n,x_n]) = F(x_1,x_2,\ldots,x_n)$$

Definição

Sejam $f: I_1 \times I_2 \times \ldots \times I_n \longrightarrow \mathcal{F}^m$ e

$F: \mathcal{F}_{I_1} \times \mathcal{F}_{I_2} \times \ldots \times \mathcal{F}_{I_n} \rightarrow \mathcal{F}^m$

Dizemos que F é uma extensão de f se f fôr uma restrição aos reais de F.

Definição

Uma extensão importante de uma aplicação contínua $f\colon I \longrightarrow \mathcal{J}$ é a função de intervalos $\overline{f} : \mathcal{J}_I \longrightarrow \mathcal{J}$ chamada extensão unida e definida por:

$$\overline{f}(X) = \bigcup_{x \in X} f(x)$$

Mostremos que a extenção unida está bem definida provando através das proposições 1 e 2 que $\overline{f}(X) \in \mathcal{J}$, isto é, é um intervalo.

Proposição 1

Se $f\colon I \longrightarrow \mathcal{J}$ é contínua, $f(x) = [f_1(x), f_2(x)]$, então f_1 e f_2 são aplicações de I em R contínuas.

Demonstração:

Dado $\varepsilon > 0$ aarbitrário e $x \in I$, da continuidade de $f(x)$, existe $\delta > 0$, tal que para todo $x' \in I$ que satisfaça $|x-x'| < \delta$, ocorre $d(f(x), f(x')) < \varepsilon$

Portanto,

$$|f_1(x) - f_1(x')| < \varepsilon \qquad (1)$$

$$e \quad |f_2(x) - f_2(x')| < \varepsilon \qquad (2)$$

De (1) obtemos a continuidade de f_1 e de (2) a continuidade de f_2.

Proposição 2

Se $f\colon I \longrightarrow \mathcal{J}$ é contínua, $f(x) = [f_1(x), f_2(x)]$, então

$$\overline{f}(X) = [\min_X f_1(x), \max_X f_2(x)] \in \mathcal{J}$$

Demonstração:

Se $f(x)$ é contínua, pela proposição 1 são contínuas $f_1(x)$ e $f_2(x)$ tais que $f(x) = [f_1(x), f_2(x)]$. Da continuidade de $f_i(x)$ no intervalo fechado X, $f_i(x)$ assume todos valores entre $\min_X f_i(x)$ e $\max_X f_i(x)$ enquanto x percorre X.

Seja $a = \min_X f_1(x)$ e $b = \max_X f_2(x)$

Por definição $\overline{f}(X) = \bigcup_{x \in X} [f_1(x), f_2(x)]$

Para todo y de $\overline{f}(X)$, existe x pertencente a X tal que $y \in [f_1(x), f_2(x)] \subset [a,b]$, donde $y \in [a,b]$ e portanto, $\overline{f}(X) \subset [a,b]$ (1)

Seja $x_0 \in X$. Consideremos os intervalos

$$A_1 = [a, f_1(x_0)]$$

$$A_2 = [f_1(x_0), f_2(x_0)]$$
$$A_3 = [f_2(x_0), b]$$

A_2 está contido em $\overline{f}(X)$ porque $A_2 = f(x_0)$.

A_1 está contido em $\overline{f}(X)$ porque $f_1(x)$ assume todos valores entre $a = \min f_1(x)$ e $f_1(x_0)$ quando x percorre X.

A_3 está contido em $\overline{f}(X)$ por razão semelhante.

Logo, a reunião $A_1 \cup A_2 \cup A_3 \subset \overline{f}(X)$. Porém $A_1 \cup A_2 \cup A_3 = [a,b]$

Portanto, $[a,b] \subset \overline{f}(X)$. (2)

De (1) e (2) segue $\overline{f}(X) = [a,b]$, *q.e.d.*

Definição

Seja $f: I_1 x I_2 x \ldots x I_n \longrightarrow \mathcal{F}$ contínua.

A aplicação $\overline{f}: \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n} \longrightarrow \mathcal{F}$ definida por

$$\overline{f}(X) = \bigcup_{x \in \Pi X_i} f(x_1, x_2, \ldots, x_n) = \bigcup_{x \in X} f(x)$$

é chamada extensão unida. Quando n= 1 re-encontramos a definição.

$\overline{f}(X)$ está bem definida pois valem as proposições:

Proposição 1'

Se $f: I_1 x I_2 x \ldots x I_n \rightarrow \mathcal{F}$ é contínua, $f(x) = [f_1(x), f_2(x)]$,então f_1 e f_2 são aplicações de $I_1 x I_2 x \ldots x I_n$ em R contínuas.

Demonstração:

Idêntida a da proposição 1.

Proposição 2'

Se $f: I_1 x I_2 x \ldots x I_n \longrightarrow \mathcal{F}$ é contínua, $f(x) = [f_1(x), f_2(x)]$,então

$$\overline{f}(X) = [\min_{\Pi X_i} f_1(x1, x_2, \ldots, x_n), \max_{\Pi X_i} f_2(x_1, x_2, \ldots, x_n)]$$

Demonstração:

Pela proposição 1', f_1 e f_2 são contínuas e portanto levam $I_1 x I_2 x \ldots x I_n$ que é compacto conexo num conjunto compacto conexo da reta, isto é, num intervalo fechado limitado. Portanto f_i assume em $I_1 x \ldots x I_n$ um máximo, um mínimo e todos os valores intermediários.

O resto da demonstraçao é idêntida a da proposição 2, lembrando que nem convenção $x \in X$ significa $x_i \in X_i$, i=1,2,...,n.

Proposição 3

Se $f: I_1 x I_2 x \ldots x I_n \longrightarrow \mathcal{F}$ é contínua, então

$\bar{f}: \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n} \longrightarrow \mathcal{F}$ é contínua.

Demonstração:

Sendo f contínua num compacto, f com valores num espaço métrico é uniformemente contínua. Seja $\varepsilon > o$ arbitrário. Da continuidade uniforme existe $\delta > o$ tal que para todo $x \in I$ e $x' \in I$ satisfazendo $d(x,x') < \delta$ se verifica $d(f(x),f(x')) < \varepsilon$.

Seja X,X' elementos de $\mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n}$ tais que $d(X,X') < \delta$

Pela proposição 1 do § 2:

(i) $\forall x \in X, \exists x' \in X'$ tal que $d(x,x') < \varepsilon$

Consideremos $t \in \bar{f}(X)$, então existe $x \in X$ tal que $t \in f(x)$. Por (i), existe $x' \in X'$ tal que $d(x,x') < \delta$. Por hipótese $d(f(x),f(x')) < \varepsilon$ Como $t \in f(x)$, pela proposição 1 do § 2 existe $t' \in f(x')$ e portanto $t' \in \bar{f}(X')$, tal que $d(t,t') < \varepsilon$. Temos então

$$\forall t \in \bar{f}(X), \exists t' \in \bar{f}(X') \text{ tal que } d(t,t') < \varepsilon \quad (1)$$

Repetindo para $x' \in X'$, provamos anàlogamente que

$$\forall t' \in \bar{f}(X'), \exists t \in \bar{f}(X) \text{ tal que } d(t,t') < \varepsilon \quad (2)$$

De (1) e (2) concluímos, pela proposição 1 do § 2, que

$$d(\bar{f}(X),\bar{f}(X')) < \varepsilon$$

o que completa a prova de que $\bar{f}$ é contínua.

Proposição 4

A extensão unida é uma aplicação monotônica

Demonstração:

Sejam $f: I_1 x I_2 x \ldots x I_n \longrightarrow \mathcal{F}$

$\bar{f}: \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n} \longrightarrow \mathcal{F}$

e $X,X' \in \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n}$ tais que $X \subset X'$

Provemos que $\bar{f}(X) \subset \bar{f}(X')$

Seja $y \in \overline{F}(X)$. Existe $x \in X$ tal que $y \in f(x)$. Mas $x \in X'$, dond $f(x) \subset \overline{F}(X')$. Logo $y \in \overline{F}(X')$, q.e.d.

Composição de Extensões

Seja $f: I \longrightarrow \mathcal{F}$ e $F: \mathcal{F}_I \longrightarrow \mathcal{F}$ uma sua extensão.

Consideremos o seguinte diagrama, onde i indica inclusão.

$$\begin{tikzcd}
\mathcal{F}_I \arrow[r, "F"] & \mathcal{F} \\
I \arrow[u, "i"] \arrow[r, "f"] & f(I) \arrow[u, "i"]
\end{tikzcd}$$

Observemos que dizer F é uma extensão de f é equivalente a dizer que o diagrama é comutativo, isto é, $F \circ i = i \circ f$

Proposição 5

A composição de duas extensões é uma extensão da composta das respectivas restrições aos reais.

Demonstração:

Sejam F e G extensões de f e g, respectivamente. Se as compostas referidas na proposição estão definidas, temos $f(I) \subset J$ e $F(\mathcal{F}_I) \subset \mathcal{F}_J$

O diagrama

$$\begin{tikzcd}
\mathcal{F}_I \arrow[r, "F"] & F(\mathcal{F}_I) \arrow[r] & \mathcal{F}_J \arrow[r, "G"] & \mathcal{F} \\
I \arrow[u, "i"] \arrow[r, "f"] & f(I) \arrow[u, "i"] \arrow[r] & J \arrow[u, "i"] \arrow[r, "g"] & f(J) \arrow[u, "i"]
\end{tikzcd}$$

é comutativo porque os três diagramas parciais são comutativos. Logo $F \circ G$ é uma extensão de $f \circ g$.

Observação

Aplicando-se repetidas vêzes a proposição acima vemos que composições mais complicadas de extensões é uma extensão das compostas envol - vidas na composição das respectivas restrições aos reais, supondo-se, lògicamente, que as condições para a definição das composições estejam satisfeitas.

Proposição 6

Seja $F: \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n} \longrightarrow \mathcal{F}$ uma extensão monotônica de $f: I_1 x I_2 x \ldots x I_n \longrightarrow \mathcal{F}$, então para todo $X \in \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n}$, $\bar{F}(X) \subset F(X)$

Demonstração:

$y \in \bar{F}(X) = \bigcup_{x \in X} f(x)$ implica que existe $x \in X$ tal que $y \in f(x)$.

Sendo F uma extensão monotônica de f, $f(x) = F([x,x]) \subset F(X)$, donde $y \in F(X)$, q.e.d.

Corolário 6.1

$\bar{F}(X) = \bigcap_{F} F(X)$ onde F percorre o conjunto das extensões monotônicas de f.

Demonstração:

Pela proposição 6 $\bar{F}(X) \subset F(X)$ para tôda extensão monotônica F.

Logo,

$$\bar{F}(X) \subseteq \bigcap_{F} F(X)$$

Por outro lado como $\bar{F}$ é uma particular extensão monotônica temos $\bar{F}(X) \subset \bigcap_{F} F(X)$.

Observação

No sentido do corolário 6.1 podemos dizer que a extensão unida é a "menor" das extensões monotônicas.

§ 5 - Aritmética de Intervalos

Definição

Indiquemos por $\ast$ uma das operações + - x / sôbre números reais. Vamos prolongar $\ast : RxR \longrightarrow R$ a $\ast : \mathcal{F} x \mathcal{F} \longrightarrow \mathcal{F}$ definindo:

$$X \ast Y = \{ x \ast y \mid x \in X, y \in Y \}, \; X, Y \in \mathcal{F}$$

Se $\ast$ fôr a divisão, suporemos a restrição $Y \not\ni 0$

Seja $f: X \times Y \longrightarrow R$ onde $f(x,y) = x \ast y$.

Como f é contínua podemos tomar sua extensão unida $\bar{f}: \mathcal{F}_X x \mathcal{F}_Y \longrightarrow \mathcal{F}$
Pela definição acima verificamos que

$$X \ast Y = \bar{f}(X,Y) \in \mathcal{F}$$

Portanto a operação $\ast$ sôbre intervalos está bem definida.

Como consequências de $X \ast Y = \bar{f}(X,Y)$ temos:

(1) $X \ast Y = [\min\limits_{\substack{x \in X \\ y \in Y}}(x \ast y),\ \max\limits_{\substack{x \in X \\ y \in Y}}(x \ast y)]$

(Aplicação da proposição 2' do § 4)

(2) $[a,b] + [c,d] = [a+c, b+d]$

$[a,b] - [c,d] = [a-d, b-c]$

$[a,b] \times [c,d] = [\min(ac,ad,bc,bd), \max(ac,ad,bc,bd)]$

$[a,b] / [c,d] = [a,b] \times [1/d, 1/c]$ se $[c,d] \not\ni 0$

(consequência de (1))

<u>Propriedades da adição e da multiplicação</u>

1. Associatividade

$$I + (J + K) = (I + J) + K$$

$$I(JK) = (IJ)K$$

2. Comutatividade

$$I + J = J + I$$

$$IJ = JI$$

Estas propriedades decorrem diretamente da definição das operações.

3. Elementos Neutros

Os números reais 0 e 1 são os únicos elementos neutros para a adição e a multiplicação de intervalos, respectivamente.

De fato, $0 + J = J + 0 = J$. Se $[a,b]$ fôr um elemento neutro da adição teremos:

$$[a,b] + [0,0] = [0,0]$$

ou $[a,b] = [0,0]$ (unicidade)

Para o caso da multiplicação temos uma verificação análoga.

Não existe oposto nem inverso, a menos que o intervalo seja degenerado, e neste caso, o oposto e o inverso são únicos e dados por $[-a,-a$ e $[1/a,1/a]$,respectivamente.

Suponhamos que $[a,b]$ tenha oposto e seja êle $[x,y]$. Então $[a,b] + [x,y] = [o,o]$ nos leva ao sistema $a+x=o$ e $b+y=o$ cuja única solução é $x=-a$ e $y=-b$. Portanto, $a \leq b$ e $-a \leq -b$, o que implica $a = b$, isto é, o intervalo $[a,b]$ é degenerado e o oposto de $[a,a]$ é $[-a,-a]$.

Do mesmo modo, admitamos que $[a,b]$ tenha por inverso o intervalo $[x,y]$. Então $[a,b].[x,y]=[1,1]$. Supondo-se $a > o$, $b > o$, $x > o$ e $y > o$ $[a,b].[x,y] = [ax,by]$ e $[ax,by]= [1,1]$ conduz ao sistema $ax=1$ e $by=1$ cuja única solução é $x = 1/a$ e $y = 1/b$. Logo $a \leq b$ e $1/a \leq 1/b$, o que implica $a = b$, isto é, o intervalo $[a,b]$ é degenerado e o inverso de $[a,a]$ é $[1/a,1/a]$.

Para as demais combinações de sinais, relacionadas no quadro abaixo demonstra-se anàlogamente o mesmo resultado:

a	b	x	y
+	+	−	+
−	+	+	+
−	+	+	+
−	+	−	+

4. Propriedade semi-distributiva

$$I(J + K) \subset IJ + IK$$

Verifiquemos que esta inclusão está sempre satisfeita.

$\forall\ w \in I(J + K)$, $\exists\ x \in I$ e $u \in J + K$ tais que $w = x.u$ Por outro lado se $u \in J+K$ existem $y \in J$ e $z \in K$ tais que $u = y + z$. Portanto, $w = x.u = x.(y+z) = xy + xz$. Como $xy \in IJ$ e $xz \in IK$, concluimos que $w \in IJ + IK$, q.e.d.

Um simples contra-exemplo mostra que a inclusão em sentido contrário nem sempre está satisfeita:

$[1,2].[-2,2] + [1,2].[1,2] = [-4,4] + [1,4] = [-3,8]$

$[1,2]([-2,2] + [1,2]) = [1,2].[-1,4] = [-2,8]$

Entretanto, há dois casos em que é válida a propriedade distributiva:

(a) Se $I = [a,a]$, então $I(J+K) = IJ+IK$

A prova é imediata.

Vejamos um exemplo numérico. Sejam $I = [2,2]$, $J = [-3,4]$ e $K = [0,1]$

$I(J+K) = [2,2]([-3,5]) = [-6,10]$

$IJ+IK = [2,2].[-3,4] + [2,2].[0,1] = [-6,8]+[0,2] = [-6,10]$

(b) Se $JK > 0$, então para todo I temos $I(J+K) = IJ + IK$

A notação $JK > 0$ indica que todos números reais do intervalo JK são positivos. $JK > 0$ ocorre se $J > 0$ e $K > 0$ ou $J < 0$ e $K < 0$.

Sejam $I=[a,b]$, $J=[c,d]$ e $K=[e,f]$. Suponhamos $a > 0$, $b > 0$, $c > 0$, $d > 0$, $e > 0$ e $f > 0$. Neste caso teremos:

$[a,b]([c,d]+[e,f])=[a,b]([c+e,d+f])=[a(c+e),b(d+f)]=$

$=[ac+ae,bd+bf] = [ac,bd] + [ae,bf] = [a,b].[c,d] + [a,b].[e,f]$

Para as demais combinações de sinais, relacionadas no quadro abaixo, prova-se a propriedade de modo análogo:

a	b	c	d	e	f
−	+	+	+	+	+
+	+	−	−	−	−
−	+	−	−	−	−

Exemplo numérico. Sejam $I=[-1,2]$, $J = [1,2]$ e $K = [2,4]$

$I(J+K) = [-1,2]([1,2]+[2,4]) = [-1,2].[3,6] = [-6,12]$

$IJ+IK = [-1,2].[1,2]+[-1,2].[2,4] = [-2,4]+[-4,8] = [-6,12]$

Relação de Ordem e as Operações Aritméticas

Além da relação de ordem parcial em $\mathcal{I}$ definida pela inclusão e denotada por $\subset$ vamos considerar uma nova relação de ordem estrita parcial que denotaremos $<$ definida por:

$$[a,b] < [c,d] \quad \text{se} \quad b < c$$

Da monotonia da extensão unida, relativamente à relação de ordem $\subset$, decorrem as seguintes propriedades de compatibilidade com as operações aritméticas sôbre intervalos:

Se $I \subset K$ e $J \subset L$, então

$$I + J \subset K + L$$
$$I - J \subset K - L$$
$$IJ \subset KL$$
$$I/J \subset K/L \qquad \text{se} \quad L \not\ni 0$$

A relação de ordem $<$ possui as seguintes propriedades de compatibilidade:

$$I < K \quad e \quad J < L \Rightarrow I + J < K + L$$

$$0 < I < K \quad e \quad 0 < K < L \Rightarrow IJ < KL$$

São consequências imediatas das definições.

Continuidade das operações aritméticas sôbre intervalos

Proposição 1

As quatro operações aritméticas sôbre intervalos são contínuas aonde definidas.

Demonstração:

Seja $*$ uma das quatro operações e provemos a continuidade da operação no ponto (X_0,Y_0). No caso da divisão suporemos $Y_0 \not\ni 0$.

Seja $\varepsilon > 0$ e $I = X_0 + [-\varepsilon, \varepsilon]$, $J = Y_0 + [-\varepsilon, \varepsilon]$.

No caso da divisão podemos tomar $\varepsilon > 0$ suficientemente pequeno de modo que resulte $J \not\ni 0$.

Como $\mathcal{F}_I \times \mathcal{F}_J$ é uma vizinhança de (X_o,Y_o) inteiramente contida no campo de definição de $*$, basta provar a continuidade nessa vizinhança. Entretanto, em $\mathcal{F}_I \times \mathcal{F}_J$ a operação $*$ coincide com a extensão unida da aplicação contínua $*: I \times J \longrightarrow R$. Portanto, a operação $*$ prolongada a $\mathcal{F}_I \times \mathcal{F}_J$ é contínua, q.e.d.

§ 6 - Módulo e Comprimento de Intervalos
Condição de Lipchitz

Definição

Definiremos a aplicação módulo $|\ \ | : \mathcal{F} \longrightarrow R$ por

$$|[a,b]| = \max(|a|,|b|)$$

Podemos estender a definição, definindo

$$|\ \ | : \mathcal{F}^m \longrightarrow R$$

por $|X| = \max |X_i|$, $i=1,2,\dots,m$ onde $X=(X_1,X_2,\dots,X_m) \in \mathcal{F}^m$

Definição

Definiremos a aplicação comprimento $w: \mathcal{F} \longrightarrow R$ por

$$w([a,b]) = b - a$$

Podemos estender a definição, definindo

$$w : \mathcal{F}^m \longrightarrow R$$

por $w(X) = \max w(X_i)$, $i=1,2,\dots,m$ onde $X=(X_1,X_2,\dots,X_m) \in \mathcal{F}^m$.

Propriedades

1. $|I| = d(I,o)$
2. Se $I = [a,b]$, então $w(I) = 2|I|$

   $\therefore$ $w(I) = b-a \leq |b| + |a| \leq 2\max(|a|,|b|) = 2.|I|$
3. Se $I \subset J$, então existe um único intervalo E tal que $J = I + E$, com $E \ni o$ e $|E| = d(I,J)$

   Seja $I = [a,b]$ e J c,d. A equação $J = I + x,y$ conduz ao sistema $a+x=c$ e $b+y=d$ cuja única solução é $x=c-a$ e $y=d-b$. Logo,

$E = [x,y] = [c-a, d-b]$. Como $I \subset J$, $c \leq a \leq b \leq d$, de onde decorre $x \leq o$ e $y \geq o$, donde $o \in E$. Pelas definições de módulo e distância entre dois intervalos vemos que

$$|E| = \max(|a-c|, |b-d|) = d(I,J)$$

4. $w(I+J) = w(I) + w(J)$

Se $I = [a,b]$ e $J = [c,d]$, então $I+J = [a+c, b+d]$.

$w(I+J) = b+d-a-c = (b-a) + (d-c) = w(I) + w(J)$

5. $w(\alpha J) = |\alpha| w(J)$

Seja $J = [c,d]$ e α um número real.

$\alpha[c,d] = \begin{cases} [\alpha c, \alpha d] \\ [\alpha d, \alpha c] \end{cases}$. Portanto $w(\alpha J) = \alpha d - \alpha c$ ou $\alpha c - \alpha d$, isto é,

$w(\alpha J) = \alpha(d-c)$ ou $(-\alpha)(d-c)$. Portanto, $w(\alpha J) = |\alpha| w(J)$

Consequências:

5.1 $w(-J) = w(J)$

5.2 $w(I-J) = w(I) + w(J)$

De fato, $w(I-J) = w(I+(-J)) = w(I) + w(-J) = w(I) + w(J)$

Condição de Lipchitz

Seja $f: I_1 x I_2 x \ldots x I_n \longrightarrow \mathcal{F}$. Dizemos que f satisfaz uma <u>condição de Lipchitz</u> se existe $L > o$ tal que:

$$d(f(x), f(x')) \leq L d(x,x')$$

para todo $x, x' \in \prod I_i$ e L independente de x, x'.

<u>Definição</u>

Seja $F: \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n} \longrightarrow \mathcal{F}$. Dizemos que F satisfaz uma <u>condição de Lipchitz</u> se existe $L > o$ tal que

$$w(F(X)) \leq L w(X)$$

onde L independe de $X = (X_1, X_2, \ldots, X_n)$

Proposição 1

Seja $f: I_1 x I_2 x \ldots x I_n \rightarrow \mathcal{F}$ e f_1, f_2 tais que $f(x) = [f_1(x), f_2(x)]$

Se $f \in \mathcal{C}^1$, isto é, se $f_1, f_2 \in C^1$, então f satisfaz uma condição de Lipchitz.

Demonstração:

Pelo teorema da média temos:

$$f_1(x) - f_1(x') = \sum_j \frac{\partial f_1}{\partial x_j}(x + \theta_1(x'-x)).(x_j - x_j') \quad \text{onde} \quad 0 \leq \theta_1 \leq 1.$$

$$|f_1(x) - f_1(x')| \leq \sum_j \left|\frac{\partial f_1}{\partial x_j}(x + \theta_1(x'-x))\right|.|x_j - x_j'| \leq$$

$$\leq \left\{\sum_j \left| \frac{\partial f_1}{\partial x_j}(x + \theta_1(x'-x))\right|\right\}.\max_j |x_j - x_j'| \leq K_1 d(x,x') \quad \text{onde}$$

$$K_1 = \max_{z \in \prod I_i} \sum_j \left|\frac{\partial f_1}{\partial x_j}(z)\right|$$

Anàlogamente:

$$|f_2(x) - f_2(x')| \leq K_2 d(x,x')$$

Portanto $d(f(x), f(x')) \leq L d(x,x')$ onde

$$L = \max(K_1, K_2) \qquad \text{q.e.d.}$$

Proposição 2

Se $F: \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_m} \longrightarrow \mathcal{F}$ é monotônica e satisfaz uma condição de Lipchitz com constante L, então sua restrição aos reais f também satisfaz uma condição de Lipchitz com mesma constante.

Demonstração:

Sejam $x = (x_1, x_2, \ldots, x_n)$, $x' = (x_1', x_2', \ldots, x_n')$ tais que $x_j, x_j' \in I_j$.

Construamos $X = (X_1, X_2, \ldots, X_n)$ como:

$$X_j = \begin{cases} [x_j, x_j'] & \text{se } x_j \leq x_j' \\ [x_j', x_j] & \text{se } x_j' \leq x_j \end{cases}$$

Como $x, x' \in X$, por construção, temos: $f(x), f(x') \subset F(X)$

Portanto, $|f_1(x)-f_1(x')| \leq w(F(X)) \leq Lw(X) = L \max_j |x_j - x_j'| = Ld(x,x')$

Análogamente, $|f_2(x)-f_2(x')| \leq Ld(x,x')$

Logo, $d(f(x),f(x')) = \max(|f_1(x)-f_1(x')|, |f_2(x)-f_2(x')|) \leq Ld(x,x')$

Proposição 3

Seja $f: I_1 x I_2 x \ldots x I_n \longrightarrow R$ (a valores reais) satisfazendo a uma condição de Lipchitz. Então $\overline{f}$ satisfaz uma condição de Lipchitz.

Demonstração:

$$w(\overline{f}(X)) = w[\min_{x \in X} f(x), \max_{x \in X} f(x)] = w[f(x'), f(x'')]$$

onde x' e x'' são pontos de mínimo e de máximo em X.

Logo, $w(\overline{f}(X)) = |f(x')-f(x'')| \leq L \max_j |x_j' - x_j''| \leq L\, w(X)$, q.e.d.

Lema 1

Sejam A_i, B_i intervalos tais que $A_i \supset B_i$ e $\cup A_i$ e $\cup B_i$ são intervalos. Então $d(\cup A_i, \cup B_i) \leq \max d(A_i, B_i)$

Demonstração:

Seja $A_i = [a_i, b_i]$ e $B_i = [c_i, d_i]$. Temos $a_i \leq c_i \leq d_i \leq b_i$.

Como $\cup A_i \in \mathcal{F}$, então $\cup A_i = [\min a_i, \max b_i]$.

Anàlogamente, $\cup B_i = [\min c_i, \max d_i]$

$d(\cup A_i, \cup B_i) = \max(|\min a_i - \min c_i|, |\max b_i - \max d_i|)$

Por outro lado temos que $|\min a_i - \min c_i| \leq |\min a_i - c_i|$, $\forall i$

Seja $a_j = \min a_i$. Então $|\min a_i - \min c_i| \leq |a_j - c_j| \leq d(A_j, B_j)$ $\leq \max d(A_i, B_i)$

Do mesmo modo mostramos que $|\max b_i - \max d_i| \leq \max d(A_i, B_i)$

Portanto, $d(\cup A_i, \cup B_i) = \max d(A_i, B_i)$, q.e.d.

Proposição 4

Seja $h: I_1^{k_1} x I_2^{k_2} x \ldots x I_n^{k_n} \longrightarrow \mathcal{F}$ satisfazendo uma condição de Lipchitz. Seja $F: \mathcal{F}_{I_1} x \mathcal{F}_{I_2} x \ldots x \mathcal{F}_{I_n} \longrightarrow \mathcal{F}$ definida por:

$$F(X_1,X_2,\dots,X_n) = \bar{h}(\underbrace{X_1,\dots,X_1}_{k_1},\dots,\underbrace{X_n,\dots,X_n}_{k_n})$$

Seja f a restrição aos reais de F e $\bar{f}$ sua extensão unida.

Então, $F(X) = \bar{f}(X) + E(X)$ onde

$$E(X) \ni o \quad e \quad w(E(X)) \leq kw(X)$$

isto é, $E(X)$ satisfaz uma condição de Lipchitz.

Demonstração:

Primeiramente verifiquemos que o enunciado é coerente:

h satisfaz condição de Lipchitz $\Rightarrow$ h é contínua $\Rightarrow$ $\bar{h}$ é monotô - nica e contínua $\Rightarrow$ F é monotônica e contínua $\Rightarrow$ f é contínua $\Rightarrow$ $\bar{f}$ é monotônica e contínua e $\bar{f}(X) \subset F(X)$.

Como $F(X) \supset \bar{f}(X)$, vem $F(X) = \bar{f}(X) + E(X)$ com $E(X) \ni o$.

Provemos que E é Lipchitziana.

$w(E(X)) \leq 2|E(X)| = 2d(F(X),\bar{f}(X))$.

Para simplificar a notação suponhamos $n=1$ e $k_1 = 2$. Teremos:

$$F(X) = \bar{h}(X,X) = \bigcup_{\substack{x\in X\\ y\in X}} h(x,y)$$

$$\bar{f}(X) = \bigcup_{x\in X} f(x) = \bigcup_{x\in X} F(x) = \bigcup_{x\in X} \bar{h}(x,x) = \bigcup_{x\in X} h(x,x)$$

Mas, $d(F(X),\bar{f}(X)) = d(\bigcup_{x\in X}(\bigcup_{y\in X} h(x,y)), \bigcup_{x\in X} h(x,x))$

Como $\bigcup_{y\in X} h(x,y) \supset h(x,x)$ para todo $x \in X$ vem pelo Lema 1

$$d(F(X),\bar{f}(X)) \leq \max_{x\in X} d(\bigcup_{y\in X} h(x,y),h(x,x)) \leq$$

$$\leq \max_{x\in X}\max_{y\in Y} d(h(x,y),h(x,x))$$

Pela condição de Lipschitz de h temos:

$$d(h(x,y),h(x,x)) \leq L\max(|x-x|,|y-x|) = L|y-x| \leq Lw(X).$$

Portanto,

$$w(E(X)) \leq Kw(X) \quad \text{onde } K = 2L, \text{ q.e.d.}$$

<u>Corolário 4.1</u>

Se h é real, isto é, assume valores reais, então F satisfaz uma condição de Lipchitz.

Demonstração:

$F(X) = \bar{f}(X) + E(X)$

$w(F(X)) = w(\bar{f}(X)) + w(E(X)) \leq K_1 w(X) + K_2 w(X) \leq Kw(X).$

É consequência da proposição 4 e proposição 3.

Capítulo 2

§ 7 - Funções Racionais

Expressões Racionais

Para definir "expressão racional" vamos nos valer da notação de Backus. Nesta notação o símbolo ::= indica que o que está à sua esquerda é definido pelo que aparece à sua direita. Uma barra vertical | significa "ou exclusivo". Um termo genérico vem delimitado pelos símbolos $<$ e $>$.

Assim definiremos:

$\langle$delimitador$\rangle$ ::= (|)

$\langle$operador$\rangle$ ::= + | - | x | /

Consideraremos dado um conjunto enumerável A cujos elementos chamaremos "variáveis". Uma "constante" será um elemento qualquer de $\mathcal{F}$.

Uma "expressão racional" será uma sequência de variáveis, constantes, operadores e delimitadores e será definida recursivamente por:

$\langle$exp.rac.$\rangle$ ::= $\langle$variável$\rangle$ | $\langle$constante$\rangle$ | (-$\langle$exp.rac.$\rangle$) |

($\langle$exp.rac.$\rangle$ $\langle$operador$\rangle$$\langle$exp.rac.$\rangle$)

Exemplos de expressões racionais

Seja $A = \{X, Y, Z\}$ e denotemos constantes por $C_1, C_2, \ldots$

1) X
2) C_1
3) $-X$
4) $(-((((X+Y)xC_1)/(Y-Z)))$

Funções Racionais de Intervalos

Dada uma expressão racional podemos associar a cada variável um intervalo. A expressão racional define então uma sequência de operações sôbre intervalos, as quais se possíveis de serem executadas definem um nôvo intervalo.

Sejam $X_1, X_2, \ldots, X_n$ as "variáveis" que entram na definição da expressão racional E. A expressão racional E definirá uma função

$$R_E : \mathcal{J}_{I_1} \times \mathcal{J}_{I_2} \times \ldots \times \mathcal{J}_{I_n} \longrightarrow \mathcal{J}$$

desde que as operações aritméticas sejam possíveis.

As funções de intervalos que podem ser obtidas através de expressões racionais da maneira descrita acima serão chamadas "funções racionais de intervalos"

Definição

Sejam E_1 e E_2 duas expressões racionais com n variáveis. Se dados $I_1, I_2, \ldots, I_n \in \mathcal{J}$ tais que

$$R_{E_1} : \mathcal{J}_{I_1} \ \mathcal{J}_{I_2} \times \ldots \times \mathcal{J}_{I_n} \longrightarrow \mathcal{J}$$

e

$$R_{E_2} : \mathcal{J}_{I_1} \times \mathcal{J}_{I_2} \times \ldots \times \mathcal{J}_{I_n} \longrightarrow \mathcal{J}$$

são definidas, isto implicar em $R_{E_1} = R_{E_2}$, dizemos que E_1 e E_2 são "expressões racionais equivalentes".

Observação 1

Alguns parêntesis numa expressão racional são tornados supérfluos pelo estabelecimento de prioridades nas operações; outros, devido à associatividade e comutatividade.

A notação das expressões racionais é simplificada pela eliminação dos parêntesis supérfluos e pela notação de potência X^n em lugar de $(((X \times X) \times X \ldots X)$. Por exemplo, escreveremos $X + Y^3$ em lugar de $X + ((Y \times Y) \times Y)$.

Omitiremos também o símbolo x quando não houver perigo de confusão. Escreveremos, então, a última expressão da seguinte maneira $X + ((Y \times Y)Y)$.

Observação 2

Se numa expressão racional as variáveis forem associadas a números reais (intervalos degenerados) e se as constantes que nela aparecem também forem reais, teremos funções racionais reais de variáveis reais.

Observação 3

Seja E_1 e E_2 duas expressões racionais não equivalentes. As restrições aos reais de R_{E_1} e R_{E_2} podem coincidir e simultâneamente ocorrer $R_{E_1} \neq R_{E_2}$.

Exemplo

Sejam E_1 e E_2 as expressões racionais $X - X^2$ e $X(1-X)$. Façamos $X = [0,1]$. Teremos $X-X^2 \neq X(1-X)$. De fato,

$$X-X^2 = [0,1]-[0,1].[0,1] = [0,1] - [0,1] = [-1,1]$$

$$X(1-X) = [0,1](1-[0,1]) = [0,1]([1,1]-[0,1]) = [0,1]$$

Entretanto, para qualquer número real x, $x-x^2 = x(1-x)$.

Observação 4

Toda função racional de intervalos é monotônica e contínua, como composição de funções monotônicas e contínuas.

Observação 5

Uma extensão a intervalos de uma função racional não é necessàriamente uma função racional de intervalos.

Exemplo

Seja $f: R \longrightarrow R$ definida por $f(x) = x^2$ e a função de intervalos $F : \mathcal{F}_X \longrightarrow \mathcal{F}$ definida por $F([x,y]) = x[x,y]$.

F é òbviamente extensão de f mas não é extensão racional, pois não é monotônica. De fato,

$$I = [2,3] \subset J = [1,4] \not\Longrightarrow F(I) \subset F(J)$$

pois temos $F(I) = 2[2,3] = [4,6]$ e $F(J) = 1[1,4] = [1,4]$.

Conjeturamos neste ponto que mesmo supondo a monotonia, não teremos ainda uma condição suficiente para garantir que a extensão seja uma <u>função racional</u> de intervalos. Propomos como candidato a contra exemplo, o seguinte:

"Seja f definida por $f(x) = x^2$ e F definida por $F(X)=\overline{f}(X)$"

Precisaríamos mostrar que $\overline{f}$, a qual é extensão monotônica de f, não é função racional de intervalos, isto é, não existe nenhuma expressão racional E tal que $\overline{f} = R_E$.

<u>Proposição 1</u>

Seja E uma expressão racional, R_E a função racional de intervalos a ela associada e r a restrição de R_E aos reais. Se em E cada variável ocorre uma única vez, então teremos $R_E = \overline{r}$.

Demonstração (recursiva)

(a) Se E é uma variável ou constante a proposição é verdadeira.

(b) Se é verdadeira para E, é verdadeira para -E.

(c) Suponhamos verdadeira para E_1 e E_2. Mostremos que é verdadeira para $E_1 * E_2$, onde $*$ é um dos símbolos +, -, x ou /.

Temos $R_{E_1 * E_2} = R_{E_1} * R_{E_2}$

Mas $R_{E_1} = \overline{r}_1$ e $R_{E_2} = \overline{r}_2$, por hipótese, onde r_1 e r_2 são as restrições aos reais de R_{E_1} e R_{E_2}, respectivamente, donde

$R_{E_1} * R_{E_2} = \overline{r}_1 * \overline{r}_2$.

Temos que provar que $R_{E_1 * E_2} = \overline{r}$, sendo r a restrição aos reais de $R_{E_1 * E_2}$.

Como $R_{E_1 * E_2} = R_{E_1} * R_{E_2} = \overline{r}_1 * \overline{r}_2$, basta mostrar que $\overline{r}_1 * \overline{r}_2 = \overline{r}$.

Suporemos, para simplificar a notação, que r_1 é função de uma variável x e r_2 de outra variável y. Então, devemos mostrar que

$$\overline{r}_1(X) * \overline{r}_2(Y) = \overline{(r_1 * r_2)}(X,Y),$$ isto é, que

$$\bigcup_{x\in X} r_1(x) * \bigcup_{y\in Y} r_2(y) = \bigcup_{\substack{x\in X\\ y\in Y}} (r_1(x) * r_2(y)) \quad \text{onde } X,Y \in \mathcal{J}$$

(a) Seja $z \in \bigcup_{x X} r_1(x) * \bigcup_{y Y} r_2(y)$, então $z = z' * z''$ com $z' \in r_1(x)$ para algum $x \in X$ e $z'' \in r_2(y)$ para algum $y \in Y$; logo $z \in r_1(x) * r_2(y)$ para algum x de X e algum y de Y, donde $z \in \bigcup_{\substack{x\in X\\ y\in Y}} (r_1(x) * r_2(y))$ e portanto,

$$\bigcup_{x\in X} r_1(x) * \bigcup_{y\in Y} r_2(y) \subset \bigcup_{\substack{x\in X\\ y\in Y}} (r_1(x) * r_2(y))$$

(b) Seja $z \in \bigcup_{\substack{x\in X\\ y\in Y}} (r_1(x) * r_2(y))$, então $z \in r_1(x) * r_2(y)$ para algum $x \in X$ e algum $y \in Y$.

Como $r_1(x) \subset \bigcup_{x\in X} r_1(x)$ e $r_2(y) \subset \bigcup_{y\in Y} r_2(y)$ resulta pela monotonia de $*$ que $r_1(x) * r_2(y) \subset \bigcup_{x\in X} r_1(x) * \bigcup_{y\in Y} r_2(y)$.

Portanto, $z \in \bigcup_{x\in X} r_1(x) * \bigcup_{y\in Y} r_2(y)$, donde

$$\bigcup_{x\in X} r_1(x) * \bigcup_{y\in Y} r_2(y) \subset \bigcup_{\substack{x\in X\\ y\in Y}} (r_1(x) * r_2(y)) \qquad \text{q.e.d.}$$

<u>Exemplo</u>

Seja E a expressão $1 - X^2$. Então R_E é definido por $R_E(X) = 1 - X^2$ para $X \in \mathcal{J}$. Sua restrição aos reais r é definida por $r(x) = 1 - x^2$. Mostremos que para $X = [-1,1]$ temos $R_E(X) \neq \overline{r}(X)$

$$R_E([-1,1]) = 1 - [-1,1].[-1,1] = [1,1]-[-1,1] = [0,2]$$

$$\overline{r}([-1,1]) = \bigcup_{x\in[-1,1]} \{z \mid z=1-x^2\} = [0,1]$$

§ 8 - Importância das Funções Racionais

As seguintes considerações têm por objetivo evidenciar a importância prática das funções racionais de intervalo:

1. Como elas são avaliadas através de uma sequência finita de operações aritméticas, o seu cálculo pode ser feito exatamente (a menos do êrro de arredondamento) num computador.

2. No Cap.4 apresentaremos técnicas adaptadas à aproximação de soluções de sistemas de equações diferenciais ordinárias de 1a.ordem com segundo membro racional. Como veremos no § 15 outros sistemas com segundo membro não racional podem ser reduzidos ao caso racional.

3. Se um têrmo complementar fôr da forma

$$g(x) = r(x)f(x),$$

$r(x)$ racional, $g(x)$ monotônica e $x \in I$, podemos calcular um limite superior de sua variação quando x percorre I, tendo em vista que

$$\overline{g}(I) \subset \overline{r}(I)\overline{f}(I) \subset R(I)\overline{f}(I)$$

onde R é uma extensão racional de r.

Por exemplo, consideremos a série de Taylor com o termo complementar expresso na forma de Lagrange, da função exponencial com argumento real negativo

$$e^x = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^{k-1}}{(k-1)!} + \frac{e^t x^k}{k!} \quad \text{para algum } t \in [x,0]$$

$x < 0$.

Temos $e^t \in [e^x, e^0] \subset [0,1]$ para $t \in [x,0]$, $x < 0$.

Portanto, para todo k inteiro positivo e todo $x < 0$, a função exponencial está contida em

$$P_k(x) = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^{k-1}}{(k-1)!} + [0,1]\frac{x^k}{k!}$$

Temos $e^x \in P_k(x)$ para $k=1,2,\dots$ e $x \leq 0$

Avaliando-se $P_k(x)$ com a aritmética de intervalos arredondada (§ 10) obteremos um intervalo contendo o valor exato de e^x, para $x < 0$. A distância $d(e^x, P_k(x))$ pode ser tornada arbitràriamente pequena, considerando-se k suficientemente grande e uma aritmética com precisão suficientemente elevada.

§ 9 - Extensões de Funções Racionais

Neste parágrafo abordaremos o seguinte problema:

"Dada uma função racional f real de variáveis reais, como obter uma extensão racional a intervalos F, tal que $d(F(X),\overline{f}(X))$ seja pequena ?"

Nas aplicações gostaríamos de trabalhar com a extensão unida $\overline{f}$. Como sua avaliação não é em geral simples, preferimos substituí-la por uma extensão racional F. Como $F(X) \supset \overline{f}(X)$ teremos resultados a favor da segurança. Esta a motivação para o problema acima.

Se a função racional f pode ser obtida a partir de uma expressão racional E sem variáveis repetidas, a proposição 1 do § 7 nos dá a solução ótima: $F = R_E = \overline{f}$

No que se segue vamos supor não ser êste o caso. No caso geral não existem ainda definições aceitas para caracterizar uma "solução ótima" do problema. Caso uma tal definição seja proposta,esta levantará imediatamente a questão de existência e unicidade de solução ótima".

Limitar-nos-emos a expor algumas técnicas que em geral dão bons resultados:

1. Técnica da interseção

Sejam F_1 e F_2 duas extensões racionais de f. Como $\overline{f}(X) \subset F_1(X)$ e $\overline{f}(X) \subset F_2(X)$ temos $\overline{f}(X) \subset F_1(X) \cap F_2(X)$ e

$$F_1(X) \cap F_2(X) \subset F_1(X)$$
$$F_1(X) \cap F_2(X) \subset F_2(X)$$

Nesta técnica substituimos $\bar{f}$ por G definida por

$$G(X) = F_1(X) \cap F_2(X)$$

que pode não ser uma extensão racional de $\bar{f}$. Se $F_1(X) \not\subset F_2(X)$ e $F_2(X) \not\subset F_1(X)$ teremos uma melhoria na estimativa para $\bar{f}(X)$.

2. Técnicas de redução de repetições de variáveis

Dentro do espírito desta técnica adotamos uma expressão racional tal que f seja obtida a partir dela e que tenha o menor número de repetições de variáveis. Como o caso sem repetição é o caso ideal, esperamos que ao reduzir o número de repetições a situação seja mais favorável.

Exemplo

Sejam E_1 e E_2 as expressões racionais $\frac{X}{X-2}$ e $1 + \frac{2}{X-2}$, respectivamente. Façamos $X = [3,4]$. Teremos

$X/(X-2) = [3,4]/([3,4]-[2,2]) = [3,4]/[1,2] = [\frac{3}{4}, 4]$ e

$1+2/(X-2) = 1+2/([3,4]-[2,2]) = [1,1]+[2,2]/[1,2] = [2,3]$

Portanto, $1 + 2/(X-2) \subset X/(X-2)$, $X = [3,4]$

Casos particulares

(a) Se $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_o$

podemos adotar a extensão racional em forma de ninho:

$$F(X) = (((a_n X + a_{n-1})X + \ldots + a_1)X + a_o$$

a qual reduz o número de repetições a n. Neste caso a propriedade semi-distributiva garante que

$$F(X) \quad a_n X^n + a_{n-1} X^{n-1} + \ldots + a_o$$

(b) Se f é racional podemos exprimí-la numa forma de fração contínua. Por exemplo, seja $f(x) = (x^2-x+1)/(x^2 + 1)$. Podemos escrever $f(x) = 1 - 1/(x + 1/x)$ e adotar $F(X) = 1 - 1/(X+1/X)$

3. Técnica do Desenvolvimento de Taylor

Seja f uma função real de variável real, X o intervalo para o qual se deseja calcular a extensão unida $\overline{f}(X)$ e c um ponto de X. Desenvolvendo em forma de Taylor,

$$f(x) = f(c) + f'(c) + \frac{(x-c)^2}{2!}f''(c) + \ldots + \frac{(x-c)^{n-1}}{(n-1)!}f^{(n-1)}(c) + R_n(x,c)$$

Conforme o tratamento dado ao termo complementar teremos os procedimentos:

(a) $R_n(x,c) = \frac{(x-c)^n}{n!} f^{(n)}(c + \theta(x-c)) \qquad 0 \leq \theta \leq 1$

Seja Fn uma extensão racional de $f^{(n)}$. Então,

$$f(x) \in f(c) + (x-c)f'(c) + \ldots + \frac{(x-c)^{n-1}}{(x-c)!}f^{(n-1)}(c) + \frac{(x-c)^n}{n!} Fn(c + [0,1](x-c))$$

donde,

$$\overline{f}(X) \subset f(c) + (X-c)f'(c) + \ldots + \frac{(X-c)^{n-1}}{(n-1)!}f^{(n-1)}(c) + \frac{(X-c)^n}{n!} Fn(c + [0,1](X-c))$$

Quando $n=1$, temos a <u>forma do valor médio</u>

$$f(c) + (X-c)Fn(c + [0,1](X-c))$$

(b) $R_n(x,c)$ é uma função racional dada por

$$R_n(x,c) = \frac{f(x)-f(c)-(x-c)f'(c)-\ldots-(x-c)^{n-1}f^{(n-1)}/(n-1)!}{(x-c)^n/n!}$$

Escrevendo $f(x) = f(x-c+c) = g(x-c)$ e chamando $x-c = y$, teremos $R_n(x,c) = G(y)$ e

$$f(x) = f(c) + yf'(c) + \ldots + \frac{y^{n-1}}{(n-1)!}f^{(n-1)}(c) + \frac{y^{(n)}}{n!}G(y)$$

Portanto,

$$\bar{f}(X) \subset f(c) + Yf'(c) + \dots + \frac{Y^{n-1}f^{(n-1)}(c)}{(n-1)!} + \frac{Y^n G(Y)}{n!}$$

a qual é denominada forma centrada quando o ponto c é tomado como o ponto médio do intervalo X.

Se f é um polinômio de grau n, o resto $R_n(x,c)$ é constante.

Exemplo 1

Escrever a forma centrada de f definida por $f(x)=x-x^2$

Temos $f(x) = f(c) + (x-c)f'(c) + \frac{(x-c)^2}{2!}f''(c)$, donde

$$f(x) = c-c^2 + (x-c)(1-2c) - \frac{(x-c)^2}{2!}(2), \text{ isto é,}$$

$$f(x) = c-c^2 + (x-c)\{(1-2c) - (x-c)\}$$

Finalmente, considerando c = ponto médio do intervalo X para o qual se deseja calcular a extensão unida $\bar{f}(X)$, obtemos

$$\bar{f}(X) \subset c-c^2 + (X-c)\{(1-2c) - (X-c)\}$$

Exemplo 2

Escrever a forma centrada de $f(x_1,x_2) = (x_1+x_2)/(x_1-x_2)$

Sejam X_1 e X_2 os intervalos para os quais se deseja calcular $\bar{f}(X_1,X_2)$, c_1 e c_2 os pontos médios dos intervalos X_1 e X_2, respectivamente. A extensão de f sob forma centrada é obtida a partir de

$$f(x_1,x_2) = f(c_1,c_2) + g(x_1-c_1,x_2-c_2) = f(c_1,c_2) + g(y_1,y_2),$$

donde decorre $g(y_1,y_2) = f(x_1,x_2)-f(c_1,c_2) = f(c_1+y_1,c_2+y_2)-f(c_1,c_2)$

isto é,

$$g(y_1,y_2) = \frac{2(c_1y_2 - c_2y_1)}{(c_1-c_2)^2 + (c_1-c_2)(y_1-y_2)}$$

cujo numerador foi obtido efetuando-se operações algébricas elementares entre as variáveis e constantes, reduzindo-se o número de ocorrências das variáveis, de acôrdo com os cancelamentos abaixo:

$$(c_1-c_2)(c_1+y_1+c_2+y_2) - (c_1+c_2)(c_1+y_1-c_2-y_2) =$$

$$\cancel{c_1^2} - \cancel{c_2^2} + c_1y_1 + c_1y_2 - c_2y_1 - \cancel{c_2y_2} - \cancel{c_1^2} + \cancel{c_2^2} - \cancel{c_1y_1} + c_1y_2 -$$

$$c_2y_1 + \cancel{c_2y_2} = c_1y_2 - c_2y_1 + c_1y_2 - c_2y_1 = 2(c_1y_2 - c_2y_1)$$

Finalmente, obtemos a seguinte forma centrada da extensão

$$\frac{c_1+c_2}{c_1-c_2} + \frac{2\left\{c_1(X_2-c_2) - c_2(X_1-c_1)\right\}}{(c_1-c_2)^2 + (c_1-c_2)\left\{(X_1-c_1) - (X_2-c_2)\right\}}$$

Observação

Eventualmente mais de uma técnica podem ser utilizadas simultâneamente. Por exemplo, é recomendável aplicar a técnica da forma de ninho juntamente com a técnica do desenvolvimento de Taylor.

Ilustração numérica

Seja f definida por $f(x) = x-x^2$, $X = [a,b]$ o intervalo para o qual se deseja calcular $\overline{f}(X)$, $c = (a+b)/2 = 1/2$ e r definido por $(b-a)/2$. Temos $X-c = [-r,r]$ e $X = [1/2-r, 1/2+r]$.

1. Extensão racional sob forma de ninho

$$X(1-X) = [1/2-r, 1/2+r]([1,1]-[1/2-r,1/2+r]) =$$

$$= [1/2-r, 1/2+r].[1/2-r, 1/2+r] =$$

$$= [\min\{(1/2-r)^2, (1/2-r)(1/2+r)\}, (1/2+r)^2]$$

pois $(1/2-r)(1/2-r) = (1/2-r)^2$

$(1/2-r)(1/2+r) = 1/4 - r^2$

$(1/2+r)(1/2+r) = (1/2+r)^2$

donde vemos que se $r \leq 1/2$, o menor valor é $(1/2-r)^2$ e se $r > 1/2$, o menor valor é $(1/2-r)(1/2+r)$.

2. Extensão racional sob forma centrada

Utilizando o resultado obtido no exemplo 1 e substituindo no mesmo $c = 1/2$ e $X = [1/2-r, 1/2+r]$, obtemos

$$1/4 + [-r,r].(-[-r,r]) = 1/4 + [-r^2,r^2] =$$

$$= [1/4-r^2, 1/4+r^2]$$

3. Extensão racional sob forma do valor médio

Seja F_1 a extensão racional de f' definida por 1-2X.

Substituindo em $f(c) + (X-c)F_1(c+[0,1](X-c))$, obtemos

$$f(1/2) + [-r,r].F_1(1/2 + [0,1].[-r,r])$$

$$1/4 + [-r,r]([1,1]-[2,2]([1/2,1/2]+[0,1][-r,r])$$

$$1/4 + [-r,r]([1,1]-[2,2].[-1/2-r,1/2+r])=$$

$$1/4 + [-r,r]([1,1]-[1-2r,1+2r]) =$$

$$1/4 + [-r,r].[-2r,2r] =$$

$$[1/4-2r^2, 1/4+2r^2]$$

4. Comparação dos resultados com $\bar{f}(X)$

Temos $\bar{f}(X) = \{ f(x)=x-x^2 \mid x \in X \} = [1/4-r^2, 1/4]$

Comparando-se (1),(2) e (3) com $\bar{f}(X)$ verificamos, no caso em que $r \leq 1/2$, que o melhor resultado obtido foi proporcionado pelo uso da extensão sob forma centrada, seguido do intervalo - calculado pela extensão sob forma do valor médio.

Com base nos resultados deste e de outros exemplos, Moore lançou a seguinte conjetura (pg.45 de Interval Analysis, R.E.Moore)

" Seja f uma função racional real de variáveis reais $x_1, x_2 \ldots x_n$; $X_1, X_2, \ldots, X_n$ os intervalos onde se deseja calcular a extensão unida $\overline{f}(X_1,X_2,\ldots,X_n)$, $c_1, c_2, \ldots, c_n$ pontos médios dos intervalos $X_1, X_2, \ldots, X_n$, respectivamente e F a extensão racional sob forma centrada da função f, definida por

$$F(X_1,X_2,\ldots,X_n) = f(c_1,c_2,\ldots,c_n) + g(x_1-c_1,\ldots,x_n-c_n)$$

então, $w(F(X_1,X_2,\ldots,X_n) \leq w(\overline{f}(X_1,X_2,\ldots,X_n) + O(w(X)^2)$."

Hansen prova em "Topics in Interval Analysis ", pg.103, que essa conjectura é verdadeira.

Capítulo 3

§ 10-Aritmética de Intervalos com Arredondamento

A maior parte dos números reais que ocorrem nas aplicações não podem ser representados de forma exata quando se usa a representação decimal ou a representação numa outra base, por exemplo, a binária. E o caso da representação do número racional 1/3 ou do número irracional $\sqrt{2}$. O número de dígitos usados em sua representação é limitado e não supera, na prática, um número prefixado. O valor exato de um número real pode ser dado, teòricamente, por meio de uma representação decimal infinita. Se nessa representação considerarmos apenas os n primeiros dígitos, obtemos um valor aproximado por falta, cujo êrro é inferior a uma unidade da última "casa". Pela aplicação da técnica usual de arredondamento, êste êrro pode ser reduzido à metade, obtendo-se um resultado aproximado por falta ou excesso. O primeiro tipo de aproximação dá origem à aritmética truncada, enquanto que o segundo à aritmética arredondada.

As operações aritméticas executadas com os números escritos sob forma aproximada dão por resultados números que representam valores aproximados de algum valor exato. Mesmo que os números iniciais sejam exatos, êrros podem ser introduzidos durante a execução das operações aritméticas. O êrro oriundo da propagação desses êrros através de inúmeras operações aritméticas é denominado êrro de arredondamento. Na resolução de um problema que envolve centenas ou mesmo milhares de operações aritméticas, pergunta-se como obter uma delimitação superior do êrro de arredondamento a fim de se estimar a precisão do resultado aproximado. A resposta a esta questão muitas vêzes não é simples, sendo frequentemente sofisticada e extremamente trabalhosa.

Quando se emprega a aritmética de intervalos os dados iniciais são intervalos que contêm os valores exatos; se os dados iniciais são exatos, então os intervalos são degenerados. Mesmo neste cado, no decorrer da execução das operações aritméticas com intervalos, novos intervalos são obtidos e é preciso assegurarmos que êstes ainda contêm o valor exato do resultado final. Esta certeza é conseguida, procedendo-se da seguinte maneira, admitindo-se o emprêgo da aritmética real truncada:

(a) verifica-se se o extremo inferior do intervalo resultante é um número exato ou não. Se fôr exato permanece inalterado. Se não fôr exato, verifica-se o sinal; se êste fôr negativo subtrai-se uma unidade do dígito de ordem mais baixa, caso contrário permanece como está.

(b) verifica-se se o extremo superior do intervalo resultante é um número exato ou não. Se fôr exato permanece inalterado. Se não fôr exato, verifica-se o sinal; se êste fôr positivo acrescenta-se uma unidade no dígito de ordem mais baixa, caso contrário permanece como está.

Procedendo-se desta maneira obtém-se um intervalo que conterá sempre o resultado exato da correspondente operação aritmética executada com valores exatos, isto é, com precisão infinita, caso isto fosse possível. No final da execução de todas operações aritméticas necessárias para obtenção da solução de um problema, chegá-se a um intervalo que contém a solução exata procurada. Se êste intervalo fôr de comprimento suficientemente pequeno, a correspondente delimitação do êrro de arredondamento também será pequena, estimando-se com esta delimitação a precisão do resultado aproximado. Por exemplo, seja $[a,b]$ o intervalo encontrado. O número aproximado ao valor exato da solução, juntamente com o êrro de que se encontra afetado, é dado por $\frac{a+b}{2} \pm \frac{b-a}{2}$. Denominamos esta representação forma centrada do intervalo.

A aritmética de intervalos descrita acima, em que cada intervalo é aumentado em favor da segurança após a execução de uma operação aritmética, denomina-se aritmética de intervalos com arredondamento.

A pesquisa de algoritmos que levem à obtenção de intervalos com comprimentos suficientemente pequenos, contendo valores exatos da solução de problemas, é o objeto da análise de intervalos.

A aritmética de intervalos pode ser introduzida ou simulada - num computador. Recordemos que, nos computadores, os números em geral são representados internamente sob duas formas:

(i) Representação inteira ou com vírgula fixa:

Um número inteiro é representado por $\pm d_1d_2...d_k$ numa certa base, em geral 2 ou 10.

(ii) Representação real com vírgula flutuante:

Um número real distinto de zero é representado por $\pm,d_1d_2...d_t x B^n$ com $d_1 \neq 0$, isto é, como o produto de uma parte fracionária ou mantissa, escrita com t dígitos, pela base B elevada a um inteiro n, chamado característica. O número n obedece à restrição $-m \leq n \leq M$. O número zero é representado por $,00...0B^{-m}$. Em geral a base é 2 ou 10. Esta representação é também chamada representação normalizada em vírgula flutuante.

É interessante observar que o conjunto $F(B,t,m,M)$ dos números representados nesta forma é um subconjunto finito do conjunto dos números racionais e seus elementos se distribuem numa reta com um espaçamento não uniforme.

Conforme o computador, usá-se a aritmética truncada ou arredondada. No primeiro caso, conforme vimos, o êrro máximo de aproximação é de uma unidade do dígito de ordem mais baixa, enquanto que no segundo, de apenas 0,5 unidade.

Subrotinas para a aritmética de intervalos - As subrotinas apresentadas no apêndice B e sucintamente descritas no apêndice A foram redigidas em FORTRAN, simulando-se a aritmética real truncada, uma vez que o processamento dos programas foi feito no B-3500, no qual a aritmética é semi-truncada.

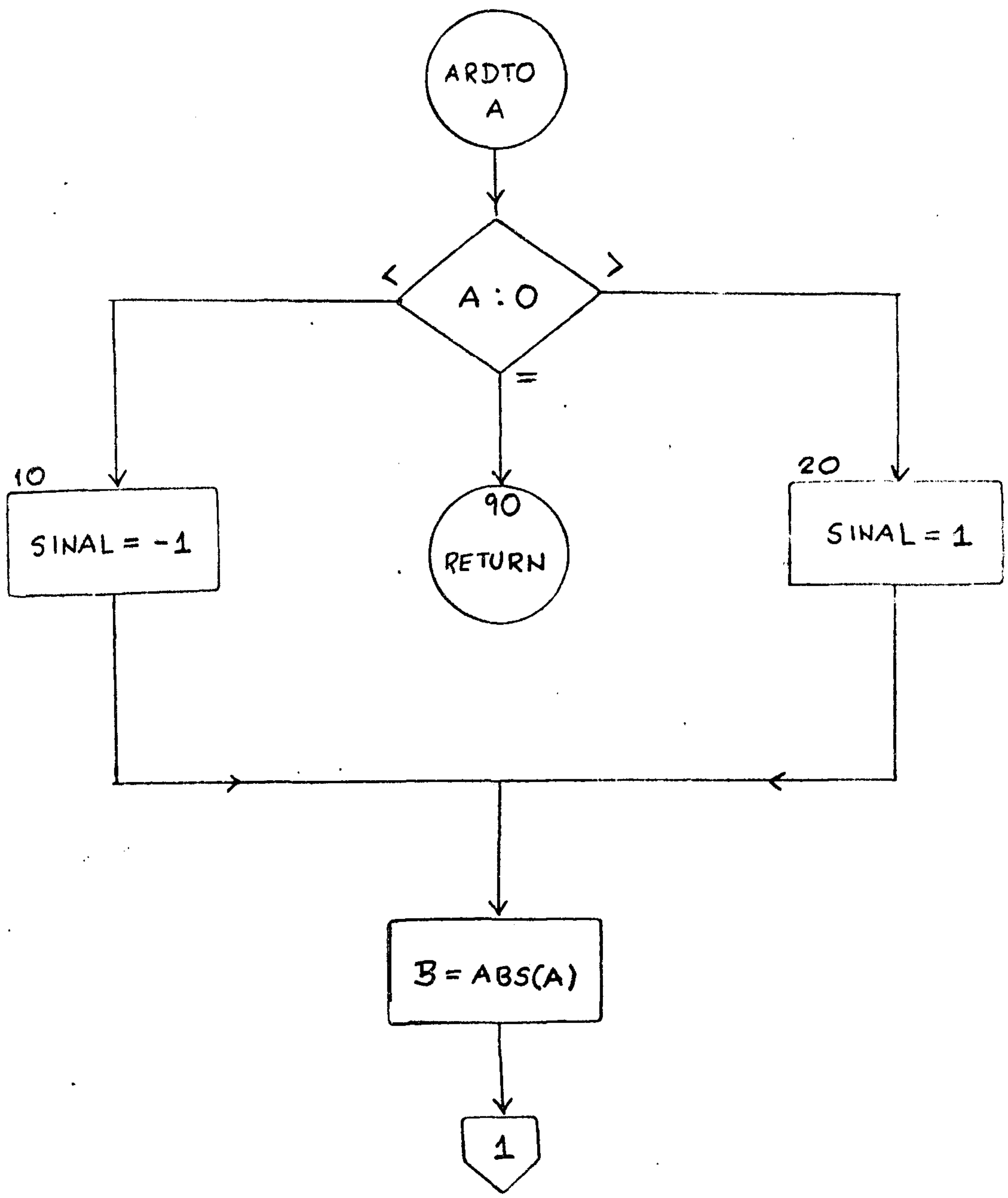
O arredondamento do intervalo foi programado da seguinte forma: a mesma operação aritmética é executada duas vêzes, uma em precisão simples e outra em precisão dupla; em seguida os resultados são comparados entre sí. Se êstes forem iguais, não há êrro de aproximação e o resultado em precisão simples permanece inalterado; no caso contrário o resultado em precisão simples será arredondado de acôrdo com a regra de arredondamento para intervalos.

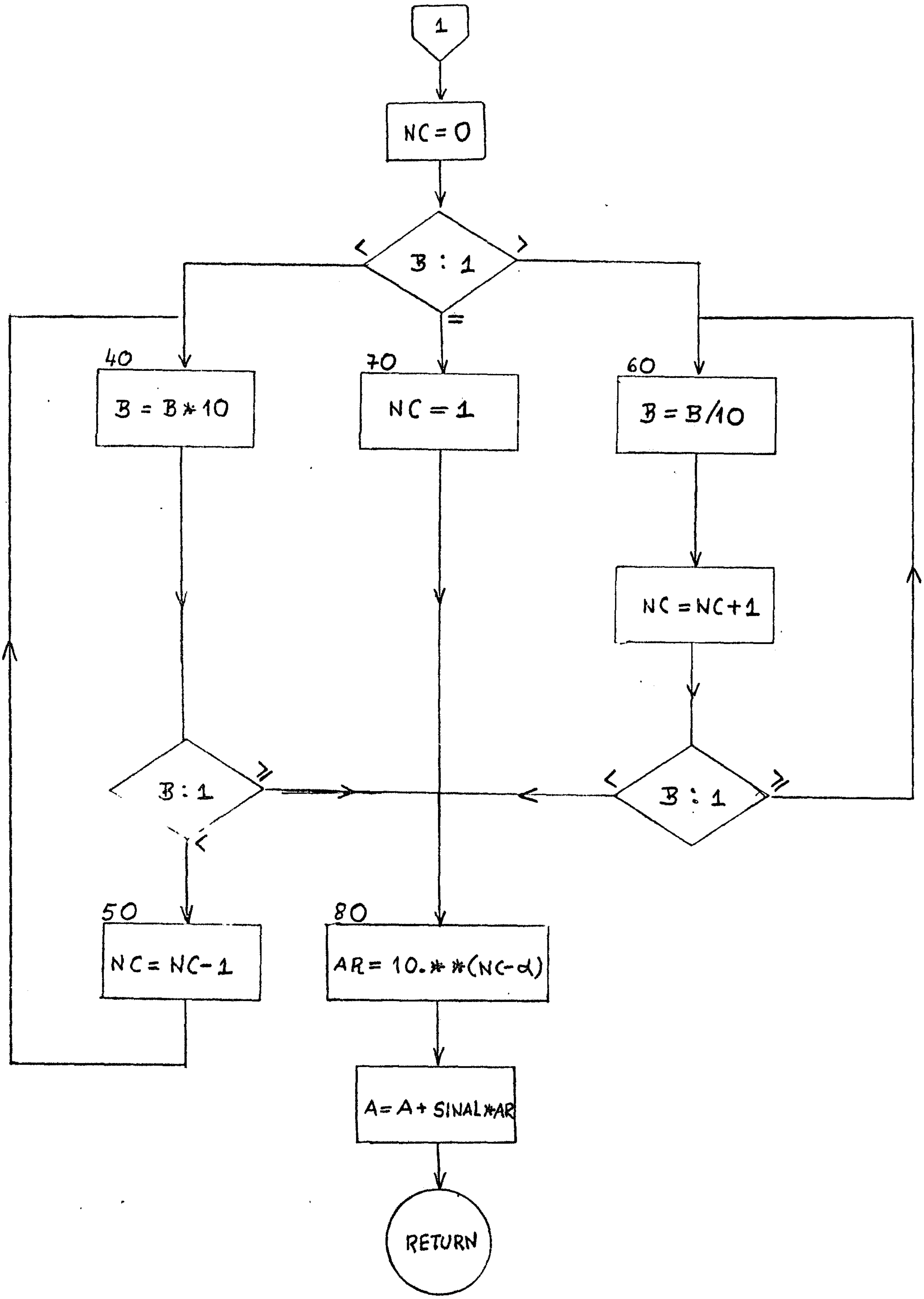
Na elaboração das subrotinas para a aritmética de intervalos com arredondamento e em precisão dupla, devido à impossibilidade de simulação em FORTRAN, com os recursos disponíveis na biblioteca de subrotinas, da aritmética em precisão dupla truncada, arredondamos as operações de soma e subtração em que os operandos somam-se em valor absoluto. Além disso deixamos de verificar se o resultado de uma determinada operação aritmética é exato ou aproximado, arredondando o extremo inferior no caso em que é negativo e arredondando o extremo superior no caso em que é positivo. Estas subrotinas em precisão dupla são necessárias para o método de Hansen, exposto em parágrafos subsequentes.

<u>Subrotinas para aritmética de ternos</u> - O terno $[\underline{x},\tilde{x},\overline{x}]$ representa o intervalo $[\underline{x},\overline{x}]$ e um número real $\tilde{x}$ entre $\underline{x}$ e $\overline{x}$. Na aritmética de ternos usa-se a aritmética de intervalos para os intervalos $[\underline{x},\overline{x}]$ e a aritmética real para os números $\tilde{x}$. As subrotinas apresentadas no apêndice B e sucintamente descritas no apêndice A receberam os nomes usados nas subrotinas para a aritmética de intervalos com arredondamento. Nas aplicações que faremos a seguir as subrotinas feitas para intervalos, em precisão simples ou dupla, são usadas apenas para ilustrar o método de Hansen, motivo pelo qual demos os mesmos nomes às subrotinas para interva -

los e para ternos referentes ao mesmo tipo de operação.

Apresentamos a seguir o diagrama de blocos (FORTRAN) da subrotina ARDTO utilizada pela subrotina ARDT2 para arredondamento de um intervalo. A subrotina ARDTO acrescenta uma unidade no dígito - de ordem mais baixa do número real dado como argumento. Embora o cálculo seja o de uma função o seu argumento fornecido pela ARDT2 é na forma de variável indexada. A subrotina ARDTO depende do size - real com o qual o programa principal fôr executado. No diagrama de blocos indicamos a magnitude desse size por α .




1
NC = 0
B : 1
<
>
=
40
B = B*10
70
NC = 1
60
B = B/10
NC = NC+1
B : 1
≥
<
B : 1
<
≥
50
NC = NC-1
80
AR = 10.**(NC-α)
A = A + SINAL*AR
RETURN

§ 11 - Resolução de um Sistema Linear por um método direto

O método de aliminação de Gauss para resolução de um sistema linear é um método direto porque envolve apenas um número finito de operações aritméticas para obtenção da solução exata do sistema. Se os coeficientes da matriz completa associada ao sistema - são exatos, a única fonte de êrro é devida aos êrros de arredon - damento cometidos na execução das operações aritméticas. Se, en - tretanto, os coeficientes não são exatos, teremos que considerar também a propagação dos êrros iniciais. Na prática os êrros ini - ciais são assimilados aos êrros de arredondamento.

O uso do método de Gauss com a aritmética de ternos (que envolve a aritmética de intervalos com arredondamento) permite a obtenção de intervalos que contêm as componentes do valor exato - da solução. Convém observar que todo método direto estabelecido para a aritmética real continua formalmente válido para a aritmética de ternos, a menos de que o número zero venha a pertencer a um intervalo, que figure, numa determinada etapa do processo, como divisor. Entretanto, o resultado nem sempre é satisfatório,- porque o comprimento do intervalo que contem a solução pode se tornar muito grande, impedindo uma delimitação razoável do êrro de arredondamento. Eis uma justificativa para uso do terno no lugar do intervalo, pois se o resultado em têrmos de intervalo ca - recer de significado, resta-nos a solução que normalmente obteríamos usando simplesmente a aritmética real, relativamente ao elemento $\tilde{x}$ do terno.

O aumento excessivo do comprimento dos intervalos, no decurso da execução das operações aritméticas, não depende apenas do arredondamento do intervalo, mas também, à impossibilidade, muitas vêzes, de se calcular o intervalo exato de variação de valores de uma função f(x) para x percorrendo um certo intervalo X. De fato, no caso de funções racionais, se, na expressão racional usada para a extensão F(X), a variável x ocorrer mais de uma vêz, então $F(X) \supset \bar{F}(X)$, que representa o intervalo exato de variação de valores de f(x) para x percorrendo X. Um algoritmo que utilize aritmética de intervalos pressupoẽ uma análise das diversas etapas em que se procura introduzir extensões F(X) que se aproximem o mais possível de $\bar{F}(X)$.

Eis o motivo porque, mesmo utilizando, na versão do método de Gauss para intervalos, uma aritmética real de precisão infinita, não seria possível obter o intervalo exato de variação de valores. Entretanto, se não há êrros iniciais, a versão do método de Gauss para intervalos permite a obtenção de intervalos arbitràriamente pequenos contendo a solução exata, desde que a precisão da aritmética seja suficientemente elevada.

<u>Programa em FORTRAN do método de Gauss</u> - O programa que apresentamos neste texto reproduz o algoritmo do método de Gauss conforme exposto no § 1 do Capítulo 1 de"<u>Métodos Numéricos - I - Algebra Linear</u>" do prof. Ivan de Q.Barros. Após equilibrada a matriz dos coeficientes usa-se a técnica de pivotação na decomposição da matriz em duas matrizes triangulares, inferior e superior. Não é feita a permutação física dos elementos entre 2 linhas, guardando-se a ordem das permutações numa matriz auxiliar chamada matriz de permutação. A decomposição da matriz dos coeficientes é feita através da subrotina DECMAT(N,A,KP) e a resolução dos sistemas triangulares é feita através da subrotina RSTISV(N,A,KP). Vide apêndices A e B.

Programa em FORTRAN para resolução de sistema linear pelo método de eliminação de Gauss e uso da aritmética de ternos arredondada

```
SIZE REAL = 12
      DIMENSION A(3,20,21),KP(20),B(3,20,21)
 1000 READ(5,200,END=2000)N
      M=N+1
      WRITE(6,700)
  700 FORMAT(1X,14HSISTEMA LINEAR,///)
      DO 1 I=1,N
      READ(5,300)(A(2,I,J),J=1,M)
    1 WRITE(6,400)(A(2,I,J),J=1,M)
      DO 2 I=1,N
      DO 2 J=1,M
      A(1,I,J)=A(2,I,J)
    2 A(3,I,J)=A(2,I,J)
   40 CALL DECMAV(N,A,KP)
      CALL RSTISV(N,A,KP)
C
      WRITE(6,800)
  800 FORMAT(//,1X,18HSOLUCAO COM TERNOS,//)
      DO 130 I=1,N
      LI=KP(I)
      CALL FCI(A(1,LI,M),B(1,LI,M))
  130 WRITE(6,600)I,LI,I,A(1,LI,M),A(2,LI,M),A(3,LI,M)
      WRITE(6,900)
  900 FORMAT(//,1X,26HSOLUCAO SOB FORMA CENTRADA,//)
      DO 132 I=1,N
      LI=KP(I)
  132 WRITE(6,600)I,LI,I,B(1,LI,M),B(2,LI,M),B(3,LI,M)
      GO TO 1000
 2000 STOP
  200 FORMAT(I2)
  300 FORMAT(5F7.4)
  400 FORMAT(/,1X,5F12.7)
  600 FORMAT(/,1X,3HKP(,I2,2H)=,I2,2X,2HX(,I2,2H)=,3E24.12)
      END
```

```
SYMBOLIC LISTING FOR PROGAM ≥

LOW ADRS   HIGH ADRS  TEMPS     DATA BASE  CODE BASE
 002106     047748    002118     002148     042992

CALLED SUBPROGRAMS ≥
NAME     ADDRESS   SEG. NO.
DECMAV    085314     001
RSTISV    067730     001
FCI       060738     001
READ.     055248     001
WRITE.    047748     001

COMMON BLOCKS REFERENCED ≥
NAME    ADDRESS
NONE
```

Resultados Numéricos

O resultado numérico dos exemplos que apresentaremos a seguir encontram-se no apêndice C. Está implícita a introdução de modificações nos comandos de entrada do programa apresentado, a fim de permitirem a leitura correta dos ternos, no caso em que os dados estão afetados de êrros iniciais.

Exemplo 1

Sistema linear de ordem 10, supondo-se os coeficientes da matriz completa exatos. A primeira solução foi obtida considerando-se a representação real em vírgula flutuante com 8 algarismos (size - real = 8); na obtenção da segunda solução empregamos 12 algarismos (size real = 12). A segunda coluna da solução corresponde à solução obtida processando-se o algoritmo de Gauss com aritmética real. A terceira coluna da solução sob forma centrada dá a delimitação do êrro de arredondamento. Observamos que o maior êrro cometido no primeiro caso é da ordem de 10^{-2}, enquanto que no segundo é da ordem de 10^{-6}.

Neste exemplo, em que não há propagação de êrros iniciais, ilustramos que podemos obter solução tão precisa quanto desejarmos se usarmos uma aritmética de precisão suficientemente elevada. É interessante observar que o valor exato, de cada componente da solução, é o número inteiro 1.

Exemplo 2

Sistema linear bem condicionado de ordem 4. O programa foi processado com size real = 12. A primeira solução supõe inexistência de êrros iniciais. A segunda solução admite que os êrros iniciais não superam 0,00005 unidade . No primeiro caso o êrro máximo é da ordem de 10^{-10} e no segundo, da ordem de 10^{-3}.

Neste exemplo ilustramos que, havendo êrro inicial, não é possível melhorar indefinidamente a precisão da solução pelo uso de uma aritmética de precisão elevada.

Exemplo 3

Sistema linear mal condicionado de ordem 4. O programa foi processado com size real = 12. A primeira e a segunda solução foram obtidas nas mesmas condições do exemplo 2. Na primeira solução o êrro máximo é da ordem de 10^{-6} e na segunda, de 10^{1}.

Neste exemplo ilustramos que o mal condicionamento do sistema é automàticamente revelado pela magnitude do êrro máximo cometido.

§ 12 - Inversão de Matriz

Seja A uma matriz de intervalos. Consideremos uma matriz numérica Ar cujos elementos $(Ar)_{ij} \in A_{ij}$. Dizemos que Ar pertence a A ou que Ar está contida em A.

Suponhamos que tôda $Ar \in A$ seja inversível. Consideremos a matriz B tal que

$$B_{ij} = \left\{ (Ar^{-1})_{ij} \mid Ar \in A \right\} \tag{1}$$

Da continuidade da operação de inversão concluimos que B_{ij} é um intervalo. A matriz de intervalos B chama-se inversa de A, sendo também indicada por A^{-1}.

Portanto, se $Ar \in A$, então $Ar^{-1} \in A^{-1}$ e podemos escrever

$$\left\{ Ar^{-1} \mid Ar \subset A \right\} \subset A^{-1}$$

Proposição 1

A^{-1} é a "menor" matriz de intervalos que contém

$$\left\{ Ar^{-1} \mid Ar \subset A \right\}$$

(isto é, se C é matriz de intervalos e $C \supset \left\{ Ar^{-1} \mid Ar \subset A \right\}$, então $C \supset A^{-1}$).

Demonstração:

Se $C \supset \left\{ Ar^{-1} \mid Ar \subset A \right\}$, então $Ar^{-1} \in C$, donde $(Ar^{-1})_{ij} \in C_{ij}$ para todo $Ar \in A$. Logo $\left\{ (Ar^{-1})_{ij} \mid Ar \in A \right\} \subset C_{ij}$ e por (1) vemos que $B_{ij} \subset C_{ij}$, donde segue que $A^{-1} = B \subset C$, q.e.d.

Observação

A recíproca da proposição 1 não é verdadeira, isto é, nem sempre

$$\left\{ Ar^{-1} \mid Ar \subset A \right\} \supset A^{-1}$$

Consequentemente, $\left\{ Ar^{-1} \mid Ar \subset A \right\}$ não pode ser identificada a uma matriz de intervalos. Vejamos um exemplo. Seja

$$A = \begin{bmatrix} [1,2] & [0,1] \\ [0,0] & [2,3] \end{bmatrix} \quad \text{e} \quad Ar = \begin{bmatrix} x & y \\ 0 & z \end{bmatrix} \in A$$

$$\text{Então,} \quad Ar^{-1} = \begin{bmatrix} \frac{1}{x} & -\frac{y}{xz} \\ 0 & \frac{1}{z} \end{bmatrix} \quad \text{e} \quad A^{-1} = \begin{bmatrix} [\frac{1}{2},1] & [-\frac{1}{2},0] \\ 0 & [\frac{1}{3},\frac{1}{2}] \end{bmatrix}$$

$$\text{Consideremos} \quad Br = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix} \in A^{-1}.$$

$$Br^{-1} = \frac{1}{2}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} \end{bmatrix} = Ar$$

$$Br = Ar^{-1} \quad \text{e} \quad Ar \notin A$$

Se utilizássemos a versão do método de Gauss para intervalos no cálculo da inversa de uma matriz A de intervalos, obteríamos intervalos que não coincidiriam com os intervalos exatos de variação de valores de cada elemento da matriz inversa, ainda que durante os cálculos não fossem cometidos êrros de arredondamento. Conforme vimos, esta impossibilidade decorre da múltipla ocorrência, por exemplo, da variável a_{11}.

O método de E.Hansen para inversão de matriz de intervalos é uma adaptação de um método de refinamento de matriz inversa numérica e permite dizer de quanto os extremos de cada intervalo calculado diferem dos extremos dos intervalos exatos de variação de valores.

<u>Lema 1</u>

Seja Er uma matriz numérica e $\|\ \|$ uma norma qualquer.

$$\text{Se } \|Er\| < 1 \quad \text{e} \quad Sr^{(m)} = I + Er + Er^2 + \dots + Er^m \tag{2}$$

então: (i) $Sr^{(m)} \longrightarrow (I-Er)^{-1}$ para $m \to \infty$

$$\text{(ii)} \quad \|(I-Er)^{-1} - Sr^{(m)}\| \leq \frac{\|Er\|^{m+1}}{1 - \|Er\|} \tag{3}$$

Demonstração:

(i) $(I-Er)Sr^{(m)} = I - Er^{m+1}$, donde $I - (I-Er)Sr^{(m)} = Er^{m+1}$ e

$$||I-(I-Er)Sr^{(m)}|| = ||Er^{m+1}||$$

Pela propriedade multiplicativa da norma de matriz $||Er^{m+1}|| \leq ||Er||^{m+1}$. Portanto, se $||Er|| < 1$,

$$||I-(I-Er)Sr^{(m)}|| \leq ||Er||^{m+1} \longrightarrow 0 \quad \text{para} \quad m \longrightarrow \infty$$

Como a norma da diferença converge para zero, cada elemento de $Sr^{(m)}$ converge para o correspondente elemento de $(I-Er)^{-1}$, qed.

(ii) $(I-Er)^{-1} - Sr^{(m)} = (I+Er+Er^2+\ldots) - (I+Er+\ldots+Er^m) = Er^{m+1}(I+Er+Er^2+\ldots)$. Daí, $||(I-Er)^{-1}-Sr^{(m)}||=||Er^{m+1}||.||I+Er+Er^2+\ldots||$. Como $||Er^{m+1}|| \leq ||Er||^{m+1}$ e $||I+Er+Er^2+\ldots|| \leq 1 + ||Er|| + ||Er||^2 + \ldots = \dfrac{1}{1 - ||Er||}$,

concluímos que

$$||(I-Er)^{-1} - Sr^{(m)}|| \leq ||Er||^{m+1}/(1-||Er||), \text{ q.e.d.}$$

<u>Definição</u>

Seja Ar uma matriz numérica e B_k uma aproximação da matriz inversa Ar^{-1} da matriz Ar. Matriz resíduo é a matriz $E_k=I-ArB_k$.

<u>Processo de refinamento de matriz inversa</u>

De $E_k = (I-ArB_k)$ decorre $Ar^{-1} = B_k(I-E_k)^{-1}$

Se $||E_k|| < 1$, pelo lema 1 $Sr^{(m)} \sim (I-E_k)^{-1}$

Uma aproximação de Ar^{-1} melhor do que B_k é dada por

$$B_{k+1} = B_k.Sr^{(m)} \tag{4}$$

<u>Método de refinamento de Segunda Ordem</u>

Fazendo-se m=1 em (4) obtemos

$$B_{k+1} = B_k(I+E_k) = B_k + B_kE_k$$

O resíduo associado à nova aproximação é $E_{k+1} = I-ArB_{k+1}$.
Daí, $E_{k+1} = I-Ar(B_k(I+E_k))=I-ArB_k-ArB_kE_k=E_k-ArB_kE_k=(I-ArB_k)E_k = E_k^2$

Portanto, $||E_{k+1}|| \leq ||E_k||^2$.

Se $||E_k|| < 1$, a sequência B_k, B_{k+1}, ..., converge com o quadrado do resíduo.

Método de refinamento de terceira ordem

Considerando-se m=2 em (4) resulta

$B_{k+1} = B_k(I+E_k+E_k^2) = B_k + B_k(E_k(I+E_k))$

O nôvo resíduo é $E_{k+1} = I - ArB_k = E_k^3$.

De fato, $E_{k+1} = I-Ar(B_k(I+E_k+E_k^2))$

$$E_{k+1} = I-ArB_k(I+(I-ArB_k)+(I-ArB_k)^2) =$$

$$I-ArB_k-ArB_k+ArB_kArB_k - ArB_k(I-ArB_k)^2 = I-2ArB_k+(ArB_k)^2-ArB_k(I-ArB_k)^2=$$

$$(I-ArB_k)^2 - ArB_k(I-ArB_k)^2 = (I-ArB_k)^2(I-ArB_k) = (I-ArB_k)^3 = E_k^3$$

Se $||E_k|| < 1$, a sequência $B_k, B_{k+1}, \ldots$, converge com o cubo do resíduo.

Métodos de k-ésima ordem

Usando mais têrmos do desenvolvimento em série de $(I-E_k)^{-1}$ obtemos, teòricamente, métodos de ordem cada vêz mais elevada.

Entretanto, devido à ocorrência dos êrros de arredondamento na execução das operações aritméticas, a precisão na solução aproximada irá depender não apenas de m, mas também da precisão da aritmética utilizada.

Por exemplo, no caso do método de refinamento de segunda ordem, dado um valor aproximado B_k de Ar^{-1}, a melhoria desta aproximação, dada por $B_k.E_k$, deve ser calculada com uma aritmética de precisão maior do que a usada na obtenção de B_k. A última etapa do método, que dá $B_{k+1} = B_k + B_k.E_k$, também deve ser efetuada usando uma aritmética de precisão maior.

Definição

Seja A uma matriz de intervalos com $A_{ij} = [a_{ij}, b_{ij}]$.

Verifica-se que

$$||A|| = \max_i \sum_j \max(|a_{ij}|, |b_{ij}|) \qquad (5)$$

é uma norma. Seja Ar uma matriz numérica e $||Ar|| = \max_i \sum_j |(Ar)_{ij}|$

Se $Ar \subset A$, então $||Ar|| \leq ||A||$, por decorrência imediata da definição (5).

Método de Hansen

Seja A uma matriz de intervalos e indiquemos seus elementos A_{ij} por $[a_{ij}, b_{ij}]$. Seja As uma matriz numérica fixada, pertencente a A (por exemplo, com seus elementos $(As)_{ij} = (a_{ij}+b_{ij})/2$) e B uma aproximação da matriz inversa As^{-1} de As.

Seja Ar uma matriz qualquer pertencente a A.

Consideremos os resíduos $Er=I-ArB$ e $E = I - AB$ (6)

Se $||E|| < 1$, sendo $Er \subset E$, $||Er|| < 1$ e pelo lema 1

$$||(I-Er)^{-1} - Sr^{(m)}|| \leq \frac{||Er||^{m+1}}{1-||Er||} \leq \frac{||E||^{m+1}}{1-||E||} = \rho \quad (7)$$

Seja $P^{(m)}$ a matriz de intervalos em que todos elementos são iguais a $[-\rho, \rho]$.

De (7), para todo i e j, $|((I-Er)^{-1}-Sr^{(m)})_{ij}| \leq \rho$ isto é, $-\rho \leq ((I-Er)^{-1} - Sr^{(m)})_{ij} \leq \rho$ e portanto, $((I-Er)^{-1} - Sr^{(m)})_{ij} \in [-\rho, \rho]$, donde a matriz real $(I-Er)^{-1} - Sr^{(m)} \subset P^{(m)}$, ou seja, $(I-Er)^{-1} \subset Sr^{(m)} + P^{(m)}$.

Multiplicando-se por B ambos os membros da última inclusão obtemos

$$Ar^{-1} = B(I-Er)^{-1} \subset B(S^{(m)} + P^{(m)}) \quad (8)$$

Tendo em vista que

$Sr^{(m)}=I+Er+Er^2+...+Er^m=((Er+I)Er+I)Er+...+I \subset ((E+I)E+I)E+...+I=S^{(m}$

como decorrência da propriedade de inclusão monotônica das operaçõe aritméticas com intervalos, de (8) vemos que

$$Ar^{-1} \subset B(S^{(m)} + P^{(m)}) \quad (9)$$

Portanto $\{Ar^{-1} \mid Ar \subset A\} \subset B(S^{(m)} + P^{(m)})$ (10)

De (10) e da proposição 1,

$$A^{-1} \subset B(S^{(m)} + P^{(m)}) \quad (11)$$

Avaliação do segundo membro de (11)

Faremos uso das seguintes abreviações:

a.r.p.s. = aritmética real (ponto flutuante) truncada em precisão simples.

a.i.p.s. = aritmética de intervalo com arredondamento em precisão simples.

a.i.p.d. = aritmética de intervalo com arredondamento em precisão dupla.

Considerando $S^{(m)}-I = (((E+I)E+I)E+...+I)$, escrevemos

$$B(S^{(m)} + P^{(m)}) = B + B(S^{(m)}-I + P^{(m)}) \tag{12}$$

Diagrama de blocos:para avaliação de (12):

-Leitura da matriz de intervalos A (a.i.p.s.)

-Definição de As com $(As)_{ij}=(a_{ij}+b_{ij})/2$. (a.r.p.s.)

-Cálculo de B, usando um método qualquer de inversão numérica, por exemplo, o que envolve o método de eliminação de Gauss. (a.r.p.s.)

-Cálculo de $E=I-AB$ com a.i.p.d., arredondando os resultados para precisão simples.

-Cálculo de $||E||$ (a.r.p.s.)

-Cálculo de um limite superior de ρ (a.r.p.s.)

-Definição de $P^{(m)}$ em precisão simples.

-Cálculo de $S^{(m)}-I$ (a.i.p.s.)

-Cálculo de $(S^{(m)}-I) + P^{(m)}$ (a.i.p.s.)

-Cálculo de $B.(S^{(m)}-I+P^{(m)})$ (a.i.p.s.)

-Cálculo de $B + B(S^{(m)}-I + P^{(m)})$ (a.i.p.d.)

Estimativa a priori do êrro

O segundo membro de (10) é uma matriz de intervalos que contém o conjunto de matrizes $\{Ar^{-1} \mid Ar \subset A\}$ e portanto contém a matriz inversa A^{-1} da matriz de intervalos A. Uma questão importante é responder de quanto diferem os intervalos correspondentes nas duas matrizes. Mais precisamente, sendo H uma matriz de intervalos e

$$B(S^{(m)}+P^{(m)}) = A^{-1} + H,$$

qual o maior valor de $|H_{ij}|$?

Moore demonstra em Interval Analysis, pg.37, que

$$|H_{ij}| \leq \frac{8(n+1)^2 ||B||^3}{1-(n+1)w(A)||B||} \cdot (w(A))^2 \qquad (13)$$

onde n é a ordem da matriz e $w(A) = \max\limits_{i,j} w(A_{ij})$

A (13) nos diz que cada extremo dos intervalos calculados estão afetados de um êrro proporcional ao quadrado de w(A), supondo-se o uso nos cálculos de uma aritmética com precisão - suficientemente elevada.

A escôlha de m em (10) depende:

- da precisão da aritmética.
- da magnitude de $||E||$.

Inversão de matriz por um método direto

O método de eliminação de Gauss para resolução de um sistema linear dá um método direto para inversão de matriz, baseado na seguinte propriedade: A solução do sistema $Ax = b_i$ é a coluna i-ésima da matriz A^{-1} inversa da matriz A, sendo b_i o vetor definido por $(o,o,\ldots,o,\underset{i}{1},o,\ldots,o)$. A subrotina para cálculo da inversa chama a subrotina DECMAV e, em seguida, definindo b_i, chama a subrotina RSTISV, gerando dessa forma os elementos da i-ésima coluna da matriz A^{-1}. A subrotina para inversão, denominada INVMV encontra-se no apêndice B.

Apresentaremos a seguir, a título de ilustração, programas em FORTRAN para cálculo de uma aproximação da matriz A^{-1} inversa de uma matriz de intervalos, sendo o primeiro o método de Hansen e o segundo o método direto baseado no processo de eliminação de Gauss. Para exemplificação consideramos a seguinte matriz de intervalos:

$$A = \begin{bmatrix} [0{,}999 \,,\, 1{,}010] & [-0{,}001 \,,\, 0{,}001] \\ [-0{,}001 \,,\, 0{,}001] & [0{,}999 \,,\, 1{,}010] \end{bmatrix}$$

PROGRAMA FORTRAN do METODO DE HANSEN

```
C     METODO DE HANSEN
      DOUBLE PRECISION AD,BD,ED,Z,ZZ,AA
      DIMENSION A(2,10,11),B(2,10,10),BD(2,10,10),E(2,10,10),F(2,10,1
     *    S(2,10,10),AR(10,10),BR(10,10),AA(2,10,10),AD(2,10,10),Z(2),
     *    ED(2,10,10)
      READ(5,100)N,M
      WRITE(6,129)
      DO 10 I=1,N
      READ(5,110)(A(1,I,J),A(2,I,J),J=1,N)
   10 WRITE(6,130)(A(1,I,J),A(2,I,J),J=1,N)
      DO 20 I=1,N
      DO 20 J=1,N
   20 AR(I,J)=A(1,I,J)/2.+A(2,I,J)/2.
      WRITE(6,131)
      CALL INVN(N,AR,BR)
      DO 30 I=1,N
      DO 30 J=1,N
      CALL DEFTN(B(1,I,J),BR(I,J),BR(I,J))
      BD(1,I,J)=B(1,I,J)
      BD(2,I,J)=B(2,I,J)
      AD(1,I,J)=A(1,I,J)
      AD(2,I,J)=A(2,I,J)
      CALL DEFTN(A(1,I,J),B(1,I,J),B(2,I,J))
   30 CALL DEFTND(BD(1,I,J),BD(1,I,J),BD(2,I,J))
      DO 25 I=1,N
   25 WRITE(6,152)(B(1,I,J),B(2,I,J),J=1,N)
C
      CALL PRVD(N,BD,AD)
C
      WRITE(6,132)
      DO 35 I=1,N
   35 WRITE(6,152)(ED(1,I,J),ED(2,I,J),J=1,N)
C
      DO 50 I=1,N
      DO 50 J=1,N
      CALL DEFTND(AD(1,I,J),BD(1,I,J),BD(2,I,J))
      IF(I-J)42,44,42
   42 ZZ=0.
      CALL DEFTND(Z,ZZ,ZZ)
      CALL SUBTD(Z,ED(1,I,J))
      CALL DEFTND(ED(1,I,J),Z(1),Z(2))
      GO TO 50
   44 ZZ=1.
      CALL DEFTND(Z,ZZ,ZZ)
      CALL SUBTD(Z,ED(1,I,J))
      CALL DEFTND(ED(1,I,J),Z(1),Z(2))
   50 CONTINUE
C
C
      DO 55 I=1,N
      DO 55 J=1,N
      E(1,I,J)=ED(1,I,J)
      E(2,I,J)=ED(2,I,J)
   55 CALL ARDT2(E(1,I,J),ED(1,I,J))
C
      WRITE(6,133)
      DO 41 I=1,N
   41 WRITE(6,152)(E(1,I,J),E(2,I,J),J=1,N)
C
C
```

```
      SOMAS=0.
      ANRMA=-1.
      DO 91 I=1,N
      IF(ANRMA-SOMAS)60,70,70
   60 ANRMA=SOMAS
   70 SOMAS=0.
      DO 91 J=1,N
      IF(ABS(E(1,I,J))-ABS(E(2,I,J)))80,80,90
   80 SOMAS=SOMAS+ABS(E(2,I,J))
      GO TO 91
   90 SOMAS=SOMAS+ABS(E(1,I,J))
   91 CONTINUE
      IF(ANRMA-SOMAS)92,93,93
   92 ANRMA=SOMAS
C
   93 IF(ANRMA-1.)310,300,300
  300 WRITE(6,153)ANRMA
      STOP
C
  310 DO 320 K=1,M
      WRITE(6,141)
      DO 350 I=1,N
      DO 350 J=1,N
  350 CALL DEFINO(BD(1,I,J),AD(1,I,J),AD(2,I,J))
      RO=(ANRMA**(K+1))/(1.-ANRMA)
      CALL ARCTO(RO)
      WRITE(6,140)ANRMA,K,RO
      CALL MPV(RO,N,P)
      WRITE(6,134)
      DO 311 I=1,N
  311 WRITE(6,152)(P(1,I,J),P(2,I,J),J=1,N)
      CALL MSV(N,K,E,S)
      WRITE(6,135)
      DO 312 I=1,N
  312 WRITE(6,152)(S(1,I,J),S(2,I,J),J=1,N)
      CALL SM2V(N,S,P)
      WRITE(6,136)
      DO 313 I=1,N
  313 WRITE(6,152)(S(1,I,J),S(2,I,J),J=1,N)
      CALL PRV(N,S,A)
      WRITE(6,137)
      DO 314 I=1,N
  314 WRITE(6,152)(S(1,I,J),S(2,I,J),J=1,N)
      CALL SM2VD(N,BD,S)
C
      DO 315 I=1,N
      DO 315 J=1,N
  315 CALL FCID(BD(1,I,J),AA(1,I,J))
      WRITE(6,139)
      DO 316 I=1,N
  316 WRITE(6,152)(AA(1,I,J),AA(2,I,J),J=1,N)
      WRITE(6,138)
      DO 320 I=1,N
  320 WRITE(6,152)(BD(1,I,J),BD(2,I,J),J=1,N)
      STOP
C
  100 FORMAT(2I2)
  110 FORMAT(4F6.3)
  129 FORMAT(1X,8HMATRIZ A,/)
  130 FORMAT(1X,2F12.8,4X,2F12.8)
```

```
  131 FORMAT(/,1X,8HMATRIZ B,/)
  132 FORMAT(/,1X,9HMATRIZ AB,/)
  133 FORMAT(/,1X,13HMATRIZ E=I-AB,/)
  134 FORMAT(/,1X,8HMATRIZ P,/)
  135 FORMAT(/,1X,8HMATRIZ S,/)
  136 FORMAT(/,1X,10HMATRIZ S+P,/)
  137 FORMAT(/,1X,13HMATRIZ B(S+P),/)
  138 FORMAT(/,1X,42HMATRIZ INVERSA B(S+P)+B(EM DUPLA PRECISAO),/)
  139 FORMAT(/,1X,33HMATRIZ INVERSA SOB FORMA CENTRADA)
  140 FORMAT(/,1X,12HNORMA DE E =,F10.6,5X,2HM=,I1,5X,3HRO=,F12.8)
  141 FORMAT(1H1)
  152 FORMAT(1X,2F12.8,4X,2F12.8)
  153 FORMAT(1X,6HANORMA=,E16.8)
      END

09/14/70     8:23 AM     ASR#3.2     69157  COMPILER
  0 MIN 28 SEC FOR COMPILATION PASS
  135 CARDS AT 289 CARDS PER MINUTE
 17666 DIGITS DATA, 20040 DIGITS CODE.


                 SYMBOLIC LISTING FOR PROGAM :

                 LOW ADRS  HIGH ADRS TEMPS      DATA BASE CODE BASE
                  002028    039740    002046     00208C    0197C0

                 CALLED SUBPROGRAMS :
                 NAME     ADDRESS   SEG. NO.
                 INVM      084974     001
                 DEFIN     073010     001
                 DEFINO    064084     001
                 PRVC      079160     001
                 SUBTO     077748     001
                 ARDT2     068872     001
                 ARDTO     069670     001
                 MPV       077198     001
                 MSV       074780     001
                 SM2V      073190     001
                 PRV       064262     001
                 SM2VO     061120     001
                 FOID      058038     001
                 READ.     052548     001
                 WRITE.    045046     001
                 EXPON.    039740     001

                 COMMON BLOCKS REFERENCED :
                 NAME    ADDRESS
                 NONE
```

EXEMPLO [illegible]

```
MATRIZ A

   .99900000  1.01000000        -.00100000   .00100000
  -.00100000   .00100000         .99900000  1.01000000

MATRIZ B

  1.00000000  1.00000000         .00000000   .00000000
   .00000000   .00000000        1.00000000  1.00000000

MATRIZ AB

   .99900000  1.01004000        -.00100002   .00100002
  -.00100003   .00100004         .99900000  1.01002000

MATRIZ E=I-AB

  -.01010000   .00101000        -.00101000   .00101000
  -.00101000   .00101000        -.01010000   .00101000


NORMA DE E =   .01110      N=1      RO=    .00012500

MATRIZ F

  -.00012500   .00012500        -.00012500   .00012500
  -.00012500   .00012500        -.00012500   .00012500

MATRIZ S

  -.01010000   .00101000        -.00101000   .00101000
  -.00101000   .00101000        -.01010000   .00101000

MATRIZ S+F

  -.01030000   .00114000        -.00114000   .00114000
  -.00114000   .00114000        -.01030000   .00114000

MATRIZ B(S+F)

  -.01030000   .00114000        -.00114000   .00114000
  -.00114000   .00114000        -.01030000   .00114000

MATRIZ INVERSA SOB FORMA CENTRADA
   .99543000   .00573000         .00000000   .00114001
   .00000000   .00114001         .99543000   .00573000

MATRIZ INVERSA B(S+F)+B(EM DUPLA PRECISAO)

   .98970000  1.00116000        -.00114001   .00114001
  -.00114001   .00114001         .98970000  1.00116000
```

PROGRAMA EM FORTRAN PARA INVERSÃO DE MATRIZ POR UM METODO DIRETO

```
      DIMENSION A(3,20,21),KP(20),B(3,20,21)
 1000 READ(5,200,END=2000)N
      WRITE(6,700)
      DO 1 I=1,N
      READ(5,300)A(1,I,1),A(3,I,1),A(1,I,2),A(3,I,2)
      A(2,I,1)=A(1,I,1)/2.+A(3,I,1)/2.
      A(2,I,2)=A(1,I,2)/2.+A(3,I,2)/2.
    1 WRITE(6,400)((A(K,I,J),K=1,3),J=1,N)
      CALL INVMV(N,A,B)
      WRITE(6,800)
      DO 130 I=1,N
      DO 130 J=1,N
  130 CALL FCI(B(1,I,J),A(1,I,J))
      DO 140 J=1,N
      WRITE(6,135)J
      DO 140 I=1,N
  140 WRITE(6,145)A(1,I,J),A(2,I,J),A(3,I,J)
      WRITE(6,141)
      GO TO 1000
 2000 STOP
  135 FORMAT(/,1X,6HCOLUNA,I5)
  141 FORMAT(1H1)
  145 FORMAT(/,2X,3E15.6)
  200 FORMAT(I2)
  300 FORMAT(4F6.3)
  400 FORMAT(/,1X,6F9.3)
  700 FORMAT(//,1X,11HMATRIZ DADA,27H - DADOS COM ERROS INICIAIS,///)
  800 FORMAT(//,1X,31HMATRIZ INVERSA - FORMA CENTRADA,//)
      END

12/05/70      9≥34 AM     ASR#3.2       69157   COMPILER
  0 MIN  9 SEC FOR COMPILATION PASS
   30 CARDS AT 187 CARDS PER MINUTE
 25752 DIGITS DATA.  4470 DIGITS CODE.



      SYMBOLIC LISTING FOR PROGAM ≥

      LOW ADRS  HIGH ADRS TEMPS     DATA BASE CODE BASE
       002044    032266    002056    002092    027796

      CALLED SUBPROGRAMS ≥
      NAME     ADDRESS   SEG. NO.
      INVMV    052134     001
      FCI      045258     001
      READ.    039768     001
      WRITE.   032266     001

      COMMON BLOCKS REFERENCED ≥
      NAME   ADDRESS
      NONE
```

EXEMPLO

```
MATRIZ DADA - DADOS COM ERROS INICIAIS


   .999     1.005     1.010    -.001      .000      .001
  -.001      .000      .001     .999     1.005     1.010

MATRIZ INVERSA - FORMA CENTRADA


COLUNA     1
     .995558E+00     .995520E+00     .546151E-02
     .000000E-99     .000000E-99     .100202E-02
COLUNA     2
     .000000E-99     .000000E-99     .100202E-02
     .995549E+00     .995520E+00     .546051E-02
```

Resultados numéricos

O resultado numérico dos exemplos que apresentaremos a seguir encontram-se no apêndice C e os programas correspondentes no apêndice B. O método direto para inversão utiliza a aritmética - de ternos arredondada e o programa correspondente foi processado com size real = 12. O método de Hansen utiliza a aritmética de intervalos arredondada e o programa correspondente foi processado com size real = 6, uma vez que utiliza precisão dupla no processo de refinamento.

Calculamos uma aproximação da matriz inversa de duas matrizes A (exemplo 4) e B (exemplo 5), usando um método direto de inversão (exemplos 4-5-6-7) e o método de Hansen (exemplos 8-9-10-11), supondo os coeficientes exatos (exemplos 4-5-8-9) e afetados de êrros iniciais não superiores a 0,00005 (exemplos 6-7-10-11).

Coeficientes exatos

Comparando-se os resultados numéricos dos exemplos 4 e 5 (método direto) com os dos exemplos 8 e 9 (método de Hansen), em que supusemos os coeficientes exatos, verificamos que o método - direto pode dar uma solução melhor. De fato, no caso dos exemplos 4 e 8, a ordem de grandeza do êrro máximo cometido no método de Hansen é 10^{-9}, enquanto que no método direto é da ordem 10^{-10}. No caso da matriz dada nos exemplos 5 e 9 a ordem de grandeza do êrro máximo é a mesma, isto é, 10^{-6}

É interessante observar que a discrepância entre a ordem de grandeza do êrro máximo cometido no cálculo de uma aproximação da matriz inversa das matrizes A (exemplo 4) e B (exemplo 5), ambas matrizes de quarta ordem, resolvidas por um mesmo processo e usando mesma precisão na execução das operações aritméticas, nos sugere que a matriz B deve ser mal condicionada.

Coeficientes afetados de êrros iniciais

Os resultados numéricos dos exemplos 6-7 (método direto) e 10-11 (método de Hansen) mostram uma pequena melhoria nos valores aproximados obtidos pelo método de Hansen.

De fato, no cálculo de uma aproximação da inversa da matriz A, usando o método direto (exemplo 6) obtivemos um êrro máximo $\doteq 0,8410534 \times 10^{-4}$ e usando o método de Hansen (exemplo 10), um êrro máximo igual a $0,82928901 \times 10^{-4}$. Essa diferença, entretanto é maior no caso da matriz B (possìvelmente mal condicionada) em que obtivemos para o êrro máximo cometido os valores $0,224570967166 \times 10^{1}$ (exemplo 7) e $0,2552475001 \times 10^{0}$ (exemplo 11).

Escôlha de m no método de Hansen

Nos exemplos 10 e 11 calculamos uma aproximação da matriz inversa das matrizes A e B, considerando os valores de m de 1 a 6. Verificamos que para $m \geqslant 4$ os valores não se modificaram e que a diferença entre os valores correspondentes para m=3 e m=4 é insignificante, apesar de, no caso da matriz A termos $||E||=0{,}147175x10^{-}$ e para a matriz B, $||E|| = 0{,}829016x10^{-2}$.

Pela análise dos resultados é de se esperar que um valor ade quadro para m é a métade do size real utilizado em precisão simples

§ 13 - Solução de Equações Algébricas

Introdução

Seja X um intervalo, $g: X \longrightarrow R$ e $\bar{x} \in X$ uma raíz de $g(x)=0$. Escrevendo $g(x) = x - f(x)$, verificamos que $\bar{x}$ é um ponto fixo da aplicação $f: X \longrightarrow R$.

Dado $x_o \in X$ suponhamos que se possa construir a sequência

$$x_{n+1} = f(x_n) \tag{1}$$

Convergência da sequência (1)

Uma condição para que a sequência (1) seja convergente para a raíz $\bar{x}$ de $g(x)=0$, é que a aplicação f seja uma contração, isto é, que exista um número real k, $o < k < 1$, tal que

$$|f(x)-f(y)| \leq k|x-y| \tag{2}$$

quaisquer que sejam x,y de X. De fato, se a (2) estiver verificada deduz-se fàcilmente que $|\bar{x} - x_n| \leq k^n|\bar{x} - x_o|$ e como $k^n|\bar{x} - x_o| \longrightarrow 0$ para $n \longrightarrow \infty$, concluímos que $x_n \longrightarrow \bar{x}$.

Lema 1

Seja $f: X = [a,b] \longrightarrow R$. Se f e f' são contínuas em X e se $\max\limits_{x \in X}|f'(x)| < 1$, então f é uma contração.

Demonstração:

Dados dois pontos quaisquer x e y de X, pelo teorema do valor médio existe z, $a \leq x < z < y \leq b$ tal que

$$f(x) - f(y) = f'(z)(x - y)$$

Tomando-se o módulo e majorando-se $|f'(z)|$ por $k=\max\limits_{x \in X}|f'(x)|$ obtemos $|f(x)-f(y)| \leq k|x - y|$, donde decorre que f é uma contração, pois $k < 1$ por hipótese, q.e.d.

Método de Newton

Seja $g: X=[a,b] \longrightarrow R$, $g \in C^1(X)$, $\bar{x} \in X$ uma raíz simples de $g(x)=0$, isto é, não é raíz simultâneamente de g e g'.

Vamos procurar uma particular transformação de $g(x)=0$ em que $x = f(x)$, de modo que f seja uma contração. Pelo lema 1 vimos que f é uma contração se $\max_{x \in X} |f'(x)| < 1$. Observamos que se $f'(\bar{x}) = 0$, da continuidade de $f'(x)$ resulta $|f'(x)| < 1$ para todo x pertencente a uma vizinhança conveniente de $\bar{x}$.

Consideremos $f(x) = x + A(x)g(x)$ e procuremos determinar $A(x)$ de modo que $f'(\bar{x}) = 0$. Derivando,

$$f'(x) = 1 + A(x)g'(x) + A'(x)g(x)$$

Como $g(\bar{x}) = 0$ e $g'(\bar{x}) \neq 0$ resulta $A(\bar{x}) = -1/g'(\bar{x})$.

Uma função $A(x)$ que satisfaz essa condição é $A(x)=-1/g'(x)$.

Portanto,

$$f(x) = x - g(x)/g'(x) \tag{3}$$

Algoritmo

Seja x_0 um ponto qualquer do intervalo $X = [a,b]$, $\bar{x} \in X$ uma raíz simples de $g(x) = 0$.

O algoritmo de Newton, na análise real, para construção da sequência (1) é o seguinte:

$$\begin{aligned} &x_0 \quad \text{arbitrário} \\ &x_{n+1} = f(x_n) = x_n - g(x_n)/g'(x_n) \end{aligned} \tag{4}$$

Convergência

O método de Newton é um método de segunda ordem, isto é, a sequência x_n construída pelo algoritmo (4) converge para $\bar{x}$ quadràticamente, isto é, $(x_{n+1} - \bar{x}) = 0(x_n - \bar{x})^2$.

De fato, seja x_n um ponto qualquer da sequência e consideremos o desenvolvimento de Taylor em torno de $\bar{x}$:

$$f(x_n) = f(\bar{x}) + (x_n-\bar{x})f'(\bar{x}) + 0(x_n-\bar{x})^2$$

Como $f'(\bar{x}) = 0$ e $f(\bar{x}) = \bar{x}$ resulta

$x_{n+1} = \bar{x} + O(x_n-\bar{x})^2$, donde $(x_{n+1} - \bar{x}) = O(x_n - \bar{x})^2$

Contração de Intervalos

Usaremos o conceito de contração para construir uma sequência de intervalos que seja convergente ao ponto fixo $\bar{x}$ de $f(x)$.

Definição

Seja Y um intervalo. A função de intervalos

$$G : \mathcal{F}_Y \longrightarrow \mathcal{F}$$

é uma contração se $G(X) \subset X$ para todo intervalo X de $\mathcal{F}_Y$.

A contração G se diz uma contração forte de intervalos se existe um número real k, $0 < k < 1$, tal que

$$w(G(X)) \leq k w(X)$$

Proposição 1

Seja $f: X = [a,b] \longrightarrow R$, $\bar{x} \in X$ um ponto fixo de f e

$$F : \mathcal{F}_X \longrightarrow \mathcal{F} \text{ uma extensão de } f$$

Se F é uma contração forte de intervalos, então a sequência

$$X_o = X$$
$$X_{n+1} = F(X_n)$$

pode ser construída e converge para $\bar{x}$.

Demonstração:

Se F é uma contração de intervalos a sequência X_n pode ser construída, pois

$$X_o \supset X_1 \supset X_2 \supset \ldots \supset X_n \supset \ldots$$

Se $\bar{x} \in X_n$, então $f(\bar{x}) \in \bar{F}(X_n) \subset F(X_n) = X_{n+1}$. Como $\bar{x} = f(\bar{x})$ concluímos que $\bar{x} \in X_{n+1}$. Portanto, $\bar{x} \in X_n$, qualquer que seja o inteiro n.

Sendo F uma contração forte de intervalos existe k,

$0 < k < 1$, tal que $w(X_n) \leq k^n w(X_o)$.

Portanto, $w(X_n) \longrightarrow o$ para $n \longrightarrow \infty$.

Seja $X_n = [a_n, b_n]$. Como $d(X_n, [\bar{x},\bar{x}]) = \max([a_n-\bar{x}], [b_n-\bar{x}]) \leq b_n - a_n = w(X_n)$, por $\bar{x} \in [a_n, b_n]$, concluímos que $X_n \longrightarrow [\bar{x},\bar{x}]$.

Método de Newton para Intervalos

Seja $g: X = [a,b] \longrightarrow R$, $\bar{x} \in X$ uma raíz de $g(x) = 0$, $g \in C^1(X)$, g uma função racional, G e G' extensões racionais de g e g', respectivamente.

A versão ingênua do algoritmo de Newton para intervalos consiste simplesmente na substituição em (4) dos números reais por intervalos, dando origem ao seguinte algoritmo:

$$\begin{aligned} &X_o \text{ arbitrário}, \qquad X_o \ni \bar{x} \\ &X_{n+1} = X_n - G(X_n)/G'(X_n) \end{aligned} \qquad (5)$$

Entretanto, esta sequência de intervalos, construída a partir de um X_o arbitrário, $X_o \neq [\bar{x},\bar{x}]$ não converge para $[\bar{x},\bar{x}]$.

De fato, sendo $w(X_{n+1}) = w(X_n - G(X_n)/G'(X_n)) =$

$w(X_n) + w(G(X_n)/G'(X_n))$ e $w(G(X_n)/G'(X_n)) > 0$, vemos que $w(X_{n+1}) > w(X_n)$, o que mostra ser impossível a convergência dos extremos dos intervalos para a raíz $\bar{x}$ de $g(x)$.

Antes de estudarmos uma versão do método de Newton para intervalos que produza uma sequência de intervalos convergente, veremos alguns lemas.

Lema 2

Seja X um intervalo, $g: X \longrightarrow R$ uma função racional, $g \in C^1(X)$, $\bar{x} \in X$ uma raíz simples de $g(x) = 0$ e G' uma extensão racional de g'. Dado um ponto $\tilde{x}$ de X consideremos o intervalo

$$N(X) = \tilde{x} - g(\tilde{x})/G'(X)$$

Uma condição necessária e suficiente para que o intervalo $N(X)$ esteja definido é que $0 \notin G'(X)$.

Prova imediata.

Lema 3

Seja X um intervalo, $g: X \longrightarrow R$ uma função racional, $g \in C^1(X)$ $\bar{x} \in X$ uma raiz de $g(x)=0$. G' uma extensão racional de g'. Dado um ponto $\tilde{x}$ de X, consideremos o intervalo

$$N(X) = \tilde{x} - g(\tilde{x})/G'(X)$$

Se $0 \notin G'(X)$, então $\bar{x} \in N(X)$.

Demonstração:

Se $\tilde{x} = \bar{x}$ resulta $N(X) = \bar{x}$. Caso contrário, suponhamos $\tilde{x} > \bar{x}$ e consideremos o intervalo $[\bar{x},\tilde{x}]$. Da continuidade de g' em $[\bar{x},\tilde{x}]$ e do teoremos do valor médio, existe $z \in [\bar{x},\tilde{x}] \subset X$ tal que $g(\tilde{x}) = g'(z)(\tilde{x} - \bar{x})$.

Portanto, $\bar{x} = \tilde{x} - g(\tilde{x})/g'(z), \quad z \in X$

Como $g'(z) \in \bar{g}'(X) \subset G'(X)$, concluímos que $\bar{x} \in N(X)$, por serem monotônicas as operações aritméticas com intervalos. (Se G' é uma função de intervalos racional, G' é monotônica).

No caso $\tilde{x} < \bar{x}$ a demonstração é semelhante.

Lema 4

Seja X um intervalo, $g:X \longrightarrow R$ uma função racional, $g \in C^1(X$ G' uma extensão racional de g' e $0 \notin G'(X)$. Dado um ponto $\tilde{x}$ de X consideremos o intervalo

$$N(X) = \tilde{x} - g(\tilde{x})/G'(X)$$

Se $N(X) \subset X$, então existe $\bar{x} \in N(X)$ tal que $g(\bar{x}) = 0$.

Se $0 \notin G'(X)$, o intervalo $G'(X) > 0$ ou $G'(X) < 0$. Suponhamos $G'(X) > 0$ e indiquemos $G'(X)$ por $[c,d]$. Se $g(\tilde{x}) = 0$, o lema está demonstrado. Suponhamos $g(\tilde{x}) > 0$. Para todo $x < \tilde{x}$, aplicando o teorema do valor médio obtemos

$$g(\tilde{x}) - g(x) = g'(z)(\tilde{x}-x), \qquad x < z < \tilde{x}$$

Como $c < g'(z) < d$, $c(\tilde{x}-x) < g(\tilde{x}) - g(x) < d(\tilde{x}-x)$, donde

$$d(x-\tilde{x}) + g(\tilde{x}) < g(x) < c(x-\tilde{x}) + g(\tilde{x})$$

o que significa que a $g(x)$ permanece entre as retas

$$y_1 = c(x-\tilde{x}) + g(\tilde{x})$$

$$y_2 = d(x-\tilde{x}) + g(\tilde{x})$$

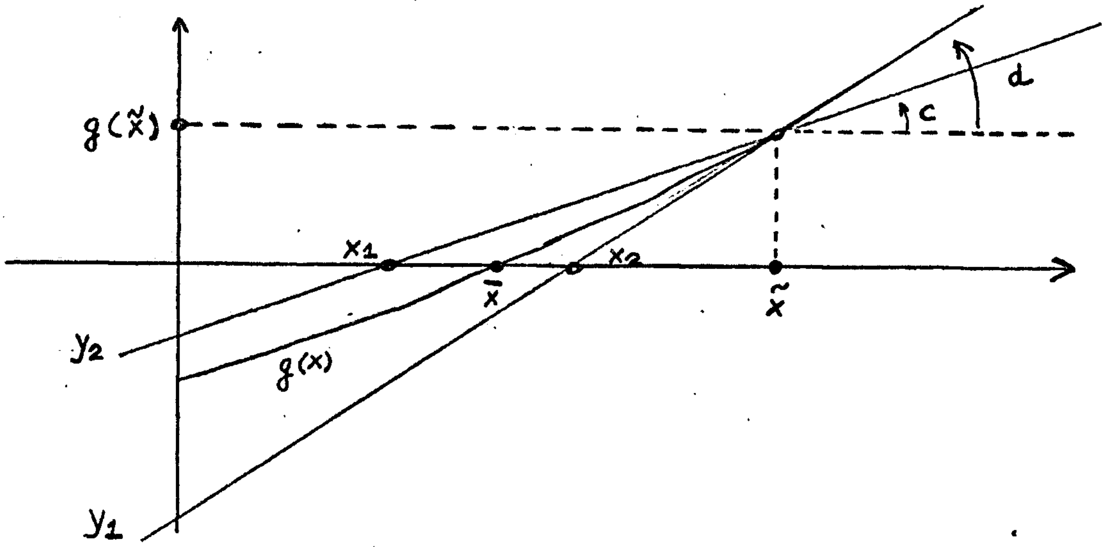
para todo $x < \tilde{x}$, conforme ilustra a figura da página seguinte.

O ponto de abcissa x_1 onde se anula y_1 é obtido resolvendo-se a equação $c(x-\tilde{x}) + g(\tilde{x}) = o$, cuja solução é $x_1 = \tilde{x} - g(\tilde{x})/c$. O ponto x_2 em que $y_2 = o$ é calculado de modo análogo e vale $x_2 = \tilde{x} - g(\tilde{x})/d$.

Portanto $g(x) = o$ para algum ponto $\bar{x}$ do intervalo $[x_1, x_2]$.

O intervalo $[x_1, x_2]$ está contido em X como decorrência da hipótese $N(X) \subset X$. Como $\bar{x} \in X$ e $G'(X)$ é uma extensão monotônica, usando o lema 3 concluímos que $\bar{x} \in N(X)$.

No caso em que $G'(X) < o$ demonstra-se de modo análogo.



Observação

Na prática, devido aos êrros de arredondamento, obtemos um intervalo $N_1(X) \supset N(X)$. Porém, se ocorrer $N_1(X) \subset X$, então $\bar{x} \in N_1(X)$, tendo em vista que $N(X) \subset N_1(X) \subset X$.

Considerando $N: \mathcal{J}_X \longrightarrow \mathcal{J}$, é interessante observar que $\bar{x}$ não é um ponto fixo de N, a menos que $\tilde{x} = \bar{x}$.

Uma versão do Método de Newton para Intervalos

Seja $X = [a,b] \longrightarrow R$ uma função racional, $g \in C^1(X)$, $\bar{x} \in X$ uma raíz de $g(x) = 0$, G' uma extensão racional de g'.

No método de Newton para números reais

$$\bar{x} \sim f(\tilde{x}) = \tilde{x} - g(\tilde{x})/g'(\tilde{x}), \quad \tilde{x} \in X$$

Entretanto, aplicando-se o teorema do valor médio para dois pontos $\tilde{x}$ e $\bar{x}$ de X, obtemos para todo $\bar{x} > \tilde{x}$

$$\bar{x} = \tilde{x} - g(\tilde{x})/g'(z), \quad a \leq \bar{x} < z < \tilde{x} \leq b$$

Consideremos o intervalo

$$N(X) = \tilde{x} - g(\tilde{x})/G'(X), \quad 0 \notin G'(X) \tag{6}$$

Pelo lema 3 a raíz $\bar{x} \in N(X)$.

O segundo membro de (6) é o segundo membro de (5) em que se substituiu o número real $g'(x)$ pelo intervalo $G'(X)$.

A igualdade (6) dá origem ao seguinte algoritmo de Newton - para intervalos:

$$X_0 \text{ arbitrário}, \; X_0 \ni \bar{x}$$
$$X_{n+1} = N(X_n) = \tilde{x}_n - g(\tilde{x}_n)/G'(X_n), \tag{7}$$

onde $0 \notin G'(X_n)$ e o ponto $\tilde{x}_n$ escolhido de uma forma bem definida.

Escôlha de $\tilde{x}_n$

Usando-se a aritmética de intervalos arredondada adota-se para $\tilde{x}_n$ o ponto médio do intervalo X_n (pg.60 de Interval Analysis, R.E.Moore).

Entretanto, se fôr usada a aritmética de ternos arredondada, $\tilde{x}_n$ é tomado como o elemento intermediário do terno $[\underline{x},\tilde{x},\bar{x}]$ (pg.19 de Topics in Interval Analysis, E.R.Hansen). Neste caso o processamento do algoritmo (7) gera a sequência de números reais definida pelo algoritmo (4) do método de Newton para números reais e esta, conforme vimos, converge quadràticamente para $\bar{x}$.

Convergência

Suponhamos que a sequência (7) seja construída usando-se a aritmética de ternos. Neste caso a convergência (quadrática) desta sequência de intervalos para a raíz $\bar{x}$ de $g(x)=0$ decorre da convergência (quadrática) da sequência $\tilde{x}_n \longrightarrow \bar{x}$.

De fato, a sequência $X_n \longrightarrow [\bar{x},\bar{x}]$ se $d(X_n,[\bar{x},\bar{x}]) \longrightarrow 0$ para $n \longrightarrow \infty$. Como $d(X_n, [\bar{x},\bar{x}]) \leq w(X_n)$, basta provarmos que $w(X_n) \longrightarrow 0$ para $n \longrightarrow \infty$.

Já vimos pelo lema 3 que $\bar{x} \in X_n$, qualquer que seja n.

Por outro lado $w(X_{n+1}) = |g'(\tilde{x}_n)|.w(1/G'(X_n))$.

Supondo-se $1/G'(X_n)$ limitada, temos $w(1/G'(X_n)) \leq K$, $K > 0$.

Portanto, $w(X_{n+1}) \leq K.|g(\tilde{x}_n)|$ e como $|g(\tilde{x}_n)|$ converge quadràticamente para zero para $n \longrightarrow \infty$, vemos que

$w(X_{n+1}) \longrightarrow 0$ para $n \longrightarrow \infty$ (quadràticamente), q.e.d.

Programa em FORTRAN do algoritmo de Newton para Ternos

Para ilustrar o método apresentaremos um programa redigido na linguagem FORTRAN que utiliza as subrotinas de ternos descritas nos apêndices A e B. O programa calcula um valor aproximado da ra raíz da equação $g(x) = x - (1-x^2)/(3+x^2) = 0$ no intervalo $[-1,1]$. onde a mesma possui uma única raíz simples.

Adotamos $X_o = [-4/10, 4/10]$ e programamos o cálculo de 10 iterações. Verificamos, após a 5a. iteração, que os resultados se estabilizaram. A convergência de cada elemento do terno é òbviamente quadrática. Usando a forma centrada do intervalo podemos escrever:

$$\bar{x} = 0{,}29559774 \pm 0{,}00000001$$

É interessante observar que após a primeira iteração resultou $X_1 \not\subset X_o$.

Programa do Método de Newton para ternos

```
      DIMENSION X(3),Y(3),FY(3),FLX(3)
      WRITE(6,300)
      READ(5,100)X(1),X(2),X(3),M
      N=-1
      DO 10 I=1,M
      N=I-1
      WRITE(6,200)N,X(1),X(2),X(3)
      CALL DEFIN(Y,X(2),X(2),X(2))
      CALL F(Y,FY)
      CALL FL(X,FLX)
      CALL DIVD(FY,FLX)
      CALL SUBT(Y,FY)
      CALL DEFIN(X,Y(1),Y(2),Y(3))
   10 CONTINUE
      STOP
  100 FORMAT(3F8.3,I3)
  200 FORMAT(/,1X,I3,3E20.8)
  300 FORMAT(/,1X,3H  N,8X,4HXINF,16X,4HXINT,16X,4HXSUP)
  400 FORMAT(1X,3E20.8,/)
      END

10/15/70      9≥12 AM    ASR#3.2     69157  COMPILER
  0 MIN 23 SEC FOR COMPILATION PASS
   21 CARDS AT 052 CARDS PER MINUTE
   354 DIGITS DATA.   824 DIGITS CODE.



     SYMBOLIC LISTING FOR PROGAM ≥

     LOW ADRS  HIGH ADRS TEMPS     DATA BASE CODE BASE
      002054    003224    002054    002072    002400

     CALLED SUBPROGRAMS ≥
     NAME     ADDRESS   SEG. NO.
     DEFIN     030938     001
     F         031182     001
     FL        026740     001
     DIVD      023942     001
     SUBT      016214     001
     READ.     010724     001
     WRITE.    003224     001

     COMMON BLOCKS REFERENCED ≥
     NAME   ADDRESS
     NONE
```

Subrotina para calcular a função $x-(1-x^2)/(3+x^2)$

```
      SUBROUTINE F(X,FX)
      DIMENSION X(3),FX(3),Y(3),Z(3),W(3)
      CALL DEFIN(Y,X(1),X(2),X(3))
      CALL PROD(Y,X)
      CALL DEFIN(Z,1.,1.,1.)
      CALL SUBT(Z,Y)
      CALL DEFIN(W,3.,3.,3.)
      CALL SOMA(W,Y)
      CALL DIVD(Z,W)
      CALL DEFIN(FX,X(1),X(2),X(3))
      CALL SUBT(FX,Z)
      RETURN
      END

10/15/70      9≥12 AM     ASR#3.2      69157   COMPILER
  0 MIN 12 SEC FOR COMPILATION PASS
   14 CARDS AT 065 CARDS PER MINUTE
   236 DIGITS DATA.   864 DIGITS CODE.
```

Nesta subrotina usa-se a aritmética de ternos arredondada e o valor da função calculada no terno X é guardado no terno FX

Subrotina para calcular a função $1 + 8x/(3+x^2)^2$.

```
      SUBROUTINE FL(X,FLX)
      DIMENSION X(3),FLX(3),Y(3),Z(3),W(3)
      CALL DEFIN(FLX,1.,1.,1.)
      CALL DEFIN(Y,X(1),X(2),X(3))
      CALL DEFIN(Z,8.,8.,8.)
      CALL DEFIN(W,3.,3.,3.)
      CALL PROD(Z,X)
      CALL PROD(Y,X)
      CALL SOMA(W,Y)
      CALL PROD(W,W)
      CALL DIVD(Z,W)
      CALL SOMA(FLX,Z)
      RETURN
      END

10/15/70      9≥12 AM     ASR#3.2      69157   COMPILER
  0 MIN  7 SEC FOR COMPILATION PASS
   15 CARDS AT 113 CARDS PER MINUTE
   272 DIGITS DATA.   742 DIGITS CODE.
```

Nesta subrotina usa-se a aritmética de ternos arredondada. O valor da função calculada no terno X é guardado no terno FLX

Observamos que esta função é a derivada de $x-(1-x^2)/(3+x^2)$.

N	XINF	XINT	XSUP
0	-.40000000E+00	.00000000E-99	.40000000E+00
1	.23864980E+00	.33333333E+00	.55255948E+00
2	.29279793E+00	.29600000E+00	.30100613E+00
3	.29559676E+00	.29559778E+00	.29559925E+00
4	.29559773E+00	.29559773E+00	.29559776E+00
5	.29559773E+00	.29559773E+00	.29559775E+00
6	.29559773E+00	.29559773E+00	.29559775E+00
7	.29559773E+00	.29559773E+00	.29559775E+00
8	.29559773E+00	.29559773E+00	.29559775E+00
9	.29559773E+00	.29559773E+00	.29559775E+00
10	.29559773E+00	.29559773E+00	.29559775E+00

Método de Newton Generalizado

Seja $g\colon X = [a,b] \longrightarrow R$ uma função racional, $g \in C^1(X)$, $\bar{x} \in X$ uma raiz simples de $g(x) = 0$, G' uma extensão racional de g'. Para cada Y de $\mathcal{F}_X$ vamos escolher um ponto $\tilde{y}$ de uma maneira bem definida, conforme seja usada na computação a aritmética de intervalo ou a aritmética de termo. Consideremos a função de intervalo $N\colon \mathcal{F}_X \longrightarrow \mathcal{F}$ tal que para todo Y de $\mathcal{F}_X$

$$N(Y) = \tilde{y} - g(\tilde{y})/G'(Y), \qquad 0 \notin G'(Y)$$

O algoritmo do método de Newton Generalizado consiste em definir a sequência de intervalos da seguinte forma:

$$\begin{aligned} &X_0 \text{ arbitrário}, \quad X_0 \supseteq \bar{x} \\ &X_{n+1} = N(X_n) \cap X_n \end{aligned} \qquad (8)$$

A aplicação $T\colon \mathcal{F}_X \longrightarrow \mathcal{F}$, em que $T(Y) = N(Y) \cap Y$ para todo Y de $\mathcal{F}_X$ é, pela própria definição, uma contração. Pode-se provar que se $w(X)$ é suficientemente pequeno a aplicação T é uma contração forte e, portanto, pela proposição 1, a sequência X_n converge para $[\bar{x},\bar{x}]$.

A proposição 2, que veremos a seguir, além de mostrar que a sequência X_n converge, dá uma medida dessa convergência.

Proposição 2

Se $w(X_0)$ fôr suficientemente pequeno, a sequência (8) do método de Newton Generalizado converge quadràticamente para $\bar{x}$, isto é, existe um número $K > 0$, tal que $w(X_{n+1}) \leq K(w(X_n))^2$.

Demonstração

Seja Z um intervalo contido em X e consideremos $\tilde{x} \in Z$, e

$$N(Z) = \tilde{x} - g(\tilde{x})/G'(Z)$$

Como $o \notin G'(Z)$ podemos supor, sem perda de generalidade, $G'(Z) > o$, para todo $Z \subset X$.

Vejamos antes que as relações (9) e (10), expostas a seguir, são verdadeiras.

$$|g(\tilde{x})| \leq |G'(Z)|.w(Z) \tag{9}$$

Seja $Z = [x_1, x_2] \subset X$, $\tilde{x}$, $\overline{x} \in Z$. Supondo $\overline{x} \neq \tilde{x}$, consideremos $\overline{x} < \tilde{x}$. Pelo teorema do valor médio

$$g(\tilde{x}) - g(\overline{x}) = g'(z)(\tilde{x}-\overline{x}), \qquad \overline{x} < z < \tilde{x}$$

Como $g(\overline{x}) = o$, $|g(\tilde{x})| = |g'(z)|.|\tilde{x}-\overline{x}|$

Estando $[\tilde{x}-\overline{x}] \subset Z$, $\tilde{x}-\overline{x} \leq x_2 - x_1 = w(Z)$, donde

$$|g(\tilde{x})| \leq |g'(z)|.w(Z)$$

Por outro lado, para todo z de Z, $g'(z) \in \overline{g}'(Z) \subset G'(Z)$, e portanto, $|g'(z)| \leq \max\limits_{z \in Z} |g'(z)| \leq |G'(Z)|$. Substituindo esta majoração na igualdade precedente temos

$$|g(\tilde{x})| \leq |G'(Z)|.w(Z)$$

$$G(Z) \subset G(\tilde{x}) + K_1[-w(Z), w(Z)] \tag{10}$$

onde supomos G extensão de g, $\tilde{x} \in Z \subset X$. Pelo corolário 4.1 (§6 do Cap.1) temos $w(G(Z)) \leq K_1 w(Z)$. Daquí vemos que

$$G(Z) \subset [G(\tilde{x})-K_1 w(Z),\ G(\tilde{x})+K_1 w(Z)] = G(\tilde{x}) + K_1[-w(Z), w(Z)]$$

Aplicando o resultado (10) para a função de intervalos contínua $G'(Z)$ e $\tilde{x} = \overline{x}$, obtemos

$$G'(Z) \subset G'(\overline{x}) + K_1[-w(Z), w(Z)] \tag{11}$$

Da definição de $N(Z)$, de (11) e da propriedade monotônica das operações aritméticas com intervalos resulta

$$N(Z) \subset \tilde{x} - g(\tilde{x})/(G'(\overline{x}) + K_1[-w(Z), w(Z)]) \tag{12}$$

Para <u>w(Z) suficientemente pequeno</u> podemos admitir que $G'(\overline{x}) - K_1 w(Z) > o$, uma vêz que supusemos $G'(\overline{x}) > o$. De (12) decorre

$$w(N(Z)) \leq |g(\tilde{x})|.w(1/(G'(\overline{x}) + K_1[-w(Z), w(Z)])) \tag{13}$$

Como $\dfrac{1}{G'(\bar{x})+K_1[-w(Z),w(Z)]} = \left[\dfrac{1}{G'(x)+K_1w(Z)}, \dfrac{1}{G'(\bar{x})-K_1w(Z)}\right]$

$$w(1/(G'(x)+K_1[-w(Z),w(Z)])) = \frac{2K_1w(Z)}{(G'(x))^2 - K_1^2(w(Z))^2}$$

Substituindo em (13)

$$w(N(Z)) \leq \frac{|G'(\tilde{x})|.2K_1w(Z)}{(G'(\bar{x}))^2 - K_1^2(w(Z))^2} \qquad (14)$$

Substituindo (9) em (14)

$$w(N(Z)) \leq \frac{2|G'(Z)|w(Z).K_1.w(Z)}{(G'(\bar{x}))^2 - K_1^2(w(Z))^2} \qquad (15)$$

De (11), $|G'(Z)| \leq G'(\bar{x}) + K_1w(Z)$, tendo em vista que $G'(\bar{x}) + K_1w(Z)$ é positivo. Substituindo em (15) obtemos

$$w(N(Z)) \leq \left\{\frac{2K_1(G'(\bar{x}) + Kw(Z))}{(G'(\bar{x}))^2 - K_1^2(w(Z))^2}\right\}.(w(Z))^2, \qquad (16)$$

Como a parcela entre chaves vale

$$\frac{2K_1}{G'(\bar{x})} + \frac{2K_1^2.w(Z).\ 1 + K_1w(Z)/G'(\bar{x})}{(G'(\bar{x}))^2 - K_1^2(w(Z))^2} \qquad (17)$$

se w(Z) fôr <u>suficientemente pequeno</u> podemos majorar a segunda parcela do segundo membro de (17) por

$$\frac{2K_1}{G'(x)} + \frac{2K_1^2w(Z)}{(G'(\bar{x}))^2} < \frac{2K_1}{G'(\bar{x})} + 1 \qquad (18)$$

Substituindo (18) em (17) e (17) em (16), resulta

$$w(N(Z)) \leq \left(\frac{2K_1}{G'(\bar{x})} + 1\right).(w(Z))^2 \leq \underline{K}.(w(Z))^2 \qquad (19)$$

sendo $K \geq 2K_1/G'(\bar{x}) + 1$

Dados dois intervalos quaisquer S e R com $S \cap R \neq \emptyset$, temos $w(S \cap R) \leq \min(w(S),w(R))$. Escolhendo-se o intervalo ini -

cial X_o, suficientemente pequeno, de modo que a desigualdade (19) seja válida para o mesmo, teremos

$$w(N(X_o) \cap X_o) \leq K.(w(X_o))^2 \qquad (20)$$

Considerando a sequência X_{n+1} definida por $N(X_n) \cap X_n$ para a qual $X_{n+1} \subset X_n$, temos

$$w(X_{n+1}) = w(N(X_n) \cap X_n) \leq w(N(X_n)) \qquad (21)$$

Finalmente, de (19) e (21), deduzimos, por indução finita, que $w(X_{n+1}) \leq K.(w(X_n))^2$, q.e.d.

Programa em FORTRAN do Método de Newton Generalizado

Para ilustrar o método de Newton Generalizado apresentamos um programa redigido em FORTRAN que utiliza as subrotinas de ternos descritas nos apêndices A e B, para calcular um valor aproximado da raiz da equação $g(x) = x/(1+|x|)$, cuja única solução é òbviamente x=o. Temos $g'(x) = 1/(1+|x|)^2$ e adotamos o terno inicial $X_o = [-7, 4711, 247921]$.

Neste exemplo X_o foi escolhido de modo que a convergência não fosse rápida; o método de Newton para números reais é sempre divergente para $|x_o| \geq 1$. Observamos que para n=o(1)13 a convergência é linear, tornando-se em seguida quadrática, até n =17, quando diminui a velocidade de convergência por influência dos êrros de arredondamento. Para n=18 $\underline{x}$ torna-se zero por underflow. Para n= 19, $\underline{\tilde{x}}$ e $\bar{\bar{x}}$ correspondem ao valor exato.

Programa do Método de Newton Generalizado para ternos

```
      DIMENSION X(3),Y(3),FY(3),FLX(3),Z(3)
      WRITE(6,80)
      READ(5,50)X(1),X(2),X(3),M
      DO 40 I=1,M
      N=I-1
      WRITE(6,60)N,X(1),X(2),X(3)
      CALL DEFIN(Y,X(2),X(2),X(2))
      CALL F(Y,FY)
      CALL FL(X,FLX)
      CALL DIVD(FY,FLX)
      CALL SUBT(Y,FY)
      CALL INTIN(Y,X,Z,K)
      IF(K-1)40,10,40
   10 WRITE(6,70)I
      STOP
   40 CALL DEFIN(X,Z(1),Z(2),Z(3))
      STOP
   50 FORMAT(3F7.0,I2)
   60 FORMAT(/,1X,I3,E22.8,2E20.8)
   70 FORMAT(/,1X,27HINTERSECCAO NAO DEFINIDA NA,I2,15H-ESIMA ITERACAO)
   80 FORMAT(/,1X,3H  N,10X,4HXINF,16X,4HXINT,16X,4HXSUP,/)
      END

10/15/70     10:14 AM     ASR#3.2     69157   COMPILER
  0 MIN  9 SEC FOR COMPILATION PASS
   23 CARDS AT 151 CARDS PER MINUTE
   506 DIGITS DATA.  1004 DIGITS CODE.
```

```
SYMBOLIC LISTING FOR PROGAM :

LOW ADRS   HIGH ADRS  TEMPS     DATA BASE  CODE BASE
 002054     003556     002054    002072     002552

CALLED SUBPROGRAMS :
NAME     ADDRESS    SEG. NO.
DEFIN     026074      001
F         034270      001
FL        029116      001
DIVD      026318      001
SUBT      018348      001
INTIN     016546      001
READ.     011056      001
WRITE.    003556      001

COMMON BLOCKS REFERENCED :
NAME    ADDRESS
NONE
```

Subrotina para calcular a função $1/(1+|x|)^2$

```
      SUBROUTINE FL(X,FLX)
      DIMENSION X(3),FLX(3),Y(3),VAX(3)
      CALL DEFIN(Y,1.,1.,1.)
      CALL DEFIN(FLX,1.,1.,1.)
      CALL VABIN(X,VAX)
      CALL SOMA(Y,VAX)
      CALL PROD(Y,Y)
      IF(SINAL(Y))20,10,20
   10 WRITE(6,100)Y
      STOP
   20 CALL DIVD(FLX,Y)
      RETURN
  100 FORMAT(1X,2HY=,E15.8,20HDIVISAO NAO DEFINIDA)
      END

10/15/70    10:14 AM    ASR#3.2    69157  COMPILER
 0 MIN 10 SEC FOR COMPILATION PASS
  15 CARDS AT 084 CARDS PER MINUTE
  266 DIGITS DATA.   520 DIGITS CODE.
```

Nesta subrotina usamos aritmética de ternos arredondada.

O valor da função no terno X é guardado no terno FLX

Observamos que esta função é a derivada de $x/(1+|x|)$

Subrotina para calcular a função $x/(1+|x|)$

```
      SUBROUTINE F(X,FX)
      DIMENSION X(3),FX(3),Y(3),VAX(3)
      CALL DEFIN(Y,1.,1.,1.)
      CALL VABIN(X,VAX)
      CALL SOMA(Y,VAX)
      CALL DEFIN(FX,X(1),X(2),X(3))
      IF(SINAL(Y))20,10,20
   10 WRITE(6,100)Y
      STOP
   20 CALL DIVD(FX,Y)
      RETURN
  100 FORMAT(1X,2HY=,E15.8,20HDIVISAO NAO DEFINIDA)
      END

10/15/70    10:14 AM    ASR#3.2    69157  COMPILER
 0 MIN  8 SEC FOR COMPILATION PASS
  15 CARDS AT 111 CARDS PER MINUTE
  248 DIGITS DATA.   660 DIGITS CODE.
```

Nesta subrotina usamos aritmética de ternos arredondada

O valor da função no terno X é guardado no terno FX.

N	XINF	XINT	XSUP
0	-.70000000E+01	.47110000E+04	.24792100E+06
1	-.70000000E+01	.23515001E+04	.47100003E+04
2	-.70000000E+01	.11717503E+04	.23505006E+04
3	-.70000000E+01	.58187560E+03	.11707512E+04
4	-.70000000E+01	.28693866E+03	.58087732E+03
5	-.70000000E+01	.13947107E+03	.28594214E+03
6	-.70000000E+01	.65739095E+02	.13847819E+03
7	-.70000000E+01	.28877039E+02	.64754079E+02
8	-.70000000E+01	.10455255E+02	.27910510E+02
9	-.70000000E+01	.12712756E+01	.95425512E+01
10	-.70000000E+01	-.31442216E+01	.71155683E+00
11	-.23855215E+01	-.83698230E+00	.71155683E+00
12	-.38135336E+00	.16510173E+00	.71155683E+00
13	-.31855176E+00	-.14757794E+00	.23395890E-01
14	-.18978460E-01	.22087150E-02	.23395890E-01
15	-.10455890E-03	-.49845500E-04	.48679000E-05
16	-.24880000E-08	.39725000E-08	.10433000E-07
17	-.12000000E-14	-.40000000E-15	.40000000E-15
18	.00000000E-99	.65000000E-22	.13000000E-21
19	.00000000E-99	.00000000E-99	.00000000E-99

Capítulo 4

PROBLEMA DE VALORES INICIAIS EM EQUAÇÕES DIFERENCIAIS ORDINARIAS

§ 14 - Um método de Primeira Ordem

Definição

Um sistema de equações diferenciais ordinárias de primeira ordem é um sistema da forma

$$y_i' = f_i(x,y_1,y_2,\dots,y_m) \qquad (i=1,2,\dots,m) \tag{1}$$

onde as funções f_i são supostas definidas na região $x_o \leq x \leq a$ e $-\infty < y_i < \infty \qquad (i=1,2,\dots,m)$

A m-upla de funções $y_1(x)$, $y_2(x),\dots,y_m(x)$ definidas e diferenciáveis no intervalos $[x_o,a]$ que satisfaça identicamente em x a relação (1) é chamada solução do sistema.

O problema de achar uma solução do sistema que satisfaça a determinadas condições iniciais

$$y_i(x_o) = y_{io} \tag{2}$$

onde y_{io} são constantes, é denominado problema de valores iniciais.

Notação Vetorial

Considerando as seguintes representações:

$$y = (y_1,y_2,\dots,y_m)$$

$$y' = (y_1',y_2',\dots,y_m')$$

$$f = (f_1,f_2,\dots,f_m)$$

$$y_o = (y_{o1},y_{o2},\dots,y_{om})$$

podemos reescrever o problema (1)-(2) sob a forma vetorial

$$\begin{aligned} y' &= f(x,y) \\ y(x_o) &= y_o \end{aligned} \tag{3}$$

sendo $f : [x_o, a] \times R^m \longrightarrow R^m$

Supomos o espaço vetorial R^m dotado de uma normal $\| \quad \|$.

Equações Diferenciais Ordinárias de Ordem n

Uma equação diferencial ordinária de ordem n, qualquer que seja o inteiro $n \geq 2$, pode ser reduzida à forma (3) mediante introdução de variáveis auxiliares.

Consideremos o problema com valores iniciais

$$
\begin{aligned}
z^{(n)} &= g(x, z, z', z'', \ldots, z^{(n-1)}) \\
z(x_o) &= z_1 \\
z(x_1) &= z_2 \\
&\vdots \\
z^{(m-1)}(x_o) &= z_m
\end{aligned}
$$

Mediante as substituições $y_i = z^{(i-1)}$ $(i=1,2,\ldots,m)$ e convencionando-se $z^{(o)} = z$, obtemos o sistema:

$$
\begin{aligned}
y_1' &= y_2 \\
y_2' &= y_3 \\
&\vdots \\
y_n' &= g(x, y_1, y_2, \ldots, y_n)
\end{aligned}
$$

com $y_i(x_o) = z_i$ $(i=1,2,\ldots,n)$

Sistema autônomo

Um sistema de equações diferenciais ordinárias se diz autônomo se a função f em (3) não depender explìcitamente de x. Substituindo-se x por y_{m+1} em (3) transformamos o sistema (3) não autônomo de m equações no seguinte sistema autônomo de (m+1) equações:

$$
\begin{aligned}
y_i' &= f_i(y_{m+1}, y_1, \ldots, y_m) \qquad (i=1,2,\ldots,m) \\
y_{m+1}' &= 1
\end{aligned}
$$

satisfazendo a $y_i(x_o) = y_{io}$ $(i=1,2,\ldots,m)$ e $y_{m+1}(x_o) = x_o$.

Proposição 1

Sejam $B_j = [a_j, b_j]$, $a_j < b_j$ $(j=1,2,\ldots,m)$ $X' = [x_o, a]$, $x_o < a$, $B = [x_o, a] \times B_1 \times B_2 \times \ldots \times B_m$, $f(x,y)$ uma função definida e contínua em $D_f \supset B$, satisfazendo em D_f à condição de Lipchitz

$$||f(x,y)-f(x,\tilde{y})|| \leq K_f||y-\tilde{y}|| \tag{5}$$

Se y_{jo} cai no interior de B_j e se x^* é tal que

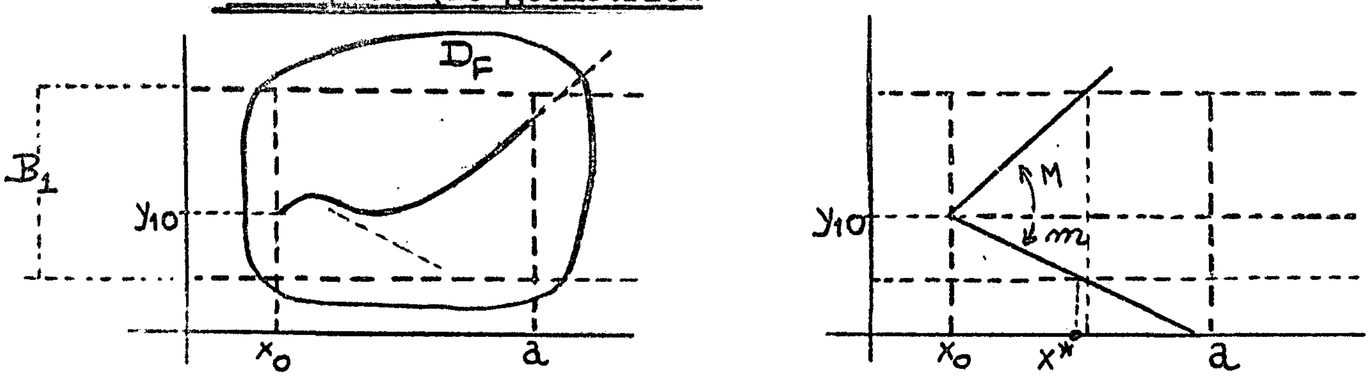
$$y_{jo} + (x^* - x_o)f_j(x,y) \in B_j \tag{6}$$

para todo $(x,y) \in B$, então existe uma única solução do problema (3 em

$$D = [x_o, x^*] \times B_1 \times B_2 \times \ldots \times B_m$$

Existe sempre x^*, $x_o < x^* < a$, satisfazendo à condição (6)

Interpretação geométrica



m e M são tais que $\bar{F}(B) = \{f(x,y) | (x,y) \in B\} \subset F(B) = [m, M]$

Método de Primeira Ordem para Intervalos

Introdução

Suponhamos que o segundo membro de (1) sejam funções de intervalos, definidas em

$$D_F = \mathcal{F}_{[x_o, a]} \times \mathcal{F}_{B_1} \times \ldots \times \mathcal{F}_{B_m}$$

satisfazendo às seguintes condições para $j=1,2,\ldots,m$:

(a) F_j é contínua e é uma extensão de f_j.

(b) F_j é monotônica.

(c) Existe um número real positivo K_F tal que

$w(F_j(X,Y_1,\ldots,Y_m)) \leq K_F w(X,Y)$

onde $X \subset [x_o,a]$, $Y_j \subset B_j$ e

$w(X,Y) = \max\{w(X),w(Y_1),\ldots,w(Y_m)\}$

Observação

(I)

Se $F = (F_1,F_2,\ldots,F_m)$ é uma função de intervalos racional em D_F e sua restrição real é a função $f = (f_1,f_2,\ldots,f_m)$ definida em B, então as condições (a),(b) e (c), acima descritas, estão satisfeitas pela F.

(II)

As condições (a),(b) e (c) implicam a condição (5).

De fato, seja $Y_j = [c_j,d_j]$ $(j=1,2,\ldots,m)$ e $Y = [c,d]$ com $c = (c_1,c_2,\ldots,c_m)$ e $d = (d_1,d_2,\ldots,d_m)$. Suponhamos que $w(F_j(X,Y)) \leq K_F \max\{w(X),w(Y)\}$ Então, $w(F_j(x,Y)) \leq K_F w(Y)$ para todo real $x \in [x_o,a]$. De (a) vemos que $f_j(x,y) \in F_j(x,Y)$, para todo y de Y. Consideremos os pontos c e d de Y. De (b) resulta $f_j(x,c) - f_j(x,d) = F_j(x,Y) - F_j(x,Y)$. Como $[z,w]-[z,w] = [-1,1]w([z,w])$, $|f_j(x,c)-f_j(x,d)| \leq w(F_j(x,Y)) \leq K_F \cdot \max\{w(x),w(Y)\}$ Sendo $\max\{w(x),w(Y)\} = w(Y) = \max\{w(Y_1),w(Y_2),\ldots,w(Y_m)\} = \max\{|d_1-c_1|,|d_2-c_2|,\ldots,|d_m-c_m|\} = ||c-d||$, concluímos que $|f_j(x,c)-f_j(x,d)| \leq K_F||c-d||$

Daí, $||f(x,c)-f(x,d)|| = \max_j |f_j(x,c)-f_j(x,d)| \leq K_F||c-d||$ (7)

(III)

Determinação de $X^* = [x_o, x^*]$.

Se $y_{jo} \in Y_{jo} \subset B_j$ (pròpiamente contido), então a equação

$$Y_{jo} + (X_j^* - x_o)F_j(B) = B_j$$

possui uma solução X_j^* com $w(X_j^*) > 0$ $(j=1,2,\ldots,m)$

Seja $X^* = [x_o, a] \cap X_1^* \cap X_2^* \cap \ldots \cap X_m^* = [x_o, x^*]$.

Por exemplo, se m=1, calculamos o intervalo X_1^* resolvendo a equação $Y_o + (X_1^* - x_o)F_1(B) = B_1$ e, em seguida, considerando $X^* = [x_o, a] \cap X_1^*$.

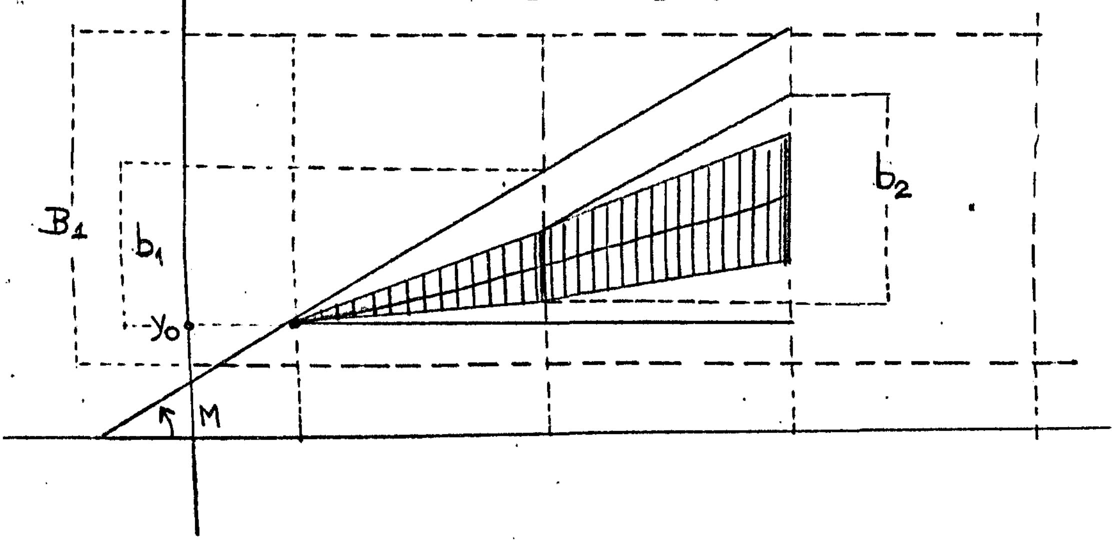
Descrição do Método de Primeira Ordem para Intervalos

Dado um inteiro positivo n, seja $h = w(X^*)/n$ e consideremos para cada i $(i=1,2,\ldots,n)$

$$\begin{aligned} X_i &= x_o + [i-1, i]h = X_{i-1} + h \\ b_i &= Y_{i-1} + [0,1]hF(B) = Y_{i-1} + S \\ Y_i &= Y_{i-1} + hF(X_i, b_i) \end{aligned} \qquad (8)$$

onde $b_i = (b_{1i}, b_{2i}, \ldots, b_{mi})$, $y_i = (y_{1i}, y_{2i}, \ldots, y_{mi})$ e $y_o = (y_{1o}, y_{2o}, \ldots, y_{mo})$.

A relação de recorrência (8) expressa a versão para intervalos do método de primeira ordem para equações diferenciais.



Observação

Seja y solução do problema (3), então $y(x) \in Y_i$ para todo $x \in X_i$. De fato, a solução y satisfaz à equação

$$y(x) = y(x_{i-1}) + \int_{x_{i-1}}^{x} f(t,y(t))dt, \qquad x \in X_i$$

Portanto,

$$(x-x_{i-1})\min_{X_i x b_i} f(x,y) \leq \int_{x_{i-1}}^{x} f(t,y(t))dt \leq (x-x_{i-1})\max_{X_i x b_i} f(x,y)$$

donde $(x-x_{i-1})[\min f(x,y), \max f(x,y)] \subset (x-x_{i-1})F(X_i, b_i)$

Portanto, $y(x) \in Y_{i-1} + (x-x_{i-1})F(X_i, b_i)$ (8')

ou $$y(x) \in Y_{i-1} + (x-x_{i-1})F(X_i, Y_{i-1}+[0,1]hF(B)) \subset Y_i$$

Lema 1

Seja z_n (n=o,1,2,...) uma sequência numérica tal que

$$|z_{n+1}| \leq A|z_n| + B$$

A e B constantes não negativas, independentes de n.

Então, $|z_n| \leq A^n|z_o| + \frac{A^n - 1}{A - 1}.B$

A demonstração se faz por indução completa.

Convergência

Vimos que $y(x) \in Y_i$ para todo $x \in X_i$. Desejamos mostrar que êsse intervalo converge para y(x) à medida que se aumenta o número de subdivisões do intervalo X^*. Para isto basta mostrarmos que $w(Y_i) \longrightarrow 0$ para $n \longrightarrow \infty$. Substituindo $x=x_i$ em (8') e calculando $w(Y_i)$, obtemos:

$$w(Y_i) = w(Y_{i-1}) + hw(F(X_i, Y_{i-1} + S))$$

$w(Y_i) \leq w(Y_{i-1}) + hK_F \max(h, w(Y_{i-1} + S))$.

Porém, $w(Y_{i-1} + S) = w(Y_{i-1}) + hc$, sendo $c = w(|\,0,1\,|.F(B))$.

Daí, $w(Y_i) \leq w(Y_{i-1}) + hK_F \max\,(h, w(Y_{i-1}) + ch)$

Como $\max\left\{(h, w(Y_{i-1}) + ch)\right\} = w(Y_{i-1}) + \max(c,1).h$, temos

$w(Y_i) \leq (1+hK_F)w(Y_{i-1}) + \max(c,1).h^2.K_F$

Tendo em vista que $w(Y_o)=w(y_o)=0$, aplicando o lema 1,obtemos

$w(Y_i) \leq \left\{\dfrac{(1+hK_F)^i - 1}{hK_F}\right\}\max(c,1).h^2.K_F = \left\{(1+hK_F)^i - 1\right\}\max(c,1)h$

Para h suficientemente pequeno, decorre da expansão em série de Taylor de e^{hK_F} que $1 + hK_F < e^{hK_F}$. Substituindo i por $(x_i - x_o)/h$ obtemos

$$w(Y_i) \leq \max(c,1)(e^{K_F(x_i - x_o)} - 1).h$$

Finalmente, tendo em vista que $h = w(X^*)/n$, vemos que existe uma constante K tal que

$$w(Y_i) \leq K/n$$

Portanto, $w(Y_i) \longrightarrow 0$ para $n \longrightarrow \infty$, q.e.d.

Ilustração do método

Seja $y' = y^2$, $y(0)=1$, $G(Y) = Y^2$ a extensão de y^2. A fim de conservarmos a notação usada definiremos $F(X,Y) = G(Y)$. Então $D_F = \mathcal{F}_{[0,a]} \times \mathcal{F}_B$, sendo f a restrição de F a $B = [0,a] \times B_1$.

Supomos que B_1 é um intervalo não degenerado que contém em seu interior o valor inicial $y(0) = 1$ e que $a > 0$. Sendo F uma função de intervalos racional, então as condições (a), (b) e (c) estão verificadas. Temos $F(B) = F([0,a], B_1) = B_1^2$. X_1^* é solução da equação $1 + X_1^*.B_1^2 = B_1$

Como $x_o = 0$, $0 \in X_1^*$ e para todo x de $X^* = [0,a] \cap X_1^*$

o problema tem solução única (Proposição 1)

Determinação de X^* (Ilustração geométrica)

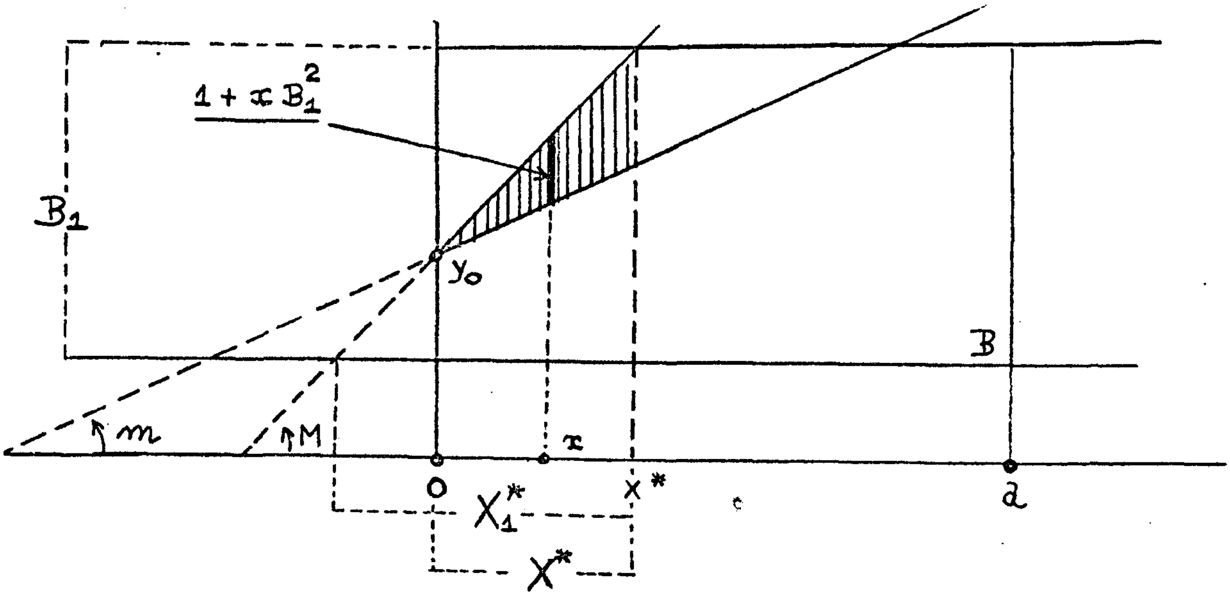
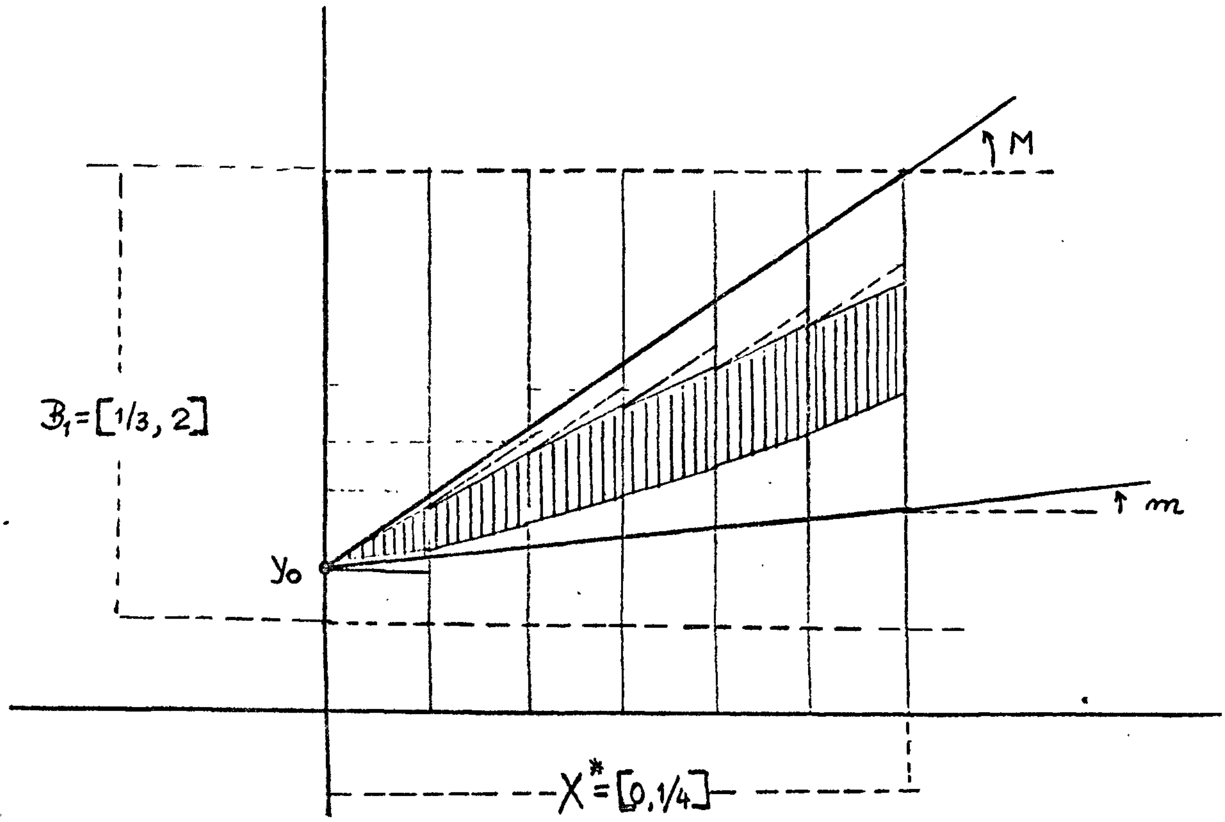


Ilustração geométrica da construção de $Y_i(x)$



Exemplo numérico

Seja $B_1 = [\frac{1}{3}, 2]$ e $a = 1$

Determinação de x^*

$1 + X_1^* [\frac{1}{9}, 4] = [\frac{1}{3}, 2]$, donde

$X_1^* [\frac{1}{9}, 4] = [\frac{1}{3}, 2] - 1 = [-\frac{2}{3}, 1]$

Seja $X_1^* = [c, d]$ Como $0 \in X_1^*$, temos $c < 0 < d$ e

$[c, d] . [\frac{1}{3}, 2] = [4c, 4d] = [-\frac{2}{3}, 1]$

onde $c = \frac{1}{4}\left(-\frac{2}{3}\right)$ e $d = 1/4$

Resulta $X_1^* = [-\frac{1}{6}, \frac{1}{4}]$ e $X^* = [-\frac{1}{6}, \frac{1}{4}] \circ [0, 1]$
$= [0, \frac{1}{4}]$

Programa em FORTRAN

Para ilustração apresentamos um programa redigido em FORTRAN, que usa a aritmética de ternos arredondada. As principais etapas do programa são:

-Leitura dos dados (N,X,Y,B1,XSTR)

N = 10, número de subdivisões do intervalo X^*

X = 0, valor inicial de x_i

Y = $[1,1,1]$, valor da condição inicial y(0)=1

B1= $[0,333 \quad 2,333 \quad 2,000]$

XSTR = X^* = $[0,000 \quad 0,125 \quad 0,250]$

-Calcula h

-Calcula S

-Imprime a solução (n x y(x))

-Incrementa X do passo h

-Calcula h.F(B)

-Calcula y = y + hF(y+S)

-Retorna à etapa de impressão, repetindo n vêzes esta rotina.

A função F(B) é calculado através de uma subrotina.

Método de Primeira Ordem para Intervalos ($m = 1$)

```
      DIMENSION X(3),Y(3),B(3),B1(3),XSTR(3),H(3),S(3),AB(3),FB(3),BB(3)
      WRITE(6,300)
      READ(5,100)N,X,Y,B1,XSTR
      P=(XSTR(3)-XSTR(1))/FLOAT(N-1)
      CALL DEFIN(H,P,P,P)
      Z=0.
      T=0.5
      U=1.
      CALL DEFIN(S,Z,T,U)
      CALL PROD(S,H)
      CALL F(B1,FB)
      CALL PROD(S,FB)
      DO 10 I=1,N
      I1=I-1
      CALL FOI(Y,BB)
      WRITE(6,200)I1,X(2),Y,I1,X(2),BB
      CALL SOMA(X,H)
      CALL DEFIN(AB,Y(1),Y(2),Y(3))
      CALL SOMA(AB,S)
      CALL DEFIN(B,AB(1),AB(2),AB(3))
      CALL F(B,FB)
      CALL PROD(FB,H)
   10 CALL SOMA(Y,FB)
      STOP
  100 FORMAT(I2,12F5.3)
  200 FORMAT(/,1X,I2,F5.3,3G14.8,I6,F5.3,3G14.8)
  300 FORMAT(1X,53X,14HFORMA CENTRADA,/)
      END

12/07/70      3214 PM     ASR#3.2      69157   COMPILER
  0 MIN  7 SEC FOR COMPILATION PASS
   29 CARDS AT 223 CARDS PER MINUTE
   614 DIGITS DATA.  1382 DIGITS CODE.

      SUBROUTINE F(B,FB)
      DIMENSION B(3),FB(3)
      CALL DEFIN(FB,B(1),B(2),B(3))
      CALL PROD(FB,B)
      RETURN
      END

12/07/70      3214 PM     ASR#3.2      69157   COMPILER
  0 MIN  3 SEC FOR COMPILATION PASS
   08 CARDS AT 123 CARDS PER MINUTE
    56 DIGITS DATA.   356 DIGITS CODE.
```

n	x	$\underline{x}$	$\tilde{x}$	$\bar{x}$
0	.000	1.0000000	1.0000000	1.0000000
1	.025	1.0250000	1.0267648	1.0302500
2	.050	1.0512656	1.0549321	1.0621867
3	.075	1.0788945	1.0846142	1.0959537
4	.100	1.1079948	1.1159354	1.1317114
5	.125	1.1386861	1.1490341	1.1696393
6	.150	1.1711012	1.1840644	1.2099389
7	.175	1.2053881	1.2211987	1.2528374
8	.200	1.2417121	1.2606303	1.2985917
9	.225	1.2802583	1.3025769	1.3474932
10	.250	1.3212348	1.3472842	1.3998742

CHMA CENTRADA

n	x	$(\bar{x}+\underline{x})/2$	$\tilde{x}$	$\hat{E}rro$
0	.000	1.0000000	1.0000000	.00000000E-99
1	.025	1.0276250	1.0267648	.26250000E-02
2	.050	1.0567261	1.0549321	.54605500E-02
3	.075	1.0874241	1.0846142	.85296000E-02
4	.100	1.1198531	1.1159354	.11858300E-01
5	.125	1.1541627	1.1490341	.15476600E-01
6	.150	1.1905200	1.1840644	.19418850E-01
7	.175	1.2291127	1.2211987	.23724650E-01
8	.200	1.2701519	1.2606303	.28439800E-01
9	.225	1.3138757	1.3025769	.33617450E-01
10	.250	1.3605545	1.3472842	.39319700E-01

A solução exata da equação diferencial dada é $y(x) = 1/(1-x)$

§ 15 - Geração automática dos coeficientes de Taylor

Introdução

Consideremos um sistema de equações diferenciais ordinárias autônomo, em que o segundo membro seja dado por expressões racio - nais. Métodos de ordem k, que estendem o método de 1a.ordem estudado no § 14, baseiam-se na expansão em série de Taylor.

Neste parágrafo descreveremos um procedimento simples e programável, de modo que o computador possa gerar, iterativamente, os coeficientes da expansão em série de Taylor de uma dada função racional.

A execução do programa com subrotinas da aritmética de ternos arredondada permitirá obter valores reais e intervalos dos coeficientes de Taylor de qualquer ordem, calculados num ponto dado.

Descrição do método

Seja $y_i' = f_i(y_1,y_2,\ldots,y_n)$ um sistema de equações diferen - ciais ordinárias autônomo, de ordem n, satisfazendo às condições inicias $y_i(a) = v_i$ $(i=1,2,\ldots,n)$

Se f_i é uma função racional, f_i é descrita por uma sequência finita de operações aritméticas executadas com as variáveis y_i $(i=1,2,\ldots,n)$. A ordem em que as operações aritméticas fo - rem executadas dá a regra ou o algoritmo para avaliação de f_i. A última das operações aritméticas da regra adotada define o valor de f_i e, portanto, de y_i'.

Seja $*$ uma das operações aritméticas +, -, . ou /. A cada resultado parcial da regra para avaliação de f_i associamos uma variável auxiliar y_k, sempre definida em função de uma ou duas variáveis do sistema $(y_r,\ r=1,2,\ldots,n)$ ou de variáveis auxiliares já introduzidas $(y_r,\ r=(n+1),(n+2),\ldots,(k-1))$.

Teremos então

$$y_r * y_s = \begin{cases} y_k & \text{se } * \text{ não fôr a última operação da regra.} \\ y_i' & \text{se } * \text{ fôr a última operação da regra} \end{cases}$$

No caso do cálculo da j-ésima derivada teremos, anàlogamente,

$$y_r^{(j)} * y_s^{(j)} = \begin{cases} y_k^{(j)} & \text{se } * \text{ indica operação intermediária.} \\ y_k^{(j+1)} & \text{se } * \text{ indica última operação} \end{cases}$$

Neste caso, para cada operação $*$, devemos aplicar a regra apropriada de diferenciação, que poderá ser a diferenciação de uma soma, uma subtração, um produto ou divisão.

Entretanto, tendo em vista nosso objetivo, é mais prático gerar diretamente os coeficientes de Taylor, programando o cálculo de $y_k^{(j)}/k!$

Sejam $P(x)$ e $Q(x)$ duas funções de x e $P_j = P^{(j)}/j!$

As regras para o cálculo dos coeficientes de Taylor da soma, diferença, produto e quociente entre duas funções, são as seguintes:

$$\begin{aligned} (P \pm Q)_j &= P_j \pm Q_j \\ (P.Q)_j &= \sum_{r=0}^{j} P_r Q_{j-r} \\ \left(\frac{P}{Q}\right)_j &= \frac{1}{Q}\left\{ P_j - \sum_{r=1}^{j} Q_r \left(\frac{P}{Q}\right)_{j-r} \right\} \end{aligned} \qquad (1)$$

<u>Exemplo</u>

Vamos exemplificar o método exposto sucintamente no parágrafo anterior, considerando o seguinte sistema autônomo:

$$\begin{aligned} y'_1 &= y_1 y_2 + y_3 y_4 \\ y'_2 &= y_2(y_3 - y_4) \\ y'_3 &= y_1/y_2 \\ y'_4 &= y_1 + y_2 \end{aligned} \qquad (2)$$

satisfazendo às condições iniciais $y_i(a) = 1$, sendo a o extremo inferior do intervalo $[a,x]$ onde se procura a solução.

Introdução de variáveis auxiliares

Adotaremos uma certa ordem na qual as operações aritméticas que definem f_i deverão ser executadas. Esta ordem dá a regra para avaliação de f_i. Ao resultado parcial de cada operação aritmética que figura na regra associamos uma nova variável. Estas variáveis são variáveis auxiliares que permitirão descrever o sistema de uma forma bem simples. A última operação da regra que define f_i é y'_i. As variáveis auxiliares deverão ser definidas sistemàticamente em função de uma ou duas das variáveis de que dependem f_i e das variáveis auxiliares já introduzidas. Teremos para o exemplo dado:

$$
\begin{aligned}
y_1 \cdot y_2 &= y_5 \\
y_3 \cdot y_4 &= y_6 \\
y_5 + y_6 &= y'_1 \\
y_3 - y_4 &= y_7 \\
y_2 \cdot y_7 &= y'_2 \\
y_1 / y_2 &= y'_3 \\
y_1 + y_2 &= y'_4
\end{aligned}
\qquad (3)
$$

Descrição do Sistema (2)

A partir de (3) construímos uma tabela que descreve o sistema. Cada linha dessa tabela corresponde a uma equação de (3). Na programação FORTRAN apresentada neste texto introduziremos as matrizes D e OP de modo que D(I,1),D(I,2),OP(I), e D(I,3) descrevam a I-ésima equação da tabela, isto é:

I	D(I,1)	D(I,2)	OP(I)	D(I,3)
1	1	2	*	5
2	3	4	*	6
3	5	6	+	1
4	3	4	-	7
5	2	7	*	2
6	1	2	/	3
7	1	2	+	4

onde * indica multiplicação e $I=1,2,\ldots,N_1 = 7$

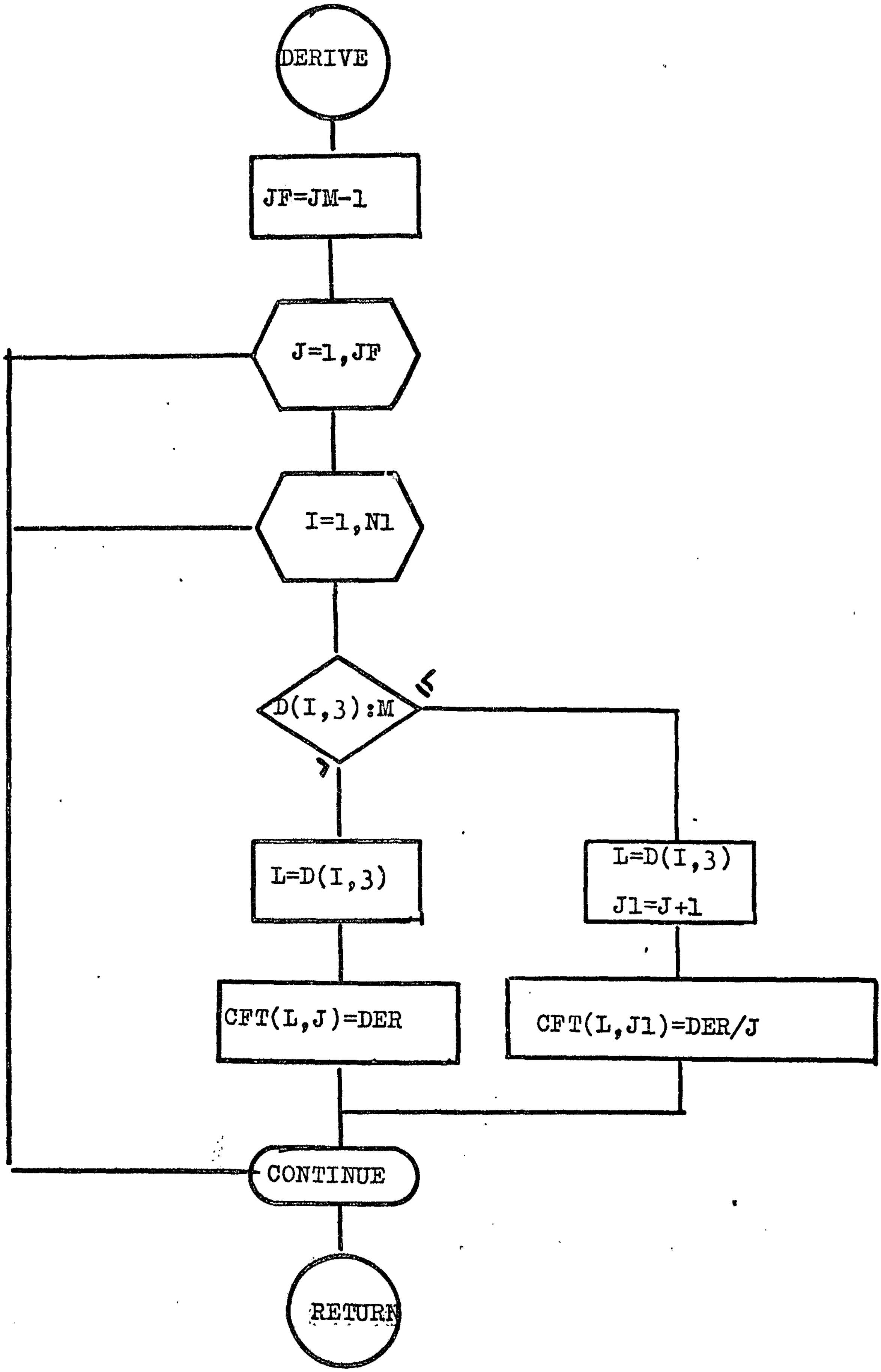
Geração dos coeficientes de Taylor

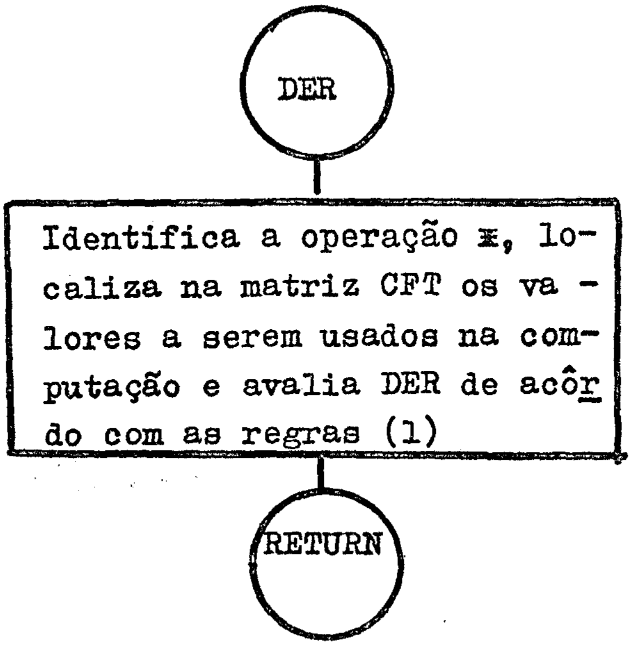
Os coeficientes de Taylor de y_i $(i=1,2,\ldots,7)$ serão colocados numa tabela onde a i-ésima linha corresponde à variável y_i e a j-ésima coluna ao coeficiente de Taylor de ordem j $(j=0,1,\ldots,JM)$ Esta tabela corresponderá à matriz CFT usadas nos programas em FORTRAN que serão apresentados neste texto. (DER,DERIVE,Programa Principal).

O programa principal lê as matrizes D e OP, que descrevem o sistema, e os valores iniciais CFT(L,0) (L=1,2,3,4) da matriz CFT.

Os coeficientes de Taylor de y_i serão calculados por meio de uma FUNCTION (DER) e de uma SUBROUTINE (DERIVE). A função irá calcular o coeficiente de Taylor de uma soma, subtração, produto ou divisão, de acôrdo com a regra (1), enquanto que a subrotina preenche a tabela CFT, corrigindo o valor fornecido pela DER quando necessário.

Apresentamos nas páginas seguintes os diagramas de blocos de um programa principal que apenas manda imprimir os coeficientes de Taylor calculados, o diagrama de blocos da subrotina DERIVE e um diagrama de blocos resumido da função DER.


DERIVE
JF=JM-1
J=1,JF
I=1,N1
D(I,3):M
L=D(I,3)
CFT(L,J)=DER
L=D(I,3)
J1=J+1
CFT(L,J1)=DER/J
CONTINUE
RETURN



Para controlar e evitar a ocorrência de "overflow" ou "underflow", podemos introduzir no cálculo dos coeficientes de Taylor um fator de normalização, chamado DELTA, o qual deverá ser reconsiderado num programa principal que utilize os valores dos coeficientes de Taylor. Então, em vez de $y_i^{(j)} / j!$ deveremos calcular $\mathrm{DELTA}^j\, y_i^{(j)}/j!$ como o valor de cada elemento (L,J) da matriz CFT. Na subrotina DERIVE quando D(I,3) > M o programa identifica a variável da I-ésima linha da matriz que descreve o sistema como sendo uma variável auxiliar. Caso contrário a operação OP(I) indica a última operação da sequência de operações que define f_i e, portanto, y_i' ou $f_i^{(j)} = y_i^{(j+1)}$. Neste caso será guardado em CFT(L,j+1) o valor DER*DELTA/(j) uma vez que $\mathrm{DELTA}^{(j-1)}/(j-1)!$ já se encontra incorporado nos valores que definiram DER.

<u>Caso em que constantes aparecem na expressão racional de f_i</u>

Adotaremos a seguinte convenção: as constantes serão sempre o primeiro fator da operação. Então, acrescentamos mais uma coluna na matriz que descreve o sistema e guardamos em D(I,4) o valor da constante. O programa da FUNCTION DER antes de iniciar o processo do cálculo dos coeficientes de Taylor verifica se D(I,1) é zero.

Se fôr diferente de zero não haverá constantes a considerar e desvia o processamento do programa para a rotina que permite o cálculo de acôrdo com as regras (1). Caso contrário, desvia para outra rotina, mais simples, abaixo descrita, onde o valor da constante poderá ser usado (caso da multiplicação e divisão).

Se P é uma constante, usaremos as seguintes regras:

$$(P \pm Q) = Q_j$$

$$(PQ)_j = PQ_j \qquad\qquad (1')$$

$$(P/Q)_j = \frac{1}{Q}\left\{ P_j - \sum_{i=1}^{j} Q_i . (P/Q)_{j-i} \right\}$$

Programas em FORTRAN

Apresentaremos a seguir listagem dos programas redigidos em FORTRAN e o resultado do processamento de dois exemplos. O exemplo 1 cuja descrição do sistema não utiliza constantes é o exemplo usado para exemplificar o método. O exemplo 2 refere-se ao seguinte sistema:

$$y_1' = 2(y_1y_2y_3y_4)$$

$$y_2' = y_2(2-y_3)$$

$$y_3' = 2/y_1$$

$$y_4' = 2y_1$$

satisfazendo às condições iniciais $y_i(a) = 1$ $(i=1,2,3,4)$. A descrição deste sistema leva ao seguinte resultado:

1	2	x	5	0
3	4	x	6	0
5	6	x	7	0
0	7	x	1	2
0	3	-	8	2
2	8	x	2	0
0	1	/	3	2
0	1	+	4	2

Teste da FUNCTION DER e SUBROUTINE DERIVE

```
      DIMENSION CF(10),D(10,10)
      ALFHA CF
      COMMON ISE,JSE,JM,N1,M,CP,D,CFT(10,10),DELTA
    1 READ(5,10,END=90)N1,DELTA
      DO 20 I=1,N1
      READ(5,21)D(I,1),D(I,2),CP(I),D(I,3),D(I,4)
   20 WRITE(6,22)D(I,1),D(I,2),CP(I),D(I,3),D(I,4)
      READ(5,23)N,M,JM
      READ(5,24)(CFT(L,1),L=1,M)
      CALL DERIVE
      MN=M+N
      DO 30 L=1,MN
   30 WRITE(6,26)(CFT(L,J),J=1,JM)
      GO TO 1
   90 STOP
   10 FORMAT(I2,F8.1)
   21 FORMAT(2F5.2,A4,2F5.2)
   22 FORMAT(/,1X,2F5.0,A4,F5.0,F5.2)
   23 FORMAT(3I2)
   24 FORMAT(10F6.2)
   26 FORMAT(/,1X,10G13.6)
      END

12/08/70      8≥45 AM     ASR#3.2      69157   COMPILER
  0 MIN 17 SEC FOR COMPILATION PASS
   23 CARDS AT 077 CARDS PER MINUTE
   258 DIGITS DATA.  2284 DIGITS CODE.  2592 DIGITS COMMON.
```

```
      SUBROUTINE DERIVE
      DIMENSION SE(10),D(10,10)
      ALFHA SE
      COMMON J,I,JM,N1,M,SE,D,CFT(10,10),DELTA
      JF=JM-1
      DO 30 J=1,JF
      DO 30 I=1,N1
      IF(IFIX(D(I,3))-M)20,20,10
   10 L=D(I,3)
      CFT(L,J)=DER(J,I)
      GO TO 30
   20 L=D(I,3)
      J1=J+1
      CFT(L,J1)=DER(J,I)*DELTA/FLOAT(J)
   30 CONTINUE
      RETURN
      END

12/08/70      8≥45 AM     ASR#3.2      69157   COMPILER
  0 MIN 28 SEC FOR COMPILATION PASS
   18 CARDS AT 038 CARDS PER MINUTE
   68 DIGITS DATA.  1122 DIGITS CODE.  2592 DIGITS COMMON.
```

```
      FUNCTION DER(JSE,ISE)
      DIMENSION OP(10),ALFA(4),D(10,10)
      ALPHA OP,ALFA
      COMMON J,I,JMSE,N1SE,MSE,OP,D,CFT(10,10),DELTSE
      DIMENSION T(10)
      DATA ALFA/6H   +  ,6H   -  ,6H   *  ,6H   /  /
      K1=D(I,1)
      K2=D(I,2)
      DO 10 I1=1,4
      IF(ALFA(I1)-OP(I))10,20,10
   10 CONTINUE
      WRITE(6,11)OP(I)
   11 FORMAT(/,1X,8HOPERACAO,A4,15HNAO RECONHECIDA)
      STOP
   20 IF(K1)30,50,30
   30 GO TO(40,41,42,43),I1
   40 SINAL=1.
      GO TO 44
   41 SINAL=-1.
   44 DER=CFT(K1,J)+SINAL*CFT(K2,J)
      RETURN
   42 DER=0.
      DO 46 JV=1,J
   46 DER=DER+CFT(K1,JV)*CFT(K2,J+1-JV)
      RETURN
   43 T(1)=CFT(K1,1)/CFT(K2,1)
      DO 49 JV=2,J
      T(JV)=0.
      DO 48 IV=2,JV
   48 T(JV)=T(JV)+CFT(K2,IV)*T(JV+1-IV)
   49 T(JV)=(CFT(K1,JV)-T(JV))/CFT(K2,1)
      DER=T(J)
      RETURN
C
   50 GO TO(61,62,63,64),I1
   61 SINAL=1.
      GO TO 65
   62 SINAL=-1.
   65 DER=CFT(K2,J)*SINAL
      RETURN
   63 DER=D(I,4)*CFT(K2,J)
      RETURN
   64 T(1)=D(I,4)/CFT(K2,1)
      DO 80 JV=2,J
      T(JV)=0.
      DO 70 IV=2,JV
   70 T(JV)=T(JV)+CFT(K2,IV)*T(JV+1-IV)
   80 T(JV)=(-T(JV))/D(I,4)
      DER=T(J)
      RETURN
      END

12/08/70     8>44 AM     ASR#3.2     69157  COMPILER
 0 MIN 48 SEC FOR COMPILATION PASS
  55 CARDS AT 067 CARDS PER MINUTE
  416 DIGITS DATA.  4404 DIGITS CODE.  2592 DIGITS COMMON.
```

1.	2.	*	5.	.00
3.	4.	*	6.	.00
5.	6.	+	1.	.00
3.	4.	-	7.	.00
2.	7.	*	2.	.00
1.	2.	/	3.	.00
1.	2.	+	4.	.00

Exemplo 1

1.00000	2.00000	2.50000	2.00000	1.41667
1.00000	.000000E-99	-.500000	.000000E-99	.208333
1.00000	1.00000	1.00000	1.00000	.750000
1.00000	2.00000	1.00000	.666667	.500000
1.00000	2.00000	2.00000	1.00000	.000000
1.00000	3.00000	4.00000	4.66667	.000000
.000000E-99	-1.00000	.000000E-99	.333333	.000000

1.	2.	*	5.	.00
3.	4.	*	6.	.00
5.	6.	*	7.	.00
0.	7.	*	1.	2.00
0.	3.	-	8.	2.00
2.	8.	*	2.	.00
0.	1.	/	3.	2.00
0.	1.	+	4.	2.00

Exemplo 2

1.00000	2.00000	4.00000	4.33333	4.33333
1.00000	-1.00000	-.500000	1.16667	-.125000
1.00000	2.00000	-1.00000	-.666667	.416667
1.00000	1.00000	1.00000	1.33333	1.08333
1.00000	1.00000	1.50000	.500000	.000000
1.00000	3.00000	2.00000	1.66667	.000000
1.00000	4.00000	6.50000	8.66667	.000000
-1.00000	-2.00000	1.00000	.666667	.000000

Usando-se aritmética de ternos faremos as seguintes adaptações aos programas apresentados: no lugar de D(I,4) introduzimos a matriz CD(I,3), pois esta constante poderá ser expressa por um terno. Por motivo análogo substituimos a FUNCTION DER pela SUBROUTINE DER, uma vêz que os valores calculados pela DER são ternos; êstes valores são guardados na matriz DR(3), um dos argumentos da subrotina DER. Nos apêndices A e B descrevemos sucintamente os parâmetros das subrotinas e no apêndice C anexamos o resultado do processamento dos exercícios 1 e 2 já vistos e do exercício 3 que veremos no parágrafo seguinte, resolvidos com a aritmética de ternos. Estes exercícios no apêndice C receberam os números 12, 13 e 14.

Observação final

O método que acabamos de estudar é aplicável não apenas a sistemas diferenciais $y_i' = f_i(y_1,y_2,\dots,y_m)$ em que f_i são funções racionais, mas também à s funções não racionais que satisfazem a equações diferenciais racionais. Esta classe de funções é mais ampla do que possa parecer à primeira vista.

Seja fog a função composta de f e g. O valor de fog num ponto x é f(g(x)). Pela "regra da cadeia" para a diferenciação de uma função composta obtemos:

$$f(g(x))' = f'(f(x)).g'(x)$$

Se f e g satisfazem equações diferenciais racionais, mostraremos que a composta fog também satisfaz.

Consideremos as funções $u_1,u_2,\dots u_m,v_1,v_2,\dots,v_n$ que satisfazem aos sistemas diferenciais:

$$u_j' = U_j(u_1,u_2,\dots,u_m) \qquad (j=1,2,\dots,m)$$

$$v_i' = V_i(v_1,v_2,\dots,v_n) \qquad (i=1,2,\dots,n)$$

Seja $w_j = u_j o v_1$. Então $(u_j o v_1)(x) = u_j(v_1(x))$. Portanto,

$$w_j' = u_j'(v_1).v_1' \qquad \text{donde}$$

$$w'_j = U_j(w_1,w_2,\dots,w_m).V_1(v_1,v_2,\dots,v_n) \qquad (j=1,2,\dots,m)$$

$$v'_i = V_i(v_1,v_2,\dots,v_n) \qquad (i=1,2,\dots,n)$$

que é um sistema de equações diferenciais racional de m+n equações nas incógnitas w_j $(j=1,2,\dots,m)$ e v_i $(i=1,2,\dots,n)$.

Exemplo 3

Seja $y_1' = \cos x/\operatorname{sen} x$, $\quad y_1(\pi/2) = 0$

Transformação num sistema diferencial racional

As funções senx e cosx não são racionais, mas satisfazem a equações diferenciais racionais. Seja $y_2 = \cos x$ e $y_3 = \operatorname{sen} x$. Então

$$\begin{aligned} y_1' &= y_2/y_3 & \qquad y_1(\pi/2) &= 0 \\ y_2' &= -y_3 & y_2(\pi/2) &= 0 \\ y_3' &= y_2 & y_3(\pi/2) &= 1 \end{aligned}$$

Descrição do sistema racional

$$\begin{aligned} y_2/y_3 &= y_4 \\ y_4 &= y_1' \\ -y_3 &= y_2' \\ y_2 &= y_3' \end{aligned}$$

Resultado do processamento do programa que gera os coeficientes de Taylor

2.	3.	/	4.	.00
0.	4.	*	1.	1.00
0.	3.	*	2.	-1.00
0.	2.	*	3.	1.00

	0	1	2	3	4
1	.000000E-99	.000000E-99	-.500000	.000000E-99	-.833333E-01
2	.000000E-99	-1.00000	.000000E-99	.166667	.000000E-99
3	1.00000	.000000E-99	-.500000	.000000E-99	.416667E-01
4	.000000E-99	-1.00000	.000000E-99	-.333333	.000000E-99

	5	6	7	8	9
1	.000000E-99	-.222222E-01	.000000E-99	-.674603E-02	.000000E-99
2	-.833333E-02	.000000E-99	.198413E-03	.000000E-99	-.275573E-05
3	.000000E-99	-.138889E-02	.000000E-99	.248016E-04	.000000E-99
4	-.133333	.000000E-99	-.539683E-01	.000000E-99	.000000

CFT(I, J) $1 \leq I \leq 4$ $0 \leq J \leq 9$

APÊNDICE A

Representação de um terno $X = [x,\tilde{x},\bar{x}]$

$$x = (x_1,x_2,x_3)$$

Representação de vetores de ternos

$$y = \begin{bmatrix} (y_{11},y_{21},y_{31}) \\ (y_{12},y_{22},y_{32}) \\ \vdots \\ (y_{1n},y_{2n},y_{3n}) \end{bmatrix}$$

Representação de matrizes de ternos

$$z = \begin{bmatrix} (z_{111},z_{211},z_{311}) & (z_{112},z_{212},z_{312}) & \cdots & (z_{11n},z_{21n},z_{31n}) \\ (z_{121},z_{221},z_{321}) & (z_{122},z_{222},z_{322}) & \cdots & (z_{12n},z_{22n},z_{32n}) \\ \vdots & & & \\ (z_{1n1},z_{2n1},z_{3n1}) & (z_{1n2},z_{2n2},z_{3n2}) & \cdots & (z_{1nn},z_{2nn},z_{3nn}) \end{bmatrix}$$

Reserva de área para programação FORTRAN

```
DIMENSION X(3),Y(3,10),Z(3,10,10)
```

onde X é um terno, Y um vetor de ternos e Z uma matriz de ternos.

Subrotinas para operações com ternos

CALL ARDTO(A)

O argumento A é uma variável.

Soma uma unidade no dígito de ordem mais baixa do número em A.

CALL ARDT2(A,AD)

Os argumentos A e AD são ternos, o primeiro em precisão simples e o segundo em precisão dupla. A e AD guardam o resultado de uma mesma operação aritmética com ternos executada em precisão simples e precisão dupla, respectivamente,em que cada elemento do terno é calculado usando-se a aritmética real com truncamento. Após a execução desta subrotina o terno A é arredondado de acôrdo com a regra de arredondamento para intervalos. O elemento A(2) permanece inalterado.

As operações aritméticas com ternos referidas nas subrotinas descritas a seguir usam as definições do § 5 do Cap.1 e a técnica de arredondamento de intervalo. As subrotinas para as operações aritméticas com ternos utilizam para o arredondamento do intervalo a subrotina ARDT2 e esta última, para o arredondamento dos extremos do intervalo, chama a subrotina ARDTO.

CALL SOMA(A,B)	A e B são ternos Calcula A = A+B
CALL SUBT(A,B)	A e B são ternos Calcula A = A-B
CALL PROD(A,B)	A e B são ternos Calcula A = A.B
CALL DIVD(A,B)	A e B são ternos Calcula A = A/B, se B $\neq$ o
CALL TESIN(A)	A é um terno Se A(1) $\leq$ A(3), continua. Caso contrário imprime A com a mensagem "Intervalo não definido" e pára.
CALL DEFIN(A,B,C,D)	A é um terno. B,C e D são variáveis. Define o terno: A(1)=B, A(2)=C e A(3) = D.
CALL FCI(A,B)	A e B são ternos. Define em B a forma centrada do intervalo [A(1),A(3)], isto é: B(1) = (A(1)+A(3))/2 B(3) = (A(3)-A(1))/2 B(2) = A(2)

SINAL(A)

O subprograma é FUNCTION SINAL(A)

A é um terno

$$\mathrm{SINAL(A)} = \begin{cases} 1 & \text{se} \quad o < A \\ o & \text{se} \quad o \in A \\ -1 & \text{se} \quad o > A \end{cases}$$

CALL VABIN(A,V)

A e V são ternos

Define em V o valor absoluto de A, isto é:

$$V = \begin{cases} A & \text{se SINAL(A)=1} \\ -A & \text{se SINAL(A)=-1} \\ [o,\mathrm{ABS(A(2))},\max(-A(1),A(3))] & \\ & \text{se SINAL(A)=o} \end{cases}$$

CALL INTIN(A,B,Z,N)

A, B e Z são ternos. N é uma variável.

Define em Z a intersecção de A com B.

$$N = \begin{cases} o & \text{se} \quad A \cap B \quad \text{está definida.} \\ 1 & \text{se} \quad A \cap B \quad \text{não está definida.} \end{cases}$$

$Z = [\max(X(1),Y(1)), Z(2), \min(X(3),Y(3))]$

onde $Z(2) = (Z(1)+Z(2))/2$

Observação: Sejam A e B duas matrizes de ternos. Para definir, por exemplo, a soma do (I,J)-ésimo elemento de A com (I,J)-ésimo elemento de B, basta usar

```
CALL SOMA(A(1,I,J),B(1,I,J))
```

em vêz de:

```
Z(1)=A(1,I,J)
Z(2)=A(2,I,J)
Z(3)=A(3,I,J)
W(1)=B(1,I,J)
W(2)=B(2,I,J)
W(3)=B(3,I,J)
CALL SOMA(Z,W)
```

Análogamente, se desejarmos definir o (I,J)-ésimo elemento de uma matriz de ternos A como sendo o terno [X,Y,Z], é suficiente usar:

```
CALL DEFIN(A(1,I,J),X,Y,Z)
```

Relação das subrotinas usadas no método de Hansen para Inversão de Matriz

CALL ARDTO(A)	A é uma variável Soma uma unidade no dígito de ordem mais baixa do número em A.
CALL ARDT2(A,AD)	A e AD são intervalos A em precisão simples. AD em precisão dupla. A e AD guardam o resultado de uma mesma operação aritmética com intervalos executada em precisão simples e precisão dupla, respectivamente, em que cada elemento do intervalo é calculado usando-se a aritmética real com truncamento. Após a execução o intervalo é arredondado de acôrdo com a regra de arredondamento para intervalos.
CALL SOMA(A,B)	A e B são intervalos Calcula A = A + B
CALL SUBT(A,B)	A e B são intervalos Calcula A = A-B
CALL PROD(A,B)	A e B são intervalos Calcula A = A.B
CALL DIVD(A,B)	A e B são intervalos Calcula A = A/B, se B $\not\ni$ o
CALL TESIN(A)	A é um intervalo Se A(1) $\leq$ A(2), continua. Caso contrário imprime A com a mensagem "Intervalo não definido" e pára.

CALL DEFIN(A,B,C)	A é um intervalo. B e C são variáveis. Define o intervalo A = [B,C]
CALL FCI(A,B)	A e B são intervalos. Define em B a forma centrada do intervalo A, isto é: $B = \left[\frac{A(1)+A(3)}{2}, \frac{A(3)-A(1)}{2}\right]$

As mesmas subrotinas em precisão dupla são as seguintes:

CALL ARDT1D(A)	A é uma variável em precisão dupla.
CALL ARDTOD(A)	A é um intervalo em precisão dupla.
CALL SOMAD(A,B)	A e B são intervalos em p.d.
CALL SUBTD(A,B)	A e B são intervalos em p.d.
CALL PRODT(A,B)	A e B são intervalos em p.d.
CALL DIVDD(A,B)	A e B são intervalos em p.d.
CALL TESIND(A)	A é um intervalo em p.d.
CALL DEFIND(A,B,C)	A é um intervalo e B e C são variáveis, todos em p.d.
CALL FCID(A,B)	A e B são intervalos em p.d.

Subrotinas para resolução de sistema linear e inversão de matriz para números reais

CALL DECMA(N,A,KP)	N é uma variável. A e KP matrizes a duas e uma dimensão, respectivamente. N = ordem do sistema A = matriz dos coeficientes do sistema KP= matriz de permutação. Entrada: N e A. Saída: KP e A Executa a permutação das equações do sistema, guardando em KP a ordem das permutas. A matriz A é decomposta, segundo o método de eliminação de Gauss, em duas matrizes triangulares, superior e inferior. No processo de decomposição é aplicada a técnica de pivotação após equilibrar a matriz A.

CALL RSTIS(N,A,KP)

N = ordem da matriz KP
KP = matriz a uma dimensão.
A = matriz a duas dimensões de ordem Nx(N+1)
Os elementos de KP e de A_{NxN} são obtidos através da subrotina DECMA.

RSTIS resolve os sistemas associados às matrizes triangulares de A_{NxN} e segundo membro definidos na coluna (N+1)-ésima de A, êstes fornecidos pelo programa principal, guardando a solução nessa mesma coluna.

CALL INVM(N,A,B)

N = ordem das matrizes quadradas A e B, ambas a duas dimensões.
A = matriz dada
B = matriz inversa de A

Usa as subrotinas DECMA e RSTIS

Subrotinas específicas para o método de Hansen

CALL SM1V(N,S)

N = ordem da matriz S
S = matriz a três dimensões correspondente a uma matriz quadrada de intervalos de ordem N
Calcula S = I + S

CALL SM2V(N,S,P)

N = ordem das matrizes S e P
S e P são matrizes a três dimensões correspondentes à matrizes quadradas de intervalos de ordem N.

Calcula S = S + P

CALL PRV(N,S,E)

N = ordem das matrizes S e E
S e E são matrizes a três dimensões correspondentes a matrizes quadradas de intervalos de ordem N.

Calcula S = E.S

CALL MPV(RO,N,P)	N = ordem da matriz P RO= variável real. P = matriz a três dimensões, correspondente a uma matriz quadrada de intervalos de ordem N. Define a matriz P em que todos elementos são iguais a [-RO,RO]
CALL MSV(N,M,E,S)	N = ordem das matrizes E e S M = variável inteira. E e S são matrizes a três dimensões, correspondentes a matrizes quadradas de intervalos de ordem N. Calcula $S = \sum_{i=1}^{M} E^i$ usando a forma de ninho (((I+E)E+I)E+...+I)E
CALL SM2VD(N,S,T)	N = ordem das matrizes S e T S e T são matrizes a três dimensões, correspondentes a matrizes quadradas de intervalos de ordem N, a primeira em precisão dupla e a segunda em precisão simples. Calcula S = S + T
CALL PRVD(N,S,E)	N = ordem das matrizes S e E S e E são matrizes a três dimensões, correspondentes a matrizes quadradas de intervalos de ordem N. Ambas em precisão dupla. Calcula S = E.S

Subrotinas para resolução de sistema linear pelo método de aliminação de Gauss com emprêgo da aritmética de ternos com arredondamento

Estas subrotinas correspondem às subrotinas DECMA e RSTIS descritas entre as subrotinas usadas no método de Hansen. Os programas têm a mesma lógica e a única modificação nos argumentos de chamada é que a matriz A é uma matriz a três dimensões, correspondente a uma matriz de ternos. As subrotinas são:

```
CALL DECMAV(N,A,KP)
CALL RSTISV(N,A,KP)
```

Subrotina para inversão de matriz por um método direto baseado no processo de eliminação de Gauss, usando a aritmética de terno arredondada

Esta subrotina corresponde à subrotina INVM descrita entre as subrotinas utilizadas pelo método de Hansen. A lógica é a mesma, substituindo-se as matrizes A e B a duas dimensões por matrizes a três dimensões, correspondentes a matrizes de ternos a duas dimensões. São chamadas as subrotinas DECMAV e RSTISV.

```
CALL INVMV(N,A,B)
```

Subrotinas para geração automática dos coeficientes de Taylor no problema de condições iniciais de equações diferenciais ordinárias racionais (sistema autônomo), usando-se a aritmética de terno arredondada

As subrotinas são DER e DERIVE cujos argumentos estão no COMMON que descreveremos em seguida. O uso correto destas subrotinas exige a leitura cuidadosa do texto, uma vez que as matrizes D e OP pressupoẽ uma descrição correta do sistema de equações diferenciais racional.

Area do COMMON das subrotinas DER e DERIVE e Programa Principal

P.Principal	DER	DERIVE	Significado
JSE	J	J	Coluna de CFT
ISE	I	I	Linha de CFT
JM	JMSE	JM	Maior valor de J
N1	N1SE	N1	Maior valor de I
M	MSE	M	Ordem do sistema
OP	OP	OPSE	Descrição (operação)
D	D	D	Descrição
CD	CD	CDSE	Descrição (constantes)
CFT	CFT	CFT	Tabela dos Coef.Taylor
DELTA	DELTSE	DELTA	Fator de Normalização
DRSE	DR	DR	Coef.Taylor de Ordem (J-1) da I-ésima variável descrita em D, OP,CD

O sufixo SE no final de cada nome que aparece no COMMON indica que a variável ou matriz é sem efeito para o respectivo programa. A matriz OP é alfabética e a matriz D, inteira.

APÊNDICE B

```
      SUBROUTINE SOMA(A,B)
      DOUBLE PRECISION C(3),D(3)
      DIMENSION A(3),B(3)
      C(1)=A(1)
      C(2)=A(2)
      C(3)=A(3)
      D(1)=B(1)
      D(2)=B(2)
      D(3)=B(3)
      C(1)=C(1)+D(1)
      C(3)=C(3)+D(3)
      A(1)=C(1)
      A(3)=C(3)
      CALL ARDT2(A,C)
      A(2)=C(2)+D(2)
      RETURN
      END

11/30/70     11≥34 AM     ASR#3.2      69157   COMPILER
  0 MIN   6 SEC FOR COMPILATION PASS
   23 CARDS AT 225 CARDS PER MINUTE
   224 DIGITS DATA.    732 DIGITS CODE.
```

```
      SUBROUTINE SUBT(A,B)
      DOUBLE PRECISION C(3),D(3)
      DIMENSION A(3),B(3)
      C(1)=A(1)
      C(2)=A(2)
      C(3)=A(3)
      D(1)=B(1)
      D(2)=B(2)
      D(3)=B(3)
      C(1)=C(1)-D(3)
      C(3)=C(3)-D(1)
      A(1)=C(1)
      A(3)=C(3)
      CALL ARDT2(A,C)
      A(2)=C(2)-D(2)
      RETURN
      END

11/30/70     11≥34 AM     ASR#3.2      69157   COMPILER
  0 MIN   4 SEC FOR COMPILATION PASS
   18 CARDS AT 219 CARDS PER MINUTE
   224 DIGITS DATA.    732 DIGITS CODE.
```

```
      SUBROUTINE PROD(A,B)
      DOUBLE PRECISION C(3),D(3),Y(4)
      DIMENSION A(3),B(3),X(4)
      C(1)=A(1)
      C(2)=A(2)
      C(3)=A(3)
      D(1)=B(1)
      D(2)=B(2)
      D(3)=B(3)
      X(1)=A(1)*B(1)
      X(2)=A(1)*B(3)
      X(3)=A(3)*B(1)
      X(4)=A(3)*B(3)
      Y(1)=C(1)*D(1)
      Y(2)=C(1)*D(3)
      Y(3)=C(3)*D(1)
      Y(4)=C(3)*D(3)
      I1=1
      I2=1
      AMI=X(1)
      AMA=X(1)
      DO 70 I=2,4
      IF(AMI-X(I))40,40,50
   50 I1=I
      AMI=X(I)
   40 IF(AMA-X(I))60,70,70
   60 I2=I
      AMA=X(I)
   70 CONTINUE
      A(1)=X(I1)
      A(3)=X(I2)
      C(1)=Y(I1)
      C(3)=Y(I2)
      CALL ARDT2(A,C)
      A(2)=C(2)*D(2)
      RETURN
      END

11/30/70     11≥34 AM     ASR#3.2      69157  COMPILER
  0 MIN  7 SEC FOR COMPILATION PASS
   38 CARDS AT 324 CARDS PER MINUTE
   466 DIGITS DATA,  1962 DIGITS CODE.
```

```
      SUBROUTINE DIVD(A,B)
      DOUBLE PRECISION C(3),D(3),Y(4)
      DIMENSION A(3),B(3),X(4)
      P=B(1)*B(3)
      IF(P)10,10,20
   10 WRITE(6,15)B
   15 FORMAT(1X,20HDIVISAO NAO DEFINIDA,3E20.8)
      STOP
   20 C(1)=A(1)
      C(2)=A(2)
      C(3)=A(3)
      D(1)=B(1)
      D(2)=B(2)
      D(3)=B(3)
      X(1)=A(1)/B(1)
      X(2)=A(1)/B(3)
      X(3)=A(3)/B(1)
      X(4)=A(3)/B(3)
      Y(1)=C(1)/D(1)
      Y(2)=C(1)/D(3)
      Y(3)=C(3)/D(1)
      Y(4)=C(3)/D(3)
      I1=1
      I2=1
      AMI=X(1)
      AMA=X(1)
      DO 80 I=2,4
      IF(AMI-X(I))60,60,55
   55 AMI=X(I)
      I1=I
   60 IF(AMA-X(I))70,80,80
   70 AMA=X(I)
      I2=I
   80 CONTINUE
      A(1)=X(I1)
      A(3)=X(I2)
      C(1)=Y(I1)
      C(3)=Y(I2)
      CALL ARDT2(A,C)
      A(2)=C(2)/D(2)
      RETURN
      END

11/30/70     11:34 AM     ASR#3.2     69157   COMPILER
  0 MIN   7 SEC FOR COMPILATION PASS
   43 CARDS AT 344 CARDS PER MINUTE
   584 DIGITS DATA.  2332 DIGITS CODE.
```

SIZE REAL = 12

```
      SUBROUTINE ARDTO(A)
      IF(A)10,90,20
   10 SINAL = -1.
      GO TO 30
   20 SINAL = +1.
   30 B=ABS(A)
      NC=0
      IF(B-1.)40,70,60
   40 B=B*10.
      IF(B-1.)50,80,80
   50 NC=NC-1
      GO TO 40
   60 B=B/10.
      NC=NC+1
      IF(B-1.)80,60,60
   70 NC=1
   80 AR=10.**(NC-12)
      A=A+SINAL*AR
   90 RETURN
      END
```

```
11/30/70     11≥35 AM     ASR#3.2      69157   COMPILER
  0 MIN  5 SEC FOR COMPILATION PASS
   21 CARDS AT 250 CARDS PER MINUTE
   146 DIGITS DATA,    590 DIGITS CODE.
```

```
      SUBROUTINE ARDT2(A,AD)
      DOUBLE PRECISION AD,R
      DIMENSION A(3),AD(3)
      IF(A(1))10,25,25
   10 R=AD(1)-A(1)
      IF (R)20,25,20
   20 CALL ARDTO(A(1))
   25 IF(A(3))50,50,30
   30 R=AD(3)-A(3)
      IF(R)40,50,40
   40 CALL ARDTO(A(3))
   50 RETURN
      END
```

```
11/30/70     11≥34 AM     ASR#3.2      69157   COMPILER
  0 MIN  4 SEC FOR COMPILATION PASS
   14 CARDS AT 184 CARDS PER MINUTE
   132 DIGITS DATA,    722 DIGITS CODE.
```

```
      SUBROUTINE TESIN(A)
      DIMENSION A(3)
      IF(A(1)-A(3))10,10,20
   10 RETURN
   20 WRITE(6,30)A
   30 FORMAT(1X,22HINTERVALO NAO DEFINIDO,2E15.8)
      STOP
      END

11/30/70     11≥34 AM     ASR#3.2      69157   COMPILER
  0 MIN  4 SEC FOR COMPILATION PASS
   09 CARDS AT 136 CARDS PER MINUTE
   146 DIGITS DATA.   270 DIGITS CODE.
```

```
      SUBROUTINE FCI(A,B)
      DOUBLE PRECISION C(2)
      DIMENSION A(3),B(3)
      B(2)=A(2)
      C(1)=A(1)
      C(2)=A(3)
      B(1)=(C(1)+C(2))/2.
      B(3)=(C(2)-C(1))/2.
      IF(A(3)-A(1)-B(3)*2.)10,20,10
   10 CALL ARDTC(B(3))
   20 RETURN
      END

11/30/70     11≥34 AM     ASR#3.2      69157   COMPILER
  0 MIN  4 SEC FOR COMPILATION PASS
   13 CARDS AT 173 CARDS PER MINUTE
   176 DIGITS DATA.   762 DIGITS CODE.
```

```
      SUBROUTINE DEFIN(A,B,C,D)
      DIMENSION A(3)
      A(1)=B
      A(2)=C
      A(3)=D
      RETURN
      END

11/30/70     11≥35 AM     ASR#3.2      69157   COMPILER
  0 MIN  3 SEC FOR COMPILATION PASS
   08 CARDS AT 126 CARDS PER MINUTE
    56 DIGITS DATA.   206 DIGITS CODE.
```

```
      FUNCTION SINAL(A)
      DIMENSION A(3)
      IF(A(1))10,40,50
   10 IF(A(3))20,40,40
   20 SINAL=-1.
      RETURN
   40 SINAL=0.
      RETURN
   50 SINAL=1.
      RETURN
      END

11/30/70     11≥35 AM     ASR#3.2      69157   COMPILER
  0 MIN  4 SEC FOR COMPILATION PASS
   12 CARDS AT 179 CARDS PER MINUTE
    66 DIGITS DATA,    312 DIGITS CODE.


      SUBROUTINE VABIN(A,V)
      DIMENSION A(3),V(3),Z(3)
      CALL DEFIN(V,A(1),A(2),A(3))
      CALL DEFIN(Z,-1.,-1.,-1.)
      IF(SINAL(Y))10,20,50
   10 CALL PROD(V,Z)
      RETURN
   20 V(1)=0.
      V(2)=ABS(A(2))
      XI=-A(1)
      IF(XI-A(3))30,30,40
   30 V(3)=A(3)
      RETURN
   40 V(3)=XI
   50 RETURN
      END

11/30/70     11≥35 AM     ASR#3.2      69157   COMPILER
  0 MIN  4 SEC FOR COMPILATION PASS
   17 CARDS AT 207 CARDS PER MINUTE
   206 DIGITS DATA,  1030 DIGITS CODE.


      SUBROUTINE INTIN(A,B,Z,N)
      DIMENSION A(3),B(3),Z(3)
      CALL DEFIN(Z,A(1),A(2),A(3))
      CALL SUBT(Z,B)
      IF(SINAL(Z))10,20,10
   10 N=1
      RETURN
   20 IF(A(1)-B(1))30,30,40
   30 Z(1)=B(1)
      GO TO 50
   40 Z(1)=A(1)
   50 IF(A(3)-B(3))60,60,70
   60 Z(3)=A(3)
      GO TO 80
   70 Z(3)=B(3)
   80 N=0
      Z(2)=Z(1)/2.+Z(3)/2.
      RETURN
      END
```

```
SEGMENT INVM,PRV,PRVC
SIZE REAL = 6
      SUBROUTINE SOMA(A,B)
      DOUBLE PRECISION C(2),D(2)
      DIMENSION A(2),B(2)
      C(1)=A(1)
      C(2)=A(2)
      D(1)=B(1)
      D(2)=B(2)
      C(1)=C(1)+D(1)
      C(2)=C(2)+D(2)
       A(1)=C(1)
      A(2)=C(2)
      CALL ARDT2(A,C)
      RETURN
      END

11/30/70     11≥29 AM     ASR#3.2      69157   COMPILER
  0 MIN  6 SEC FOR COMPILATION PASS
   20 CARDS AT 175 CARDS PER MINUTE
   102 DIGITS DATA.    528 DIGITS CODE.
```

```
      SUBROUTINE SUBT(A,B)
      DOUBLE PRECISION C(2),D(2)
      DIMENSION A(2),B(2)
      C(1)=A(1)
      C(2)=A(2)
      D(1)=B(1)
      D(2)=B(2)
      C(1)=C(1)-D(2)
      C(2)=C(2)-D(1)
       A(1)=C(1)
      A(2)=C(2)
      CALL ARDT2(A,C)
      RETURN
      END

11/30/70     11≥29 AM     ASR#3.2      69157   COMPILER
  0 MIN  5 SEC FOR COMPILATION PASS
   15 CARDS AT 172 CARDS PER MINUTE
   102 DIGITS DATA.    528 DIGITS CODE.
```

```
      SUBROUTINE PROD(A,B)
      DOUBLE PRECISION C(2),D(2),Y(4)
      DIMENSION X(4),A(2),B(2)
      C(1)=A(1)
      C(2)=A(2)
      D(1)=B(1)
      D(2)=B(2)
      X(1)=A(1)*B(1)
      X(2)=A(1)*B(2)
      X(3)=A(2)*B(1)
      X(4)=A(2)*B(2)
      Y(1)=C(1)*D(1)
      Y(2)=C(1)*D(2)
      Y(3)=C(2)*D(1)
      Y(4)=C(2)*D(2)
      I1=1
      I2=1
      AMI=X(1)
      AMA=X(1)
      DO 70 I=2,4
      IF(AMI-X(I))40,40,50
   50 I1=I
      AMI=X(I)
   40 IF(AMA-X(I))60,70,70
   60 I2=I
      AMA=X(I)
   70 CONTINUE
      A(1)=X(I1)
      A(2)=X(I2)
      C(1)=Y(I1)
      C(2)=Y(I2)
      CALL ARDT2(A,C)
      RETURN
      END

11/30/70     11:29 AM     ASR#3.2     69157   COMPILER
  0 MIN  8 SEC FOR COMPILATION PASS
   35 CARDS AT 259 CARDS PER MINUTE
   254 DIGITS DATA.  1758 DIGITS CODE.
```

```
      SUBROUTINE DIVD(A,B)
      DOUBLE PRECISION C(2),D(2),Y(4)
      DIMENSION A(2),B(2),X(4)
      P=B(1)*B(2)
      IF(P)10,10,20
   10 WRITE(6,15)B
   15 FORMAT(1X,20HDIVISAO NAO DEFINIDA,2E16.8)
      STOP
   20 C(1)=A(1)
      C(2)=A(2)
      D(1)=B(1)
      D(2)=B(2)
      X(1)=A(1)/B(1)
      X(2)=A(1)/B(2)
      X(3)=A(2)/B(1)
      X(4)=A(2)/B(2)
      Y(1)=C(1)/D(1)
      Y(2)=C(1)/D(2)
      Y(3)=C(2)/D(1)
      Y(4)=C(2)/D(2)
      I1=1
      I2=1
      AMI=X(1)
      AMA=X(1)
      DO 80 I=2,4
      IF(AMI-X(I))60,60,55
   55 AMI=X(I)
      I1=I
   60 IF(AMA-X(I))70,80,80
   70 AMA=X(I)
      I2=I
   80 CONTINUE
      A(1)=X(I1)
      A(2)=X(I2)
      C(1)=Y(I1)
      C(2)=Y(I2)
      CALL ARCT2(A,C)
      RETURN
      END

11/30/70     11:29 AM     ASR#3.2     69157  COMPILER
  0 MIN  9 SEC FOR COMPILATION PASS
   40 CARDS AT 253 CARDS PER MINUTE
   334 DIGITS DATA,  2110 DIGITS CODE.
```

```
      SUBROUTINE FCI(A,B)
      DOUBLE PRECISION C(2)
      DIMENSION A(2),B(2)
      C(1)=A(1)
      C(2)=A(2)
      B(1)=(C(1)+C(2))/2.
      B(2)=(C(2)-C(1))/2.
      IF(A(2)-A(1)-B(2)*2.)10,20,10
   10 CALL ARDTC(B(2))
   20 RETURN
      END
```

```
11/30/70     11≥29 AM     ASR#3.2      69157   COMPILER
  0 MIN   5 SEC FOR COMPILATION PASS
   12 CARDS AT 129 CARDS PER MINUTE
   110 DIGITS DATA.    660 DIGITS CODE.
```

```
      SUBROUTINE DEFIN(A,B,C)
      DIMENSION A(2)
      A(1)=B
      A(2)=C
      RETURN
      END
```

```
11/30/70     11≥30 AM     ASR#3.2      69157   COMPILER
  0 MIN   5 SEC FOR COMPILATION PASS
   07 CARDS AT 076 CARDS PER MINUTE
    38 DIGITS DATA.    146 DIGITS CODE.
```

```
      SUBROUTINE TESIN(A)
      DIMENSION A(2)
      IF(A(1)-A(2))10,10,20
   10 RETURN
   20 WRITE(6,30)A
   30 FORMAT(1X,22HINTERVALO NAO DEFINIDO,2E15.8)
      STOP
      END
```

```
11/30/70     11≥30 AM     ASR#3.2      69157   COMPILER
  0 MIN   4 SEC FOR COMPILATION PASS
   09 CARDS AT 134 CARDS PER MINUTE
   122 DIGITS DATA.    270 DIGITS CODE.
```

```
      SUBROUTINE ARDT1D(A)
      DOUBLE PRECISION A,SINAL,B,AR
      IF(A)10,90,20
   10 SINAL = -1.
      GO TO 30
   20 SINAL = +1.
   30 B=DABS(A)
      NC=0
      IF(B-1.)40,70,60
   40 B=B*10.
      IF(B-1.)50,80,80
   50 NC=NC-1
      GO TO 40
   60 B=B/10.
      NC=NC+1
      IF(B-1.)80,60,60
   70 NC=1
   80 AR=10.**(NC-12)
      A=A+SINAL*AR
   90 RETURN
      END

11/30/70     11≥30 AM     ASR#3.2      69157   COMPILER
  0 MIN  8 SEC FOR COMPILATION PASS
   22 CARDS AT 160 CARDS PER MINUTE
   146 DIGITS DATA.   608 DIGITS CODE.
```

```
      SUBROUTINE ARDTDD(A)
      DOUBLE PRECISION A(2)
      IF(A(1))10,15,15
   10 CALL ARDT1D(A(1))
   15 IF(A(2))25,25,20
   20 CALL ARDT1D(A(2))
   25 RETURN
      END

11/30/70     11≥30 AM     ASR#3.2      69157   COMPILER
  0 MIN  4 SEC FOR COMPILATION PASS
   09 CARDS AT 115 CARDS PER MINUTE
    58 DIGITS DATA.   382 DIGITS CODE.
```

```
      SUBROUTINE SCMAD(A,B)
      DOUBLE PRECISION A(2),B(2),C(2)
      C(1)=A(1)
      C(2)=A(2)
      A(1)=A(1)+B(1)
      A(2)=A(2)+B(2)
      CALL ARDTOD(A)
      IF(A(1))5,35,35
    5 IF(C(1))10,40,30
   10 IF(B(1))20,40,40
   20 CALL ARDT1D(A(1))
      GO TO 40
   30 IF(B(1))40,40,20
   35 IF(A(2))70,70,40
   40 IF(C(2))50,70,80
   50 IF(B(2))60,70,70
   60 CALL ARDT1D(A(2))
   70 RETURN
   80 IF(B(2))70,70,60
      END

11/30/70     11:30 AM     ASR#3.2      69157   COMPILER
  0 MIN  6 SEC FOR COMPILATION PASS
   21 CARDS AT 190 CARDS PER MINUTE
   102 DIGITS DATA.  1324 DIGITS CODE.
```

```
      SUBROUTINE SUBTD(A,B)
      DOUBLE PRECISION A(2),B(2),C(2)
      C(1)=A(1)
      C(2)=A(2)
      A(1)=A(1)-B(2)
      A(2)=A(2)-B(1)
      CALL ARDTOD(A)
      IF(A(1))5,35,35
    5 IF(C(1))10,40,30
   10 IF(B(2))40,40,20
   20 CALL ARDT1D(A(1))
      GO TO 40
   30 IF(B(2))20,40,40
   35 IF(A(2))70,70,40
   40 IF(C(2))50,70,80
   50 IF(B(1))70,70,60
   60 CALL ARDT1D(A(2))
   70 RETURN
   80 IF(B(1))60,70,70
      END

11/30/70     11:30 AM     ASR#3.2      69157   COMPILER
  0 MIN  6 SEC FOR COMPILATION PASS
   21 CARDS AT 207 CARDS PER MINUTE
   102 DIGITS DATA.  1324 DIGITS CODE.
```

```
      SUBROUTINE PRODD(A,B)
      DOUBLE PRECISION A,B,X,AMI,AMA
      DIMENSION A(2),B(2),X(4)
      X(1)=A(1)*B(1)
      X(2)=A(1)*B(2)
      X(3)=A(2)*B(1)
      X(4)=A(2)*B(2)
      AMI=X(1)
      AMA=X(1)
      DO 50 I=2,4
      IF(AMI-X(I))30,30,20
   20 AMI=X(I)
   30 IF(AMA-X(I))40,50,50
   40 AMA=X(I)
   50 CONTINUE
      A(1)=AMI
      A(2)=AMA
      CALL ARDTOD(A)
      RETURN
      END

11/30/70    11≥30 AM    ASR#3.2     69157  COMPILER
  0 MIN  5 SEC FOR COMPILATION PASS
   21 CARDS AT 233 CARDS PER MINUTE
   166 DIGITS DATA.  1066 DIGITS CODE.
```

```
      SUBROUTINE DIVDD(A,B)
      DOUBLE PRECISION A,B,X,AMI,AMA
      DIMENSION A(2),B(2),X(4)
      P=B(1)*B(2)
      IF(P)10,10,20
   10 WRITE(6,15)B
   15 FORMAT(1X,20HDIVISAO NAO DEFINIDA,2E25.16)
      STOP
   20 X(1)=A(1)/B(1)
      X(2)=A(1)/B(2)
      X(3)=A(2)/B(1)
      X(4)=A(2)/B(2)
      AMI=X(1)
      AMA=X(1)
      DO 60 I=2,4
      IF(AMI-X(I))40,40,30
   30 AMI=X(I)
   40 IF(AMA-X(I))50,60,60
   50 AMA=X(I)
   60 CONTINUE
      A(1)=AMI
      A(2)=AMA
      CALL ARDTOD(A)
      RETURN
      END

11/30/70    11≥31 AM    ASR#3.2     69157  COMPILER
  0 MIN  8 SEC FOR COMPILATION PASS
   26 CARDS AT 180 CARDS PER MINUTE
   252 DIGITS DATA.  1418 DIGITS CODE.
```

```
      SUBROUTINE DEFIND(A,B,C)
      DOUBLE PRECISION A,B,C
      DIMENSION A(2)
      A(1)=B
      A(2)=C
      RETURN
      END

11/30/70     11:31 AM     ASR#3.2      69157   COMPILER
  0 MIN   4 SEC FOR COMPILATION PASS
   08 CARDS AT 105 CARDS PER MINUTE
    38 DIGITS DATA,    146 DIGITS CODE.
```

```
      SUBROUTINE TESIND(A)
      DOUBLE PRECISION A(2)
      IF(A(1)-A(2))10,10,20
   10 RETURN
   20 WRITE(6,30)A
   30 FORMAT(1X,22HINTERVALO NAO DEFINIDO,2D25.16)
      STOP
      END

11/30/70     11:31 AM     ASR#3.2      69157   COMPILER
  0 MIN   4 SEC FOR COMPILATION PASS
   09 CARDS AT 118 CARDS PER MINUTE
   140 DIGITS DATA,    270 DIGITS CODE.
```

```
      SUBROUTINE FOID(A,B)
      DOUBLE PRECISION A(2),B(2),C(2)
      B(1)=(A(1)+A(2))/2.
      B(2)=(A(2)-A(1))/2.
      IF(A(2)-A(1)-B(2)*2.)10,20,10
   10 C(1)=DABS(B(1))
      C(2)=DABS(B(2))
      CALL ARDTOD(C)
      B(2)=B(2)/DABS(B(2))*C(2)
   20 RETURN
      END

11/30/70     11:31 AM     ASR#3.2      69157   COMPILER
  0 MIN   5 SEC FOR COMPILATION PASS
   12 CARDS AT 123 CARDS PER MINUTE
   122 DIGITS DATA,   1086 DIGITS CODE.
```

```
      SUBROUTINE DECMA(N,A,KP)
      DIMENSION FESC(10),A(10,1),KP(10)
      DO 50 I=1,N
      KP(I)=I
      FESC(I)=ABS(A(I,1))
      DO 40 J=2,N
      IF(FESC(I)-ABS(A(I,J)))30,40,40
   30 FESC(I)=ABS(A(I,J))
   40 CONTINUE
      FFF=FESC(I)
   50 FESC(I)=1./FFF
C
      N1=N-1
      DO 120 K=1,N1
C
      IMAX=K
      KMAX=KP(K)
      AMAX=FESC(KMAX)*ABS(A(KMAX,K))
      K1=K+1
      DO 70 K2=K1,N
      KK2=KP(K2)
      AM=FESC(KK2)*ABS(A(KK2,K))
      IF(AMAX-AM)60,70,70
   60 AMAX=AM
      KMAX=KK2
      IMAX=K2
   70 CONTINUE
      IF(A(KMAX,K))100,90,100
   90 WRITE(6,91)
   91 FORMAT(1X,17HDETERMINANTE NULO)
      STOP
  100 JOTA=KP(K)
      KP(K)=KMAX
      KP(IMAX)=JOTA
C
      DO 120 K2=K1,N
      KK2=KP(K2)
      A(KK2,K)=A(KK2,K)/A(KMAX,K)
      DO 120 J=K1,N
  120 A(KK2,J)=A(KK2,J)-A(KK2,K)*A(KMAX,J)
      KK2=KP(N)
      IF(A(KK2,N))125,90,125
  125 RETURN
      END

11/30/70     11:31 AM     ASR#3.2      69157   COMPILER
  0 MIN  8 SEC FOR COMPILATION PASS
   45 CARDS AT 328 CARDS PER MINUTE
   302 DIGITS DATA,  4072 DIGITS CODE.
```

```
      SUBROUTINE RSTIS(N,A,KP)
      DIMENSION A(10,1),KP(10)
      M=N+1
      DO 124 I=2,N
      S=0.
      I1=I-1
      LI=KP(I)
      DO 122 J=1,I1
      LJ=KP(J)
  122 S=S+A(LI,J)*A(LJ,M)
  124 A(LI,M)=A(LI,M)-S
C
      LN=KP(N)
      A(LN,M)=A(LN,M)/A(LN,N)
      N1=N-1
      DO 128 K=1,N1
      I=N-K
      LI=KP(I)
      I1=I+1
      S=0.
      DO 126 J=I1,N
      LJ=KP(J)
  126 S=S+A(LI,J)*A(LJ,M)
  128 A(LI,M)=(A(LI,M)-S)/A(LI,I)
C
      RETURN
      END

11/30/70     11:31 AM     ASR#3.2      69157   COMPILER
  0 MIN   7 SEC FOR COMPILATION PASS
   28 CARDS AT 235 CARDS PER MINUTE
   130 DIGITS DATA.  2812 DIGITS CODE.
```

```
      SUBROUTINE INVM(N,A,B)
      DIMENSION A(10,1),B(10,1),KP(10)
      CALL DECMA(N,A,KP)
      M=N+1
      DO 40 I=1,N
      DO 30 J=1,N
      IF(I-J)20,10,20
   10 A(J,M)=1.
      GO TO 30
   20 A(J,M)=0.
   30 CONTINUE
      CALL RSTIS(N,A,KP)
      DO 40 J=1,N
      LJ=KP(J)
   40 B(J,I)=A(LJ,M)
      RETURN
      END

11/30/70     11:31 AM     ASR#3.2      69157   COMPILER
  0 MIN   5 SEC FOR COMPILATION PASS
   18 CARDS AT 185 CARDS PER MINUTE
   130 DIGITS DATA.  1110 DIGITS CODE.
```

```
      SUBROUTINE SM1V(N,S)
      DIMENSION S(2,10,1),Z1(2),Z2(2)
      DO 30 I=1,N
      DO 30 J=1,N
      IF(I-J)30,20,30
   20 Z=1.
      CALL DEFIN(Z1,Z,Z)
      CALL DEFIN(Z2,S(1,I,J),S(2,I,J))
      CALL SOMA(72,Z1)
      CALL DEFIN(S(1,I,J),Z2(1),Z2(2))
   30 CONTINUE
      RETURN
      END

11/30/70     11≥31 AM     ASR#3.2      69157  COMPILER
  0 MIN  5 SEC FOR COMPILATION PASS
   14 CARDS AT 152 CARDS PER MINUTE
   112 DIGITS DATA.  1086 DIGITS CODE.


      SUBROUTINE SM2V(N,S,P)
      DIMENSION S(2,10,1),P(2,10,1),Z1(2),Z2(2)
      DO 10 I=1,N
      DO 10 J=1,N
      CALL DEFIN(Z1,S(1,I,J),S(2,I,J))
      CALL DEFIN(Z2,P(1,I,J),P(2,I,J))
      CALL SOMA(Z1,Z2)
   10 CALL DEFIN(S(1,I,J),Z1(1),Z1(2))
      RETURN
      END

11/30/70     11≥31 AM     ASR#3.2      69157  COMPILER
  0 MIN  6 SEC FOR COMPILATION PASS
   11 CARDS AT 101 CARDS PER MINUTE
   102 DIGITS DATA.  1504 DIGITS CODE.


      SUBROUTINE PRV(N,S,E)
      DIMENSION E(2,10,1),S(2,10,1),ES(2,10,10),Z1(2),Z2(2),Z3(2)
      DO 10 I=1,N
      DO 10 K=1,N
      Z=0.
      CALL DEFIN(Z3,Z,Z)
      DO 10 J=1,N
      CALL DEFIN(Z1,E(1,I,J),E(2,I,J))
      CALL DEFIN(Z2,S(1,J,K),S(2,J,K))
      CALL PROD(Z1,Z2)
      CALL SOMA(Z3,Z1)
   10 CALL DEFIN(ES(1,I,K),Z3(1),Z3(2))
      DO 20 I=1,N
      DO 20 J=1,N
   20 CALL DEFIN(S(1,I,J),ES(1,I,J),ES(2,I,J))
      RETURN
      END

11/30/70     11≥32 AM     ASR#3.2      69157  COMPILER
  0 MIN  6 SEC FOR COMPILATION PASS
   18 CARDS AT 160 CARDS PER MINUTE
  2144 DIGITS DATA.  2492 DIGITS CODE.
```

```
      SUBROUTINE MPV(RO,N,P)
      DIMENSION P(2,10,1)
      DO 10 I=1,N
      DO 10 J=1,N
      Z=-RO
   10 CALL DEFIN(P(1,I,J),Z,RO)
      RETURN
      END
```

```
11/30/70     11:32 AM     ASR#3.2       69157   COMPILER
  0 MIN   4 SEC FOR COMPILATION PASS
   09 CARDS AT 120 CARDS PER MINUTE
    62 DIGITS DATA.    492 DIGITS CODE.
```

```
      SUBROUTINE MSV(N,M,E,S)
      DIMENSION E(2,10,1),S(2,10,1)
      DO 10 I=1,N
      DO 10 J=1,N
   10 CALL DEFIN(S(1,I,J),E(1,I,J),E(2,I,J))
      DO 20 K=1,M
      IF(K-M)15,30,30
   15 CALL SM1V(N,S)
      CALL PRV(N,S,E)
   20 CONTINUE
   30 RETURN
      END
```

```
11/30/70     11:32 AM     ASR#3.2       69157   COMPILER
  0 MIN   5 SEC FOR COMPILATION PASS
   13 CARDS AT 151 CARDS PER MINUTE
    70 DIGITS DATA.  1172 DIGITS CODE.
```

```
      SUBROUTINE SM2VD(N,S,T)
      DOUBLE PRECISION S,Z1,Z2
      DIMENSION S(2,10,1),T(2,10,1),Z1(2),Z2(2)
      DO 10 I=1,N
      DO 10 J=1,N
      CALL DEFIND(Z1,S(1,I,J),S(2,I,J))
      Z2(1)=T(1,I,J)
      Z2(2)=T(2,I,J)
      CALL SDMAD(Z1,Z2)
   10 CALL DEFIND(S(1,I,J),Z1(1),Z1(2))
      RETURN
      END
```

```
11/30/70     11:32 AM     ASR#3.2       69157   COMPILER
  0 MIN   4 SEC FOR COMPILATION PASS
   13 CARDS AT 160 CARDS PER MINUTE
   126 DIGITS DATA.  1456 DIGITS CODE.
```

```
      SUBROUTINE PRVD(N,S,E)
      DOUBLE PRECISION S(2,10,1),E(2,10,1),ES(2,10,10),Z1(2),Z2(2),Z3(2)
      DO 10 I=1,N
      DO 10 K=1,N
      CALL DEFIND(Z3,0.,0.)
      DO 10 J=1,N
      CALL DEFIND(Z1,E(1,I,J),E(2,I,J))
      CALL DEFIND(Z2,S(1,J,K),S(2,J,K))
      CALL PRODD(Z1,Z2)
      CALL SOMAD(Z3,Z1)
   10 CALL DEFIND(ES(1,I,K),Z3(1),Z3(2))
      DO 20 I=1,N
      DO 20 J=1,N
   20 CALL DEFIND(S(1,I,J),ES(1,I,J),ES(2,I,J))
      RETURN
      END

11/30/70     11≥32 AM     ASR#3.2     69157   COMPILER
  0 MIN  6 SEC FOR COMPILATION PASS
   17 CARDS AT 154 CARDS PER MINUTE
  3390 DIGITS DATA,  2474 DIGITS CODE.
```

```
      SUBROUTINE RSTISV(N,A,KP)
      DIMENSION A(3,20,1),KP(1),S(3),Z1(3),Z2(3)
      Z=0.
      M=N+1
      DO 124 I=2,N
      CALL DEFIN(S,Z,Z,Z)
      I1=I-1
      LI=KP(I)
      DO 122 J=1,I1
      LJ=KP(J)
      CALL DEFIN(Z1,A(1,LI,J),A(2,LI,J),A(3,LI,J))
      CALL DEFIN(Z2,A(1,LJ,M),A(2,LJ,M),A(3,LJ,M))
      CALL PROD(Z1,Z2)
  122 CALL SOMA(S,Z1)
      CALL DEFIN(Z1,A(1,LI,M),A(2,LI,M),A(3,LI,M))
      CALL SUBT(Z1,S)
  124 CALL DEFIN(A(1,LI,M),Z1(1),Z1(2),Z1(3))
C
      LN=KP(N)
      CALL DEFIN(Z1,A(1,LN,M),A(2,LN,M),A(3,LN,M))
      CALL DEFIN(Z2,A(1,LN,N),A(2,LN,N),A(3,LN,N))
      CALL DIVD(Z1,Z2)
      CALL DEFIN(A(1,LN,M),Z1(1),Z1(2),Z1(3))
      N1=N-1
      DO 128 K=1,N1
      I=N-K
      LI=KP(I)
      I1=I+1
      CALL DEFIN(S,Z,Z,Z)
      DO 126 J=I1,N
      LJ=KP(J)
      CALL DEFIN(Z1,A(1,LI,J),A(2,LI,J),A(3,LI,J))
      CALL DEFIN(Z2,A(1,LJ,M),A(2,LJ,M),A(3,LJ,M))
      CALL PROD(Z1,Z2)
  126 CALL SOMA(S,Z1)
      CALL DEFIN(Z1,A(1,LI,M),A(2,LI,M),A(3,LI,M))
      CALL DEFIN(Z2,A(1,LI,I),A(2,LI,I),A(3,LI,I))
      CALL SUBT(Z1,S)
      CALL DIVD(Z1,Z2)
  128 CALL DEFIN(A(1,LI,M),Z1(1),Z1(2),Z1(3))
      RETURN
      END

11/30/70     11235 AM     ASR#3.2      69157   COMPILER
  0 MIN   9 SEC FOR COMPILATION PASS
   43 CARDS AT 275 CARDS PER MINUTE
   288 DIGITS DATA,  8944 DIGITS CODE.
```

```
      SUBROUTINE DECMAV(N,A,KP)
      DIMENSION FESC(20),A(3,20,1),KP(1),Z1(3),Z2(3),Z3(3)
      DO 50 I=1,N
      KP(I)=I
      FESC(I)=ABS(A(1,I,1))
      DO 40 J=2,N
      IF(FESC(I)-ABS(A(1,I,J)))30,40,40
   30 FESC(I)=ABS(A(1,I,J))
   40 CONTINUE
      FFF=FESC(I)
   50 FESC(I)=1./FFF
C
      N1=N-1
      DO 120 K=1,N1
C
      IMAX=K
      KMAX=KP(K)
      AMAX=FESC(KMAX)*ABS(A(1,KMAX,K))
      K1=K+1
      DO 70 K2=K1,N
      KK2=KP(K2)
      AM=FESC(KK2)*ABS(A(1,KK2,K))
      IF(AMAX-AM)60,70,70
   60 AMAX=AM
      KMAX=KK2
      IMAX=K2
   70 CONTINUE
      IF(A(1,KMAX,K))100,90,100
   90 WRITE(6,91)
   91 FORMAT(1X,17HDETERMINANTE NULO)
      STOP
  100 JOTA=KP(K)
      KP(K)=KMAX
      KP(IMAX)=JOTA
C
      DO 120 K2=K1,N
      KK2=KP(K2)
      CALL DEFIN(Z1,A(1,KK2,K),A(2,KK2,K),A(3,KK2,K))
      CALL DEFIN(Z2,A(1,KMAX,K),A(2,KMAX,K),A(3,KMAX,K))
      CALL DIVD(Z1,Z2)
      CALL DEFIN(A(1,KK2,K),Z1(1),Z1(2),Z1(3))
      DO 120 J=K1,N
      CALL DEFIN(Z2,A(1,KMAX,J),A(2,KMAX,J),A(3,KMAX,J))
      CALL PROD(Z2,Z1)
      CALL DEFIN(Z3,A(1,KK2,J),A(2,KK2,J),A(3,KK2,J))
      CALL SUBT(Z3,Z2)
  120 CALL DEFIN(A(1,KK2,J),Z3(1),Z3(2),Z3(3))
      KK2=KP(N)
      IF(A(2,KK2,N))125,90,125
  125 IF(A(1,KK2,N)*A(3,KK2,N))90,90,130
  130 RETURN
      END

11/30/70     11≥35 AM     ASR#3.2      69157   COMPILER
  0 MIN  9 SEC FOR COMPILATION PASS
   53 CARDS AT 321 CARDS PER MINUTE
   700 DIGITS DATA.  7722 DIGITS CODE.
```

```
HOLL
FILE 5=DDIR
FILE 6=DDIR
SEGMENT INVM,PRV,PRVD
SIZE REAL = 6
C       METODO DE HANSEN
        DOUBLE PRECISION AD,BD,ED,Z,ZZ,AA
        DIMENSION A(2,10,11),B(2,10,10),BD(2,10,10),E(2,10,10),P(2,10,10),
      *     S(2,10,10),AR(10,10),BR(10,10),AA(2,10,10),AD(2,10,11),Z(2),
      *     ED(2,10,10)
      3 READ(5,100,END=900)N,M
        WRITE(6,111)
        WRITE(6,444)
        N1=N
        DO 1 I=1,N
        READ(5,400)(A(1,I,J),J=1,N1)
      1 WRITE(6,500)(A(1,I,J),J=1,N1)
        WRITE(6,950)
        DO 2 I=1,N
        DO 2 J=1,N1
        ZZ=A(1,I,J)
        A(1,I,J)=ZZ-0.00005
      2 A(2,I,J)=ZZ+0.00005
        DO 20 I=1,N
        DO 20 J=1,N
     20 AR(I,J)=A(1,I,J)/2.+A(2,I,J)/2.
        CALL INVM(N,AR,BR)
        DO 30 I=1,N
        DO 30 J=1,N
        CALL DEFIN(B(1,I,J),BR(I,J),BR(I,J))
        BD(1,I,J)=B(1,I,J)
        BD(2,I,J)=B(2,I,J)
        AD(1,I,J)=A(1,I,J)
        AD(2,I,J)=A(2,I,J)
        CALL DEFIN(A(1,I,J),B(1,I,J),B(2,I,J))
     30 CALL DEFIND(ED(1,I,J),BD(1,I,J),BD(2,I,J))
C
        CALL PRVD(N,ED,AD)
C
        DO 50 I=1,N
        DO 50 J=1,N
        CALL DEFIND(AD(1,I,J),BD(1,I,J),BD(2,I,J))
        IF(I-J)42,44,42
     42 ZZ=0.
        CALL DEFIND(Z,ZZ,ZZ)
        CALL SUBTD(Z,ED(1,I,J))
        CALL DEFIND(ED(1,I,J),Z(1),Z(2))
        GO TO 50
     44 ZZ=1.
        CALL DEFIND(Z,ZZ,ZZ)
        CALL SUBTD(Z,ED(1,I,J))
        CALL DEFIND(ED(1,I,J),Z(1),Z(2))
     50 CONTINUE
C
        DO 55 I=1,N
        DO 55 J=1,N
        E(1,I,J)=ED(1,I,J)
        E(2,I,J)=ED(2,I,J)
     55 CALL ARDT2(E(1,I,J),ED(1,I,J))
C
```

(Exemplos numéricos 10-11)

```
      SOMAS=0.
      ANORMA=-1.
      DO 91 I=1,N
      IF(ANORMA-SOMAS)60,70,70
   60 ANORMA=SOMAS
   70 SOMAS=0.
      DO 91 J=1,N
      IF(ABS(E(1,I,J))-ABS(E(2,I,J)))80,80,90
   80 SOMAS=SOMAS+ABS(E(2,I,J))
      GO TO 91
   90 SOMAS=SOMAS+ABS(E(1,I,J))
   91 CONTINUE
C
      IF(ANORMA-SOMAS)92,93,93
   92 ANORMA=SOMAS
C
   93 IF(ANORMA-1.)310,300,300
  300 WRITE(6,153)ANORMA
      STOP
C
  310 DO 320 K=1,M
      DO 350 I=1,N
      DO 350 J=1,N
  350 CALL DEFIND(BD(1,I,J),AD(1,I,J),AD(2,I,J))
      RO=(ANORMA**(K+1))/(1.-ANORMA)
      CALL ARDTO(RO)
C
      CALL MPV(RO,N,F)
      CALL MSV(N,K,E,S)
      CALL SM2V(N,S,F)
      CALL PRV(N,S,A)
      CALL SM2VD(N,BD,S)
C
      WRITE(6,888)ANORMA,K,RO
      DO 333 I=1,N
      DO 333 J=1,N
  333 CALL FOID(BD(1,I,J),AA(1,I,J))
      DO 320 J=1,N
      WRITE(6,999)J
      DO 320 I=1,N
  320 WRITE(6,998)AA(1,I,J),AA(2,I,J)
      WRITE(6,777)
      GO TO 3
  900 STOP
C
  100 FORMAT(2I2)
  111 FORMAT(/,1X,11HMATRIZ DADA,//)
  153 FORMAT(1X,6HANORMA=,E16.8)
  400 FORMAT(5F7.4)
  444 FORMAT(//,1X,24HDADOS COM ERROS INICIAIS,//)
  500 FORMAT(/,1X,5F10.4)
  777 FORMAT(1H1)
  888 FORMAT(/,1X,27HMATRIZ INVERSA CONSIDERANDO,/,1X,12HNORMA DE E =,E
     *2.6,5X,2HK=,I2,5X,3HRO=,E12.6,//)
  950 FORMAT(//,1X,26HSOLUCAO SOB FORMA CENTRADA,//)
  998 FORMAT(/,1X,2E25.12)
  999 FORMAT(/,1X,6HCOLUNA,I5,/)
      END

12/04/70     10:09 AM     ASR#3.2     69157  COMPILER
```

```
      DIMENSION OP(10),D(10,3),CD(3,10),CFT(3,10,10),DRSE(3)
      ALPHA OP
      INTEGER D
      COMMON JSE,ISE,JM,N1,M,OP,D,CD,CFT,DELTA,DRSE
    1 READ(5,10,END=90)JM,N1,M,N,DELTA
      DO 20 I=1,N1
      READ(5,21)D(I,1),D(I,2),OP(I),D(I,3),CD(1,I),CD(2,I),CD(3,I)
   20 WRITE(6,22)D(I,1),D(I,2),OP(I),D(I,3),CD(1,I),CD(2,I),CD(3,I)
      READ(5,24)((CFT(K,L,1),K=1,3),L=1,M)
      DO 100 L=1,M
  100 WRITE(6,101)CFT(1,L,1),CFT(2,L,1),CFT(3,L,1)
  101 FORMAT(/,1X,3G9.3)
      CALL DERIVE
      MN=M+N
      DO 30 J=1,JM
      WRITE(6,25)J
      DO 30 L=1,MN
   30 WRITE(6,26)CFT(1,L,J),CFT(2,L,J),CFT(3,L,J)
      GO TO 1
   90 STOP
C
   10 FORMAT(4I2,E8.1)
   21 FORMAT(2I2,A2,I2,3F6.3)
   22 FORMAT(/,1X,2I2,A2,I2,3G9.3)
   24 FORMAT(12F6.2)
   25 FORMAT(/,1X,7HCOLUNA ,I2)
   26 FORMAT(/,1X,3G17.8)
      END

12/15/70      9≥29 AM     ASR#3.2      69157   COMPILER
  0 MIN 11 SEC FOR COMPILATION PASS
   29 CARDS AT 155 CARDS PER MINUTE
   312 DIGITS DATA.  4480 DIGITS CODE.  4548 DIGITS COMMON.
```

```
SYMBOLIC LISTING FOR PROGAM ≥

LOW ADRS  HIGH ADRS TEMPS     DATA BASE CODE BASE
 006602    011388    006608    006632    006908

CALLED SUBPROGRAMS ≥
NAME     ADDRESS   SEG. NO.
DERIVE    024378     001
READ.     018888     001
WRITE.    011388     001

COMMON BLOCKS REFERENCED ≥
NAME    ADDRESS
         000128
```

```
      SUBROUTINE DER
      DIMENSION OP(10),D(10,3),CD(3,10),CFT(3,10,10),DR(3),
     *     ALFA(4),T(3,10),ZZ(3)
      ALPHA OP,ALFA
      INTEGER D
      COMMON J,I,JMSE,N1SE,MSE,OP,D,CD,CFT,DELTSE,DR
      DATA ALFA/6H +    ,6H -    ,6H *    ,6H /    /
      Z=0.
      K1=D(I,1)
      K2=D(I,2)
      DO 10 I1=1,4
      IF(ALFA(I1)-OP(I))10,20,10
   10 CONTINUE
      WRITE(6,11)OP(I)
   11 FORMAT(/,1X,8HOPERACAO,A4,15HNAO RECONHECIDA)
      STOP
   20 IF(K1)30,50,30
   30 GO TO(40,41,42,43),I1
   40 CALL DEFIN(DR,1.,1.,1.)
      GO TO 44
   41 CALL DEFIN(DR,-1.,-1.,-1.)
   44 CALL PROD(DR,CFT(1,K2,J))
      CALL SOMA(DR,CFT(1,K1,J))
      RETURN
   42 CALL DEFIN(DR,Z,Z,Z)
      DO 46 JV=1,J
      CALL DEFIN(ZZ,CFT(1,K2,J+1-JV),CFT(2,K2,J+1-JV),CFT(3,K2,J+1-JV))
      CALL PROD(ZZ,CFT(1,K1,JV))
   46 CALL SOMA(DR,ZZ)
      RETURN
   43 CALL DEFIN(T(1,1),CFT(1,K1,1),CFT(2,K1,1),CFT(3,K1,1))
      CALL DIVD(T(1,1),CFT(1,K2,1))
      DO 49 JV=2,J
      CALL DEFIN(T(1,JV),Z,Z,Z)
      DO 48 IV=2,JV
      CALL DEFIN(ZZ,CFT(1,K2,IV),CFT(2,K2,IV),CFT(3,K2,IV))
      CALL PROD(ZZ,T(1,JV+1-IV))
   48 CALL SOMA(T(1,JV),ZZ)
      CALL DEFIN(ZZ,CFT(1,K1,JV),CFT(2,K1,JV),CFT(3,K1,JV))
      CALL SUBT(ZZ,T(1,JV))
      CALL DIVD(ZZ,CFT(1,K2,1))
   49 CALL DEFIN(T(1,JV),ZZ(1),ZZ(2),ZZ(3))
      CALL DEFIN(DR,T(1,J),T(2,J),T(3,J))
      RETURN
C
   50 GO TO(61,62,63,64),I1
   61 CALL DEFIN(DR,1.,1.,1.)
      GO TO 65
   62 CALL DEFIN(DR,-1.,-1.,-1.)
   65 CALL PROD(DR,CFT(1,K2,J))
      RETURN
   63 CALL DEFIN(DR,CD(1,I),CD(2,I),CD(3,I))
      CALL PROD(DR,CFT(1,K2,J))
      RETURN
   64 CALL DEFIN(T(1,1),CD(1,I),CD(2,I),CD(3,I))
      CALL DIVD(T(1,1),CFT(1,K2,1))
```

```
      DO 80 JV=2,J
      CALL DEFIN(T(1,JV),Z,Z,Z)
      DO 70 IV=2,JV
      CALL DEFIN(ZZ,CFT(1,K2,IV),CFT(2,K2,IV),CFT(3,K2,IV))
      CALL PROD(ZZ,T(1,JV+1-IV))
   70 CALL SOMA(T(1,JV),ZZ)
      CALL DEFIN(ZZ,-1.,-1.,-1.)
      CALL PROD(T(1,JV),ZZ)
   80 CALL DIVD(T(1,JV),CD(1,I))
      CALL DEFIN(DR,T(1,J),T(2,J),T(3,J))
      RETURN
      END

12/15/70      9≥29 AM     ASR#3.2      69157   COMPILER
  0 MIN 38 SEC FOR COMPILATION PASS
   70 CARDS AT 109 CARDS PER MINUTE
   872 DIGITS DATA. 10866 DIGITS CODE.  4548 DIGITS COMMON.
```

```
      SUBROUTINE DERIVE
      DIMENSION OPSE(10),D(10,3),CDSE(3,10),CFT(3,10,10),DR(3),ZZ(3)
      INTEGER D
      ALPHA OPSE
      COMMON J,I,JM,N1,M,OPSE,D,CDSE,CFT,DELTA,DR
      JF=JM-1
      DO 30 J=1,JF
      DO 30 I=1,N1
      IF(D(I,3)-M)20,20,10
   10 L=D(I,3)
      CALL DER
      CALL DEFIN(CFT(1,L,J),DR(1),DR(2),DR(3))
      GO TO 30
   20 L=D(I,3)
      J1=J+1
      CALL DER
      CALL DEFIN(ZZ,DELTA,DELTA,DELTA)
      CALL PROD(DR,ZZ)
      Z=J
      CALL DEFIN(ZZ,Z,Z,Z)
      CALL DIVD(DR,ZZ)
      CALL DEFIN(CFT(1,L,J1),DR(1),DR(2),DR(3))
   30 CONTINUE
      RETURN
      END

12/15/70      9≥29 AM     ASR#3.2      69157   COMPILER
  0 MIN  6 SEC FOR COMPILATION PASS
   26 CARDS AT 245 CARDS PER MINUTE
   116 DIGITS DATA.  1274 DIGITS CODE.  4548 DIGITS COMMON.
```

APÊNDICE C

Exemplo 1

Matriz completa dos coeficientes de um sistema linear de ordem 10

2.0	6.0	8.0	14.0	.0	-2.0	10.0	-6.0	-4.0	.0	28.0
7.0	-21.0	-28.0	-42.0	8.0	-1.0	-44.0	12.0	13.0	-1.0	-97.0
1.0	-2.0	-.3	-1.0	2.2	5.0	.1	3.8	.7	-.9	8.6
2.0	-4.0	-2.6	2.9	-24.4	-32.5	-5.1	-20.1	-1.5	-1.5	-86.8
-3.0	11.0	-1.6	5.9	43.8	42.1	5.2	28.2	6.1	17.1	154.8
10.0	-35.0	9.6	-38.9	-264.0	-6.3	2.8	35.2	34.9	59.1	-192.6
1.0	.0	.0	.0	.0	.0	.0	.0	.0	-5.0	-4.0
6.0	-18.5	-1.3	-2.0	-40.2	10.4	13.1	72.1	61.1	180.0	280.7
-8.0	34.0	-10.3	19.9	107.4	-50.6	-37.3	-341.3	-13.9	186.5	-113.6
6.0	-16.5	-5.0	10.9	52.2	-3.3	13.7	-26.0	-4.8	39.4	66.6

Primeira Solução

		$\underline{x}$	$\tilde{x}$	$\overline{x}$
KP(1)= 3	X(1)=	.99867720E+00	.99999890E+00	.10013231E+01
KP(2)= 1	X(2)=	.99951690E+00	.99999919E+00	.10004833E+01
KP(3)= 2	X(3)=	.99974547E+00	.10000020E+01	.10002551E+01
KP(4)=10	X(4)=	.99988578E+00	.10000000E+01	.10001138E+01
KP(5)= 4	X(5)=	.99998141E+00	.10000003E+01	.10000187E+01
KP(6)= 5	X(6)=	.99998415E+00	.99999978E+00	.10000157E+01
KP(7)= 7	X(7)=	.99996948E+00	.99999958E+00	.10000308E+01
KP(8)= 9	X(8)=	.99999613E+00	.10000000E+01	.10000039E+01
KP(9)= 8	X(9)=	.99996698E+00	.10000009E+01	.10000333E+01
KP(10)= 6	X(10)=	.99999471E+00	.99999996E+00	.10000053E+01

Solução sob forma centrada

		$\frac{\underline{x} + \overline{x}}{2}$	$\tilde{x}$	$\frac{\overline{x} - \underline{x}}{2}$
KP(1)= 3	X(1) =	.10000001E+01	.99999890E+00	.13229500E-02
KP(2)= 1	X(2) =	.10000001E+01	.99999919E+00	.48320000E-03
KP(3)= 2	X(3) =	.10000002E+01	.10000020E+01	.25481501E-03
KP(4)=10	X(4) =	.99999979E+00	.10000000E+01	.11401001E-03
KP(5)= 4	X(5) =	.10000000E+01	.10000003E+01	.18645001E-04
KP(6)= 5	X(6) =	.99999992E+00	.99999978E+00	.15775001E-04
KP(7)= 7	X(7) =	.10000001E+01	.99999958E+00	.30660001E-04
KP(8)= 9	X(8) =	.10000000E+01	.10000000E+01	.38850001E-05
KP(9)= 8	X(9) =	.10000001E+01	.10000009E+01	.33160001E-04
KP(10)= 6	X(10) =	.10000000E+01	.99999996E+00	.52950001E-05

Delimitação do êrro de arredondamento

Segunda Solução

KP	X	$\underline{x}$	$\tilde{x}$	$\bar{x}$
KP(1)= 3	X(1)=	.99999987E+00	.99999999E+00	.10000001E+01
KP(2)= 1	X(2)=	.99999995E+00	.99999999E+00	.10000000E+01
KP(3)= 2	X(3)=	.99999997E+00	.10000000E+01	.10000000E+01
KP(4)=10	X(4)=	.99999999E+00	.10000000E+01	.10000000E+01
KP(5)= 4	X(5)=	.99999999E+00	.10000000E+01	.10000000E+01
KP(6)= 5	X(6)=	.99999999E+00	.99999999E+00	.10000000E+01
KP(7)= 7	X(7)=	.99999999E+00	.99999999E+00	.10000000E+01
KP(8)= 9	X(8)=	.99999999E+00	.10000000E+01	.10000000E+01
KP(9)= 8	X(9)=	.99999999E+00	.10000000E+01	.10000000E+01
KP(10)= 6	X(10)=	.99999999E+00	.10000000E+01	.10000000E+01

Solução sob forma Centrada

KP	X	$\frac{\underline{x} + \bar{x}}{2}$	$\tilde{x}$	$\frac{\bar{x} - \underline{x}}{2}$
KP(1)= 3	X(1) =	.99999999E+00	.99999999E+00	.13136500E-06
KP(2)= 1	X(2) =	.99999999E+00	.99999999E+00	.47947500E-07
KP(3)= 2	X(3) =	.10000000E+01	.10000000E+01	.25280500E-07
KP(4)=10	X(4) =	.99999999E+00	.10000000E+01	.11329000E-07
KP(5)= 4	X(5) =	.10000000E+01	.10000000E+01	.18580000E-08
KP(6)= 5	X(6) =	.99999999E+00	.99999999E+00	.15775000E-08
KP(7)= 7	X(7) =	.99999999E+00	.99999999E+00	.30135000E-08
KP(8)= 9	X(8) =	.10000000E+01	.10000000E+01	.39900000E-09
KP(9)= 8	X(9) =	.10000000E+01	.10000000E+01	.32370000E-08
KP(10)= 6	X(10) =	.99999999E+00	.10000000E+01	.52150000E-09

Exemplo 2

Primeira Solução

SISTEMA LINEAR

2.1546000	.8431000	.3146000	.1615000	3.1826000
2.8431000	3.1415000	.6241000	.2131000	4.6123000
.3146000	.6241000	4.8216000	.8245000	5.9681000
.1615000	.2131000	.8245000	6.4131000	8.1418000

SOLUCAO COM TERNOS

KP(1)= 1	X(1)=	.121318869600E+01	.121318869601E+01	.121318869604E+01
KP(2)= 2	X(2)=	.105076868917E+00	.105076868939E+00	.105076868945E+00
KP(3)= 3	X(3)=	.954740092022E+00	.954740092028E+00	.954740092038E+00
KP(4)= 4	X(4)=	.111276838641E+01	.111276838642E+01	.111276838643E+01

SOLUCAO SOB FORMA CENTRADA

KP(1)= 1	X(1)=	.121318869602E+01	.121318869601E+01	.200000000000E-10
KP(2)= 2	X(2)=	.105076868931E+00	.105076868939E+00	.140000000000E-10
KP(3)= 3	X(3)=	.954740092030E+00	.954740092028E+00	.800000000000E-11
KP(4)= 4	X(4)=	.111276838642E+01	.111276838642E+01	.100000000000E-10

Segunda Solução

SISTEMA LINEAR - CCEFICIENTES CCM ERRCS INICIAIS

2.1546	.8431	.3146	.1615	3.1826
2.8431	3.1415	.6241	.2131	4.6123
.3146	.6241	4.8216	.8245	5.9681
.1615	.2131	.8245	6.4131	8.1418

SCLUCAC CCM TERNCS

KF(1)= 1	X(1)=	.121295203286E+01	.121318869601E+01	.121342533593E+01
KF(2)= 2	X(2)=	.104785086988E+00	.105076868939E+00	.105368683999E+00
KF(3)= 3	X(3)=	.954633968721E+00	.954740092028E+00	.954846213198E+00
KF(4)= 4	X(4)=	.111269904025E+01	.111276838642E+01	.111283773122E+01

SCLUCAC SCB FCRMA CENTRADA

KF(1)= 1	X(1)=	.121318868439E+01	.121318869601E+01	.236651535000E-03
KF(2)= 2	X(2)=	.105076885493E+00	.105076868939E+00	.291798505500E-03
KF(3)= 3	X(3)=	.954740090959E+00	.954740092028E+00	.106122238500E-03
KF(4)= 4	X(4)=	.111276838573E+01	.111276838642E+01	.693454850000E-04

Exemplo 3

Primeira Solução

SISTEMA LINEAR

6.4375000	2.9042000	-7.1313000	5.8024000	2.1345000
2.1356000	1.0124000	-2.3121000	1.9011000	1.3161000
-3.7362000	-1.6421000	4.0526000	-3.3515000	1.8422000
1.8666000	.8526000	-2.0041000	1.6824000	2.4430000

SOLUCAO COM TERNOS

KP(1)= 4	X(1)=	-.128304678530E+03	-.128304678213E+03	-.128304677769E+03
KP(2)= 2	X(2)=	.581347636982E+02	.581347639479E+02	.581347641964E+02
KP(3)= 1	X(3)=	.191593419872E+02	.191593420430E+02	.191593420838E+02
KP(4)= 3	X(4)=	.137165982803E+03	.137165983120E+03	.137165983277E+03

SOLUCAO SOB FORMA CENTRADA

KP(1)= 4	X(1)=	-.128304678149E+03	-.128304678213E+03	.380500000000E-06
KP(2)= 2	X(2)=	.581347639473E+02	.581347639479E+02	.249100000000E-06
KP(3)= 1	X(3)=	.191593420355E+02	.191593420430E+02	.483000000000E-07
KP(4)= 3	X(4)=	.137165983040E+03	.137165983120E+03	.237000000000E-06

Segunda Solução

SISTEMA LINEAR - COEFICIENTES COM ERROS INICIAIS

6.4375	2.9042	-7.1313	5.8024	2.1345
2.1356	1.0124	-2.3121	1.9011	1.3161
-3.7362	-1.6421	4.0526	-3.3515	1.8422
1.8666	.8526	-2.0041	1.6824	2.4430

SOLUCAO COM TERNOS

KF(1)= 4	X(1)=	-.136674423099E+03	-.128304678213E+03	-.120374530823E+03
KF(2)= 2	X(2)=	.532552917084E+02	.581347639479E+02	.632751825235E+02
KF(3)= 1	X(3)=	.179418810663E+02	.191593420430E+02	.203147035140E+02
KF(4)= 3	X(4)=	.132226398825E+03	.137165983120E+03	.142386066038E+03

SOLUCAO SOB FORMA CENTRADA

KF(1)= 4	X(1)=	-.128524476961E+03	-.128304678213E+03	.814994613800E+01
KF(2)= 2	X(2)=	.582652371159E+02	.581347639479E+02	.500994540755E+01
KF(3)= 1	X(3)=	.191282922901E+02	.191593420430E+02	.118641122385E+01
KF(4)= 3	X(4)=	.137306232431E+03	.137165983120E+03	.507983360650E+01

EXEMPLO 4

MATRIZ DADA

2.1546000	.8431000	.3146000	.1615000
2.8431000	3.1415000	.6241000	.2131000
.3146000	.6241000	4.8216000	.8245000
.1615000	.2131000	.8245000	6.4131000

MATRIZ INVERSA - FORMA CENTRADA

COLUNA 1

.714649380578E+00	.714649380571E+00	.750000000000E-11
-.654288813227E+00	-.654288813221E+00	.115000000000E-10
.382614689488E-01	.382614689481E-01	.130000000000E-11
-.117470646221E-02	-.117470646232E-02	.390000000000E-12

COLUNA 2

-.187031776637E+00	-.187031776634E+00	.300000000000E-11
.498148220849E+00	.498148220844E+00	.550000000000E-11
-.513804655422E-01	-.513804655420E-01	.700000000000E-12
-.523718016195E-02	-.523718016189E-02	.225000000000E-12

COLUNA 3

-.208643457703E-01	-.208643457701E-01	.700000000000E-12
-.222649032646E-01	-.222649032645E-01	.650000000000E-12
.216179623942E+00	.216179623941E+00	.200000000000E-11
-.265278659636E-01	-.265278659636E-01	.250000000000E-12

COLUNA 4

-.909961645298E-02	-.909961645293E-02	.285000000000E-12
.278643249209E-02	.278643249200E-02	.170000000000E-12
-.270493443057E-01	-.270493443056E-01	.450000000000E-12
.159544991295E+00	.159544991295E+00	.100000000000E-11

EXEMPLO 5

ATRIZ DADA

-6.1375000	2.9042000	-7.1313000	5.8024000
2.1356000	1.0124000	-2.3121000	1.9011000
-3.7362000	-1.6421000	4.0526000	-3.3515000
1.8666000	.8526000	-2.0041000	1.6824000

ATRIZ INVERSA - FORMA CENTRADA

COLUNA 1

-.258258793397E+01	-.258258793423E+01	.614000000000E-08
.341822085963E+00	.341822085871E+00	.371000000000E-08
-.108098450702E+01	-.108098450645E+01	.205500000000E-08
.140443418719E+01	.140443418822E+01	.247000000000E-08

COLUNA 2

.467175849234E+01	.467175849271E+01	.697000000000E-07
.100895932973E+02	.100895932988E+02	.425500000000E-07
-.806225519126E+01	-.806225519231E+01	.239150000000E-07
-.199002955753E+02	-.199002955777E+02	.272000000000E-07

COLUNA 3

-.572321263125E+01	-.572321263184E+01	.244350000000E-07
.919707473575E+01	.919707473570E+01	.172100000000E-07
.630829553109E+01	.630829553253E+01	.812500000000E-08
.920350561795E+01	.920350562037E+01	.860500000000E-08

COLUNA 4

-.777319244846E+01	-.777319244926E+01	.111330000000E-06
.574137047547E+01	.574137047369E+01	.724650000000E-07
.254052010938E+02	.254052010957E+02	.385500000000E-07
.365721065556E+02	.365721065595E+02	.408000000000E-07

EXEMPLO 6

MATRIZ DADA - DADOS COM ERROS INICIAIS

2.1546000	.8431000	.3146000	.1615000
2.8431000	3.1415000	.6241000	.2131000
.3146000	.6241000	4.8216000	.8245000
.1615000	.2131000	.8245000	6.4131000

MATRIZ INVERSA - FORMA CENTRADA

COLUNA 1

.714649387682E+00	.714649380571E+00	.710941410000E-04
-.654288823922E+00	-.654288813221E+00	.841053400000E-04
.382614718294E-01	.382614689481E-01	.287651540500E-04
-.117470647846E-02	-.117470646232E-02	.173497296750E-04

COLUNA 2

-.187031779944E+00	-.187031776634E+00	.377166170000E-04
.498148226340E+00	.498148220844E+00	.448860045000E-04
-.513804666690E-01	-.513804655420E-01	.153562006500E-04
-.523718045879E-02	-.523718016189E-02	.928722866500E-05

COLUNA 3

-.208643451901E-01	-.208643457701E-01	.149466622500E-04
-.222649049497E-01	-.222649032645E-01	.182032465500E-04
.216179624583E+00	.216179623941E+00	.622579850000E-05
-.265278659121E-01	-.265278659636E-01	.375509510000E-05

COLUNA 4

-.909961748592E-02	-.909961645293E-02	.101903974900E-04
.278643396021E-02	.278643249200E-02	.122275951050E-04
-.270493445133E-01	-.270493443056E-01	.418225315000E-05
.159544991388E+00	.159544991295E+00	.252253050000E-05

EXEMPLO 7

MATRIZ DADA - DADOS COM ERROS INICIAIS

6.1375000	2.9042000	-7.1313000	5.8024000
2.1356000	1.0124000	-2.3121000	1.9011000
-3.7362000	-1.6421000	4.0526000	-3.3515000
1.8666000	.8526000	-2.0041000	1.6824000

MATRIZ INVERSA - FORMA CENTRADA

COLUNA 1

-.258323742829E+01	-.258258793423E+01	.116743993950E+00
.342986214912E+00	.341822085871E+00	.671617465255E-01
-.108037912734E+01	-.108098450645E+01	.399910755100E-01
.140527790555E+01	.140443418822E+01	.476911112250E-01

COLUNA 2

.467602667667E+01	.467175849271E+01	.145562157294E+01
.100757447612E+02	.100895932988E+02	.838387513381E+00
-.807205481280E+01	-.806225519231E+01	.516369271970E+00
-.199096663684E+02	-.199002955777E+02	.573740906700E+00

COLUNA 3

-.572490224841E+01	-.572321263184E+01	.470297430405E+00
.920277552701E+01	.919707473570E+01	.319110815315E+00
.631072562631E+01	.630829553253E+01	.160620834920E+00
.920536229150E+01	.920350562037E+01	.167832203265E+00

COLUNA 4

-.778243421674E+01	-.777319244926E+01	.224570967166E+01
.577022549055E+01	.574137047369E+01	.134437084374E+01
.254180477741E+02	.254052010957E+02	.808566070250E+00
.365829325254E+02	.365721065595E+02	.844868133700E+00

EXEMPLO 8

MATRIZ DADA

2.1546	.8431	.3146	.1615
2.8431	3.1415	.6241	.2131
.3146	.6241	4.8216	.8245
.1615	.2131	.8245	6.4131

SOLUCAO SOB FORMA CENTRADA

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .331887E-04   M= 1   RO= .110152E-08

COLUNA 1

.714649380495E+00	.110600000000E-08
-.654288813050E+00	.155050000000E-08
.382614689401E-01	.390100000000E-09
-.117470646185E-02	.213755000000E-09

COLUNA 2

-.187031776586E+00	.107550000000E-08
.498148220716E+00	.139600000000E-08
-.513804655380E-01	.380050000000E-09
-.523718016250E-02	.214505000000E-09

COLUNA 3

-.208643457651E-01	.103205000000E-08
-.222649032616E-01	.130355000000E-08
.216179623901E+00	.411000000000E-09
-.265278659601E-01	.218050000000E-09

COLUNA 4

-.909961645220E-02	.102740500000E-08
.278643248851E-02	.130051000000E-08
-.270493442976E-01	.372550000000E-09
.159544991289E+00	.218000000000E-09

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .331887E-04 M= 2 RO= .365577E-13

COLUNA 1

.714649380590E+00	.151000000000E-09
-.654288813300E+00	.400500000000E-09
.382614689601E-01	.401000000000E-10
-.117470646175E-02	.375500000000E-11

COLUNA 2

-.187031776636E+00	.955000000000E-10
.498148220846E+00	.226000000000E-09
-.513804655435E-01	.325500000000E-10
-.523718016100E-02	.600500000000E-11

COLUNA 3

-.208643457691E-01	.120500000000E-10
-.222649032626E-01	.125500000000E-10
.216179623926E+00	.960000000000E-10
-.265278659616E-01	.115500000000E-10

COLUNA 4

-.909961645265E-02	.255500000000E-11
.278643249201E-02	.501000000000E-11
-.270493443066E-01	.105500000000E-10
.159544991294E+00	.120000000000E-10

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .331887E-04 M= 3 RO= .121329E-17

COLUNA 1

.714649380595E+00	.126000000000E-09
-.654288813250E+00	.350500000000E-09
.382514689551E-01	.351000000000E-10
-.117470646180E-02	.320500000000E-11

COLUNA 2

-.187031776636E+00	.755000000000E-10
.498148220846E+00	.176000000000E-09
-.513804655435E-01	.255500000000E-10
-.523718016100E-02	.500500000000E-11

COLUNA 3

-.208643457691E-01	.100500000000E-10
-.222649032626E-01	.105500000000E-10
.216179623926E+00	.760000000000E-10
-.265278659616E-01	.955000000000E-11

COLUNA 4

-.909961645265E-02	.225500000000E-11
.278643249201E-02	.501000000000E-11
-.270493443061E-01	.805000000000E-11
.159544991294E+00	.100000000000E-10

EXEMPLO 9

MATRIZ DADA

6.1375	2.9042	-7.1313	5.8024
2.1356	1.0124	-2.3121	1.9011
-3.7362	-1.6421	4.0526	-3.3515
1.8666	.8526	-2.0041	1.6824

SOLUCAO SOB FORMA CENTRADA

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .612817E-03 M= 1 RO= .375774E-06

COLUNA 1

-.258258790900E+01	.780200500000E-05
.341822094500E+00	.953650050000E-05
-.108098455501E+01	.153590050000E-04
.140443409650E+01	.252165050000E-04

COLUNA 2

.467175846951E+01	.781051000000E-05
.100895932771E+02	.954510000000E-05
-.806225514150E+01	.153705050000E-04
-.199002954796E+02	.252505500000E-04

COLUNA 3

-.572321259500E+01	.782100500000E-05
.919707474700E+01	.956401000000E-05
.630329551500E+01	.154050050000E-04
.920350556000E+01	.252900050000E-04

COLUNA 4

-.777319242750E+01	.782750500000E-05
.574137044750E+01	.957451000000E-05
.254052011350E+02	.154050500000E-04

.365721066000E+02 .252700500000E-04

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .612817E-03 M= 2 , RO= .230280E-09

COLUNA 1

-.258258793350E+01 .115050000000E-07
.341822086000E+00 .110005000000E-07
-.108098450701E+01 .210050000000E-07
.140443418750E+01 .335050000000E-07

COLUNA 2

.467175849201E+01 .220100000000E-07
.100895932971E+02 .241000000000E-07
-.806225519000E+01 .330050000000E-07
-.199002755721E+02 .770500000000E-07

COLUNA 3

-.572391253100E+01 .750050000000E-07
.919707474350E+01 .785100000000E-07
.630329553500E+01 .165005000000E-06
.920150561500E+01 .245005000000E-06

COLUNA 4

-.777319244550E+01 .515050000000E-07
.574137046650E+01 .745100000000E-07
.254052010850E+02 .750500000000E-07
.365721065550E+02 .105050000000E-06

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .612817E-03 M= 3 RO= .141120E-12

COLUNA 1

-.258258793450E+01 .650500000000E-08

.341822085500E+00	.550050000000E-08
-.108098450701E+01	.130050000000E-07
.140443418750E+01	.195050000000E-07

COLUNA 2

.467175849151E+01	.175100000000E-07
.100895932976E+02	.186000000000E-07
-.806225519000E+01	.240050000000E-07
-.199002955666E+02	.535500000000E-07

COLUNA 3

-.572321263100E+01	.740050000000E-07
.919707474350E+01	.765100000000E-07
.630829553500E+01	.165005000000E-06
.920350562000E+01	.240005000000E-06

COLUNA 4

-.777319244600E+01	.490050000000E-07
.574137046650E+01	.735100000000E-07
.254052010900E+02	.700500000000E-07
.365721065550E+02	.950500000000E-07

EXEMPLO 10

MATRIZ DADA

DADOS COM ERROS INICIAIS

2.1546	.8431	.3146	.1615
2.8431	3.1415	.6241	.2131
.3146	.6241	4.8216	.8245
.1615	.2131	.8245	6.4131

SOLUCAO SOB FORMA CENTRADA

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .147175E-03   M= 1   RC= .216635E-07

COLUNA 1

.714649380550E+00	.656258510000E-04
-.654288813250E+00	.829429510000E-04
.382614690001E-01	.234476001000E-04
-.117470645001E-02	.135587500100E-04

COLUNA 2

-.187031776651E+00	.345749510000E-04
.498148220851E+00	.436984510000E-04
-.513804655500E-01	.123533501000E-04
-.523718016000E-02	.714336001000E-05

COLUNA 3

-.208643458501E-01	.133352501000E-04
-.222649032001E-01	.168540001000E-04
.216179623916E+00	.476448600000E-05
-.265278659601E-01	.275510010000E-05

COLUNA 4

-.909961643500E-02	.926585501000E-05
.278643250001E-02	.117110000100E-04
-.270493443101E-01	.331061010000E-05
.159544991298E+00	.191438400000E-05

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .147175E-03 . M= 2 RC= .318831E-11

COLUNA 1

.714649380500E+00	.656148010000E-04
-.654288813250E+00	.829290510000E-04
.382614689501E-01	.234437501000E-04
-.117470650001E-02	.135565000100E-04

COLUNA 2

-.187031776651E+00	.345601510000E-04
.498148220951E+00	.436797510000E-04
-.513804655500E-01	.123481501000E-04
-.523718015500E-02	.714034501000E-05

COLUNA 3

-.208643458001E-01	.133173001000E-04
-.222649032501E-01	.168313501000E-04
.216179624006E+00	-.475804600000E-05
-.265278659751E-01	.275140510000E-05

COLUNA 4

-.909961639000E-02	.924716001000E-05
.278643245001E-02	.116873500100E-04
-.270493443451E-01	.330394510000E-05
.159544991330E+00	.191051150000E-05

MATRIZ INVERSA CONSIDERANDO

NORMA DE E = .147175E-03   N= 3   RC= .469238E-15

COLUNA 1

.71464938050 0E+00	.656148010000E-04
-.654288813250E+00	.829290510000E-04
.382614689501E-01	.234437501000E-04
-.117470650001E-02	.135565000100E-04

COLUNA 2

-.187031776651E+00	.345601510000E-04
.498148220951E+00	.436797510000E-04
-.513804655500E-01	.123481501000E-04
-.523718015500E-02	.714034501000E-05

COLUNA 3

-.208643458001E-01	.133173001000E-04
-.222649032501E-01	.168313501000E-04
.216179624006E+00	.475804600000E-05
-.265278659751E-01	.275140510000E-05

COLUNA 4

-.909961639000E-02	.924716001000E-05
.278643245001E-02	.116873500100E-04
-.270493443451E-01	.330394510000E-05
.159544991330E+00	.191051150000E-05

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .147175E-03   N= 4   RC= .690600E-19

COLUNA 1

.714649380500E+00	.656147010000E-04
-.654288813300E+00	.829289010000E-04
.382614689501E-01	.234437501000E-04

-.117470650001E-02 .135565000100E-04

COLUNA 2

-.187031776651E+00 .345601510000E-04

.498148220901E+00 .436796010000E-04

-.513804655000E-01 .123481001000E-04

-.523718015500E-02 .714033501000E-05

COLUNA 3

-.208643458501E-01 .133172501000E-04

-.222649032501E-01 .168311501000E-04

.216179623996E+00 .475802600000E-05

-.265278659751E-01 .275138510000E-05

COLUNA 4

-.909961642000E-02 .924710001000E-05

.278643245001E-02 .116873500100E-04

-.270493443401E-01 .330392010000E-05

.159544991327E+00 .191050500000E-05

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .147175E-03 N= 5 RC= .101639E-22

COLUNA 1

.714649380500E+00 .656147010000E-04

-.654288813300E+00 .829289010000E-04

.382614689501E-01 .234437501000E-04

-.117470650001E-02 .135565000100E-04

COLUNA 2

-.187031776651E+00 .345601510000E-04

.498148220901E+00 .436796010000E-04

-.513804655000E-01 .123481001000E-04

-.523718015500E-02 .714033501000E-05

COLUNA 3

-.208643458501E-01	.133172501000E-04
-.222649032501E-01	.168311501000E-04
.216179623996E+00	.475802600000E-05
-.265278659751E-01	.275138510000E-05

COLUNA 4

-.909961642000E-02	.924710001000E-05
.278643245001E-02	.116873500100E-04
-.270493443401E-01	.330392010000E-05
.159544991327E+00	.191050500000E-05

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .147175E-03   N= 6   RC= .149586E-26

COLUNA 1

.714649380500E+00	.656147010000E-04
-.654288813300E+00	.829289010000E-04
.382614689501E-01	.234437501000E-04
-.117470650001E-02	.135565000100E-04

COLUNA 2

-.187031776651E+00	.345601510000E-04
.498148220901E+00	.436796010000E-04
-.513804655000E-01	.123481001000E-04
-.523718015500E-02	.714033501000E-05

COLUNA 3

-.208643458501E-01	.133172501000E-04
-.222649032501E-01	.168311501000E-04
.216179623996E+00	.475802600000E-05
-.265278659751E-01	.275138510000E-05

COLUNA 4

-.909961642000E-02	.924710001000E-05
.278643245001E-02	.116873500100E-04
-.270493443401E-01	.330392010000E-05
.159544991327E+00	.191050500000E-05

-.909961642000E-02	.924710001000E-05
.278643245001E-02	.116873500100E-04
-.270493443401E-01	.330392010000E-05
.159544991327E+00	.191050500000E-05

EXEMPLO 11

MATRIZ DADA

DADOS COM ERROS INICIAIS

6.1375	2.9042	-7.1313	5.8024
2.1356	1.0124	-2.3121	1.9011
-3.7362	-1.6421	4.0526	-3.3515
1.8666	.9526	-2.0041	1.6824

SOLUCAO SOB FORMA CENTRADA

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .829016E-02 N= 1 RC= .693013E-04

COLUNA 1

-.258258787001E+01	.705272001000E-02
.341822065001E+00	.862075500100E-02
-.108098460001E+01	.138858000100E-01
.140443405001E+01	.227992500100E-01

COLUNA 2

.467175830001E+01	.457844000100E-01
.100895933001E+02	.559637001000E-01
-.806225475000E+01	.901425500100E-01
-.199002950001E+02	.148006000100E+00

COLUNA 3

-.572321260000E+01	.330189000100E-01
.919707470000E+01	.403601000100E-01
.630829545000E+01	.650093500100E-01
.920350590000E+01	.106739500010E+00

COLUNA 4

-.777319220000E+01	.797872000100E-01
.574137040000E+01	.975263000100E-01
.254052005001E+02	.157089500100E+00
.365721055001E+02	.257926500100E+00

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .829016E-02 M= 2 RC= .574518E-06

COLUNA 1

-.258258855501E+01	.567207501000E-02
.341822385001E+00	.693290500100E-02
-.108098300001E+01	.111671000100E-01
.140443660001E+01	.183353000100E-01

COLUNA 2

.467175695001E+01	.447102500100E-01
.100895957501E+02	.546487501000E-01
-.806225405000E+01	.880251500100E-01
-.199002935001E+02	.144529500100E+00

COLUNA 3

-.572321360000E+01	.318461000100E-01
.919707610000E+01	.389244000100E-01
.630829830000E+01	.626991000100E-01
.920350900000E+01	.102945000100E+00

COLUNA 4

-.777318535000E+01	.789653500100E-01
.574137045000E+01	.965219500100E-01
.254051905001E+02	.155472500100E+00
.365720855001E+02	.255271500100E+00

MATRIZ INVERSA CONSIDERANDO

NORMA DE E = .829016E-02 N= 3 RC= .476284E-08

COLUNA 1

-.258258854501E+01	.566062501000E-02
.341822385001E+00	.691891501000E-02
-.108098295001E+01	.111445501000E-01
.140443660001E+01	.182982000100E-01

COLUNA 2

.467175690001E+01	.447011000100E-01
.100895958001E+02	.546376001000E-01
-.806225410000E+01	.880072000100E-01
-.199002940001E+02	.144500000100E+00

COLUNA 3

-.572321365000E+01	.318361501000E-01
.919707625000E+01	.389121501000E-01
.630829835000E+01	.626796501000E-01
.920350900000E+01	.102913000010E+00

COLUNA 4

-.777318535000E+01	.789583501000E-01
.574137050000E+01	.965134001000E-01
.254051905001E+02	.155458501000E+00
.365720860001E+02	.255248000100E+00

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .829016E-02 N= 4 RC= .394847E-10

COLUNA 1

-.258258854501E+01	.566053501000E-02
.341822385001E+00	.691881501000E-02
-.108098300001E+01	.111444001000E-01

.140443655001E+01	.182979500100E-01

COLUNA 2

.467175685001E+01	.447011500100E-01
.100895958501E+02	.546376501000E-01
-.806225410000E+01	.880072000100E-01
-.199002940001E+02	.144500000100E+00

COLUNA 3

-.572321365000E+01	.318361500100E-01
.919707625000E+01	.389121500100E-01
.630829835000E+01	.626796500100E-01
.920350900000E+01	.102913000010E+00

COLUNA 4

-.777318540000E+01	.789584000100E-01
.574137050000E+01	.965134000100E-01
.254051905001E+02	.155458500100E+00
.365720860001E+02	.255248000100E+00

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .829016E-02 N= 5 RC= .327334E-12

COLUNA 1

-.258258854501E+01	.566053501000E-02
.341822385001E+00	.691881500100E-02
-.108098300001E+01	.111444000100E-01
.140443655001E+01	.182979500100E-01

COLUNA 2

.467175685001E+01	.447011500100E-01
.100895958501E+02	.546376501000E-01
-.806225410000E+01	.880072000100E-01
-.199002940001E+02	.144500000100E+00

```
COLUNA     3

   -.572321365000E+01          .318361500100E-01
    .919707625000E+01          .389121500100E-01
    .630829835000E+01          .626796500100E-01
    .920350900000E+01          .102913000010E+00

COLUNA     4

   -.777318540000E+01          .789584000100E-01
    .574137050000E+01          .965134000100E-01
    .254051905001E+02          .155458500100E+00
    .365720860001E+02          .255248000100E+00

MATRIZ INVERSA CONSIDERANDO
NORMA DE E = .829016E-02      N= 6 ,  RC= .271365E-14


COLUNA     1

   -.258258854501E+01          .566053501000E-02
    .341822385001E+00          .691881500100E-02
   -.108098300001E+01          .111444000100E-01
    .140443655001E+01          .182979500100E-01

COLUNA     2

    .467175685001E+01          .447009500100E-01
    .100895958001E+02          .546374001000E-01
   -.806225405000E+01          .880067500100E-01
   -.199002940001E+02          .144499000100E+00

COLUNA     3

   -.572321360000E+01          .318360000100E-01
    .919707620000E+01          .389119000100E-01
    .630829835000E+01          .626791500100E-01
    .920350950000E+01          .102912500010E+00

COLUNA     4
```

-.777318540000E+01	.789582000100E-01
.574137050000E+01	.965132000100E-01
.254051905001E+02	.155458500100E+00
.365720855001E+02	.255247500100E+00

EXEMPLO 12

```
1 2 * 5 .000E-99 .000E-99 .000E-99
3 4 w 6 .000E-99 .000E-99 .000E-99
5 6 + 1 .000E-99 .000E-99 .000E-99            Descrição do Sistema
3 4 - 7 .000E-99 .000E-99 .000E-99
2 7 * 2 .000E-99 .000E-99 .000E-99
1 2 / 3 .000E-99 .000E-99 .000E-99
1 2 + 4 .000E-99 .000E-99 .000E-99
```

```
1.00      1.00      1.00
1.00      1.00      1.00
1.00      1.00      1.00                      Valores Ini
1.00      1.00      1.00
```

```
COLUNA  1
    1.0000000          1.0000000          1.0000000
    1.0000000          1.0000000          1.0000000
    1.0000000          1.0000000          1.0000000
    1.0000000          1.0000000          1.0000000
    1.0000000          1.0000000          1.0000000
    1.0000000          1.0000000          1.0000000
    .00000000E-99      .00000000E-99      .00000000E-99
COLUNA  2
    2.0000000          2.0000000          2.0000000
    .00000000E-99      .00000000E-99      .00000000E-99
    1.0000000          1.0000000          1.0000000
    2.0000000          2.0000000          2.0000000
    2.0000000          2.0000000          2.0000000
    3.0000000          3.0000000          3.0000000
   -1.0000000         -1.0000000         -1.0000000
COLUNA  3
    2.5000000          2.5000000          2.5000000
   -.50000000         -.50000000         -.50000000
```

1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000
2.0000000	2.0000000	2.0000000
4.0000000	4.0000000	4.0000000
.00000000E-99	.00000000E-99	.00000000E-99

COLUNA 4

2.0000000	2.0000000	2.0000000
.00000000E-99	.00000000E-99	.00000000E-99
1.0000000	1.0000000	1.0000000
.66666666	.66666666	.66666667
1.0000000	1.0000000	1.0000000
4.6666666	4.6666666	4.6666667
.33333333	.33333334	.33333334

COLUNA 5

1.4166666	1.4166666	1.4166667
.20833333	.20833333	.20833334
.75000000	.75000000	.75000000
.50000000	.50000000	.50000000
.37499990	.37500000	.37500010
4.9166666	4.9166666	4.9166667
.25000000	.25000000	.25000000

COLUNA 6

1.0583333	1.0583333	1.0583333
.50000000E-01	.50000000E-01	.50000000E-01
.54166664	.54166664	.54166668
.32499998	.32499998	.32500002
.52499996	.52499996	.52499998
4.5333332	4.5333332	4.5333334
.21666662	.21666666	.21666670

COLUNA 7

.84305551	.84305551	.84305557
-.26388899E-01	-.26388890E-01	-.26388881E-01
.34861110	.34861110	.34861112
.18472221	.18472221	.18472222
.72916658	.72916664	.72916674
3.8583330	3.8583330	3.8583336
.16388888	.16388889	.16388891

COLUNA 8

.65535707	.65535708	.65535720
-.15873029E-02	-.15873014E-02	-.15872985E-02
.21408727	.21408728	.21408732
.11666665	.11666666	.11666667
.61349197	.61349200	.61349217
3.0793647	3.0793647	3.0793655
.97420600E-01	.97420620E-01	.97420670E-01

COLUNA 9

.46160707	.46160708	.46160722
.10615071E-01	.10615078E-01	.10615089E-01
.12256942	.12256943	.12256946
.81721220E-01	.81721221E-01	.81721239E-01
.37668636	.37668645	.37668667
2.3435760	2.3435760	2.3435768
.40848181E-01	.40848209E-01	.40848240E-01

COLUNA 10

.30225136	.30225137	.30225150
.32490024E-02	.32490071E-02	.32490117E-02
.63155834E-01	.63155855E-01	.63155888E-01
.52469126E-01	.52469127E-01	.52469146E-01
.00000000	.00000000	.00000000
.00000000	.00000000	.00000000

EXEMPLO 13

```
   .00000000         .00000000         .00000000
-----------------------------------------------------------
1 2 * 5 .000E-99 .000E-99 .000E-99
3 4 * 6 .000E-99 .000E-99 .000E-99
5 6 * 7 .000E-99 .000E-99 .000E-99
0 7 * 1 2.00     2.00     2.00          Descrição do Sistema
0 3 - 8 2.00     2.00     2.00
2 8 * 2 .000E-99 .000E-99 .000E-99
0 1 / 3 2.00     2.00     2.00
0 1 + 4 2.00     2.00     2.00
-----------------------------------------------------------
1.00     1.00     1.00
1.00     1.00     1.00
1.00     1.00     1.00                  Valores iniciais
1.00     1.00     1.00
-----------------------------------------------------------
COLUNA  1
     1.0000000         1.0000000         1.0000000
     1.0000000         1.0000000         1.0000000
     1.0000000         1.0000000         1.0000000
     1.0000000         1.0000000         1.0000000
     1.0000000         1.0000000         1.0000000
     1.0000000         1.0000000         1.0000000
     1.0000000         1.0000000         1.0000000
    -1.0000000        -1.0000000        -1.0000000
COLUNA  2
     2.0000000         2.0000000         2.0000000
    -1.0000000        -1.0000000        -1.0000000
     2.0000000         2.0000000         2.0000000
     1.0000000         1.0000000         1.0000000
     1.0000000         1.0000000         1.0000000
     3.0000000         3.0000000         3.0000000
     4.0000000         4.0000000         4.0000000
```

-2.0000000	-2.0000000	-2.0000000
COLUNA 3		
4.0000000	4.0000000	4.0000000
-.50000000	-.50000000	-.50000000
-1.0000000	-1.0000000	-1.0000000
1.0000000	1.0000000	1.0000000
1.5000000	1.5000000	1.5000000
2.0000000	2.0000000	2.0000000
6.5000000	6.5000000	6.5000000
1.0000000	1.0000000	1.0000000
COLUNA 4		
4.3333333	4.3333333	4.3333334
1.1666666	1.1666666	[illegible]66667
-.66666667	-.66666666	-.66666666
1.3333333	1.3333333	1.3333334
.49999990	.49999990	.50000010
1.6666666	1.6666666	1.6666668
8.6666665	8.6666665	8.6666669
.66666666	.66666666	.66666667
COLUNA 5		
4.3333332	4.3333332	4.3333333
-.12500001	-.12499998	-.12499998
.41666665	.41666667	.41666668
1.0833333	1.0833333	1.0833334
.20833290	.20833310	.20833350
2.4999998	2.4999998	2.5000003
8.8749990	8.8749992	8.8750009
-.41666668	-.41666667	-.41666665
COLUNA 6		
3.5499996	3.5499996	3.5500004

-.75833338	-.75833330	-.75833328
.46666664	.46666666	.46666676
.86666664	.86666664	.86666666
.70833260	.70833300	.70833420
1.9166663	1.9166664	1.9166671
9.2499971	9.2499982	9.2500025
-.46666676	-.46666666	-.46666664

COLUNA 7

3.0833323	3.0833326	3.0833342
.29861105	.29861108	.29861115
-.91666900E-01	-.91666600E-01	-.91666483E-01
.59166660	.59166660	.59166674
.70416460	.70416620	.70416860
1.1444438	1.1444443	1.1444451
10.890270	10.890274	10.890285
.91666483E-01	.91666600E-01	.91666900E-01

COLUNA 8

3.1115057	3.1115068	3.1115100
.37678561	.37678568	.37678583
-.41825449	-.41825397	-.41825344
.44047604	.44047608	.44047632
.70773370	.70773770	.70774220
.54722101	.54722193	.54722358
10.400777	10.400789	10.400809
.41825344	.41825397	.41825449

COLUNA 9

2.6001942	2.6001972	2.6002023
-.21773329	-.21773308	-.21773298
-.13963430	-.13963295	-.13963158
.38893821	.38893835	.38893875

EXEMPLO 14

-.88131000E-02	-.88049000E-02	-.87962000E-02
.52450101	.52450368	.52450698
8.9708242	8.9708532	8.9708907
.13963158	.13963295	.13963430

COLUNA 10

1.9935164	1.9935228	1.9935314
-.17328900	-.17328866	-.17328840
.41611213	.41611536	.41611887
.28891046	.28891080	.28891137
.00000000	.00000000	.00000000
.00000000	.00000000	.00000000
.00000000	.00000000	.00000000
.00000000	.00000000	.00000000

---

2 3 / 4 .000E-99 .000E-99 .000E-99

0 4 * 1 1.00 1.00 1.00 — Descrição do Sistema

0 3 * 2-1.00 -1.00 -1.00

0 2 * 3 1.00 1.00 1.00

.000E-99 .000E-99 .000E-99

.000E-99 .000E-99 .000E-99 — Valores Iniciais

1.00 1.00 1.00

COLUNA 1

.00000000E-99	.00000000E-99	.00000000E-99
.00000000E-99	.00000000E-99	.00000000E-99
1.0000000	1.0000000	1.0000000
.00000000E-99	.00000000E-99	.00000000E-99

COLUNA 2

.00000000E-99	.00000000E-99	.00000000E-99
-1.0000000	-1.0000000	-1.0000000
.00000000E-99	.00000000E-99	.00000000E-99
-1.0000000	-1.0000000	-1.0000000

COLUNA 3

-.50000000	-.50000000	-.50000000
.00000000E-99	.00000000E-99	.00000000E-99
-.50000000	-.50000000	-.50000000
.00000000E-99	.00000000E-99	.00000000E-99

COLUNA 4

.00000000E-99	.00000000E-99	.00000000E-99
.16666666	.16666666	.16666667
.00000000E-99	.00000000E-99	.00000000E-99
-.33333334	-.33333334	-.33333333

COLUNA 5

-.83333335E-01	-.83333335E-01	-.83333332E-01
.00000000E-99	.00000000E-99	.00000000E-99
.41666665E-01	.41666665E-01	.41666668E-01
.00000000E-99	.00000000E-99	.00000000E-99

COLUNA 6

.00000000E-99	.00000000E-99	.00000000E-99
-.83333336E-02	-.83333330E-02	-.83333330E-02
.00000000E-99	.00000000E-99	.00000000E-99
-.13333335	-.13333333	-.13333332

COLUNA 7

-.22222225E-01	-.22222221E-01	-.22222220E-01
.00000000E-99	.00000000E-99	.00000000E-99
-.13888890E-02	-.13888888E-02	-.13888888E-02
.00000000E-99	.00000000E-99	.00000000E-99

COLUNA 8

.00000000E-99	.00000000E-99	.00000000E-99
.19841268E-03	.19841268E-03	.19841272E-03
.00000000E-99	.00000000E-99	.00000000E-99
-.53968264E-01	-.53968252E-01	-.53968245E-01

COLUNA 9

-.67460330E-02	-.67460315E-02	-.67460306E-02
.00000000E-99	.00000000E-99	.00000000E-99
.24801585E-04	.24801585E-04	.24801590E-04
.00000000E-99	.00000000E-99	.00000000E-99

COLUNA 10

.00000000E-99	.00000000E-99	.00000000E-99
-.27557323E-05	-.27557316E-05	-.27557316E-05
.00000000E-99	.00000000E-99	.00000000E-99
.00000000	.00000000	.00000000

B I B L I O G R A F I A

1 - INTERVAL ANALYSIS
Ramon E.Moore
Prentice - Hall, Inc. - 1966

2 - TOPICS IN INTERVAL ANALYSIS
Edited by E.Hansen
Oxford University Press - 1969

3 - ERROR IN DIGITAL COMPUTATION
Edited by L.B.Rall
John Wiley & Sons, Inc. - 1964

4 - METODOS NUMERICOS - I - ALGEBRA LINEAR
Ivan de Q.Barros
John Wiley & Sons, Inc. - 1964

5 - THE THEORY OF MATRICES IN NUMERICAL ANALYSIS
A.S.Householder
Blaisdell Publishing Co. - 1964

6 - NUMERICAL MATHEMATICAL ANALYSIS
J.B.Scarborough
The Johns Hopkins Press - 1950

7 - ELEMENTOS DE ANALISE NUMERICA
Ivan de Queiroz Barros
7º Colóquio Brasileiro de Matemática