Matemática Universitária Nº 13, junho de 1991, 65–82

Aritmética de Ponto Flutuante e Análise de Erros

Oscar H. Bustos*

1. Introdução.

Junto a Ralston e Rabinowitz (1978) podemos dizer que a Análise Numérica é simultaneamente uma ciência e uma arte.

Enquanto ciência, a Análise Numérica tem a ver com os procedimentos para resolver certos problemas matemáticos usando operações aritméticas.

Ora, podem existir vários processos diferentes para resolver um mesmo problema. Qual é o mais conveniente para usar em um certo problema? Às vezes, a resposta a tal pergunta será obtida somente depois de resolver o problema. Outras vezes vai depender do conhecimento de certas propriedades das funções envolvidas, conhecimento que se poderia se obter seja em teoria ou por meio de cálculos numéricos. É neste ponto que a Análise Numérica converte-se em uma arte. Assim, podemos dizer que como arte, a Análise Numérica tem a ver com a escolha do "melhor" procedimento para resolver um determinado problema.

Sem dúvida, atingir um nível razoável de aperfeiçoamento tomará muito tempo e esforço, até conseguir o desenvolvimento da intuição.

2. Fontes de Erros.

Vamos agora analisar os erros que podemos cometer quando procuramos a solução numérica de um problema. Esses erros são chamados "erros computacionais".

*O autor agradece ao Prof. Jonas Miranda e ao Prof. Milton P. de Borba pela valiosa ajuda na tradução destas notas (escritas em espanhol) e também na correção de vários erros.

Segundo Ralston e Rabinowitz (1978) tais erros podem provir de três fontes principais:

(i) Erros grosseiros: entrada incorreta de dados (por exemplo, a colocação errada da vírgula decimal); o programa de computador não faz o que queremos, isto é, o programa pode estar efetuando corretamente operações que não desejamos.

(ii) Erros de truncamento: certos processos matemáticos requerem, por definição, um número infinito de operações, coisa inviável do ponto de vista do cálculo efetivo. Assim, devemos aproximar tais processos por um outro que requeira apenas um número finito de operações elementares (soma, subtração, multiplicação e/ou divisão). Os exemplos mais comuns de tais procedimentos são:

a) Cálculo de uma função $\exp(x)$, $\text{seno}(x)$, etc. por meio de um número finito de termos da sua expansão em série.
b) Aproximação de uma integral por uma soma finita, como na "regra do trapézio", por exemplo.
c) Solução de uma equação diferencial substituindo as derivadas por razões incrementais.
d) Solução de uma equação da forma $f(x) = 0$ por métodos iterativos como, por exemplo, o de Newton–Raphson que, em geral, convergem quando o número de iterações vai para infinito.

(iii) Erros de arredondamento: quando decidimos usar um certo computador para efetuar nossos cálculos, já estamos fixando o nosso "universo de números", que será sempre um conjunto finito e que será o único existente para efetuar todos os cálculos. Em conseqüência disso todos os outros números que não estejam nesse "universo" deverão ser representados (aproximados) por algum elemento desse conjunto. Essa aproximação chama-se "arredondamento" e a diferença entre o verdadeiro número e seu representante é conhecida sob nome de "erro de arredondamento".

3. Algumas formalizações em torno de erros.

DEFINIÇÃO 3.1: Sejam VV = valor verdadeiro, VA = valor aproximado ou calculado, E = erro; definidos por:

$$VV = VA + E. \tag{1}$$

Chama-se "erro relativo (ER) de VA com respeito a VV a

$$ER = E/VV. \tag{2}$$

EXEMPLO 3.1: Sejam $VV = 1/3$, $VA = 0.333$; então $E = (1/3) * 10^{-3}$,

Seja d um número inteiro maior do que 1. Pomos:

$$R = R(d,10) = \{y \in \mathbf{R} : y - [y] = a_1 10^{-1} + \ldots + a_d 10^{-d}, \quad a_i \in \{0,1,\ldots,9\}\}.$$

Como é costume, pomos

$$a_1 10^{-1} + \ldots + a_d 10^{-d} = 0.a_1 \ldots a_d = .a_1 \ldots a_d.$$

Por exemplo, $R(3,10)$ será o conjunto de todos os números reais com parte fracionária decimal de 3 dígitos.

PROPOSIÇÃO 3.1: Seja $x \in \mathbf{R}$.

(i) Se existir $x^* \in R(d,10)$ tal que $|x^* - x| < 0.5 * 10^{-d}$, então x^* é o único elemento em $R(d,10)$ que satisfaz $|x^* - x| \leq 0.5 * 10^{-d}$.

(ii) Se $x^* \in R(d,10)$ é tal que $|x^* - x| = 0.5 * 10^{-d}$, então $x^{**} = x^* + \text{sinal}(x - x^*)10^{-d}$ também satisfaz: $|x^{**} - x| = 0.5 * 10^{-d}$ e $x^{**} \in R(d,10)$.

DEFINIÇÃO 3.2: Na situação (i), o x^* é chamado "x arredondado a d casas decimais". Na situação (ii), se $x^* - [x^*] = .a_1 \ldots a_d$ e a_d é par, então x^* é o "x arredondado", caso contrário, x^{**} recebe esse nome. Com $x^{(d)}$ denotamos o "x arredondado a d casas decimais",.

EXEMPLO 3.2: Seja $d = 4$, então:

$$x = .43276 \Rightarrow x^{(d)} = .4328, \; x = .43274 \Rightarrow x^{(d)} = .4327,$$
$$x = .43275 \Rightarrow x^{(d)} = .4328, \; x = .43265 \Rightarrow x^{(d)} = .4326.$$

DEFINIÇÃO 3.3: Sejam x e y em $\mathbf{R}$, $k \geq 1$, k inteiro. Diz-se que a k-ésima casa decimal de y é significativa em relação a x (ou quando y é considerado como uma "aproximação" de x) se $|x - y| \leq 0.5 * 10^{-k}$.

EXEMPLO 3.3: Sejam $x = .2475110$, $y = .2474127$, então:

$$x - y = .0000983,$$

assim: $|x - y| = .0983 * 10^{-3} = .983 * 10^{-4}$, daí que a 3ª casa decimal de y é significativa mas a 4ª não é.

Por que não usar o conceito de quantidade de dígitos significativos como critério de aproximação? Vejamos: sejam x e y números reais e consideremos y como uma aproximação de x. Parece natural definir o número de dígitos significativos de y em relação a x como o número de casas decimais de y que são significativas em relação a x. Mas, vejamos o seguinte exemplo:

$$x_1 = .98632, \qquad x_2 = .00278$$
$$y_1 = .9863 \qquad y_2 = .0028.$$

Ora y_1 e y_2 podem ser considerados como aproximações de x_1 e x_2 respectivamente. Porém: são y_1 e y_2 igualmente significativos? Se os dois são os resultados finais de um certo cálculo e apenas estamos interessados no erro absoluto, então a resposta é "sim"; mas se são os passos intermediários de um cálculo que logo usará y_1 e y_2 como divisores, a resposta é "não"; porque a grandeza do erro em $1/y_1$ é bem menor do que a correspondente em $1/y_2$. Observemos que o erro relativo de y_1 em relação a x_1 é $2 * 10^{-5}$ enquanto o erro relativo de y_2 em relação a x_2 é maior que $7 * 10^{-3}$. Daí que se a soma ou diferença de dois números faz crescer o erro relativo, este resultado quando usado mais na frente como divisor, poderia ocasionar um acréscimo importante do erro. Este fenômeno é chamado "supressão cancelativa", e é um dos mais importantes para ser levado em consideração.

4. Aritmética de ponto fixo.

DEFINIÇÃO 4.1: Sejam $\beta \geq 2$, e $t \geq 2$ números inteiros. Chama-se "conjunto de números reais no sistema de ponto fixo caracterizado por β e t" ao conjunto $F = F(\beta, t)$ de todos os números da forma

$$x = \pm(d_1/\beta + d_2/\beta^2 + \ldots + d_t/\beta^t),$$

onde $d_1, \ldots, d_t$ são inteiros tais que:

$$0 \leq d_i \leq \beta - 1 \ \forall i = 1, \ldots, t.$$

Alguns computadores trabalham com este sistema acrescido do número $+1$ ou -1. Notemos que todos esses x satisfazem

$$0 \leq |x| \leq 1.$$

Outros computadores existem, porém que somente trabalham com inteiros, isto é, os x anteriores multiplicados por β^t.

A Aritmética de ponto fixo (quer dizer, a realização de operações entre números em ponto fixo e representação dos resultados como se tivéssemos somente o conjunto F) embora seja potencialmente mais precisa que a de ponto flutuante, requer uma análise bem mais cuidadosa e o fluxo natural dos cálculos é mais complicado porque fatores de escala devem ser levados em conta. Não falaremos mais sobre este sistema. Quem se interessar por mais detalhes sobre os principais pontos da aritmética de ponto fixo poderá consultar, por exemplo, WILKINSON (1963).

5. Aritmética de ponto flutuante.

DEFINIÇÃO 5.1: Sejam $\beta \geq 2, t \geq 2, U \geq 1, L \leq -1$ números inteiros. Chama-se "conjunto de números reais no sistema de ponto flutuante caracterizado por β, t, U e L" ao conjunto $F = F(\beta, t, U, L)$ de todos os números x da forma

$$x = \pm(d_1/\beta + d_2/\beta^2 + \ldots + d_t/\beta^t) * \beta^e,$$

onde $d_1, \ldots, d_t$ são inteiros tais que:

$$0 \leq d_i \leq \beta - 1, \quad \forall i = 1, \ldots, t.$$

e é outro inteiro tal que $L \leq e \leq U$.

O sistema F diz-se "normalizado" se $\forall x$ vale que $d_1 \neq 0$.

O inteiro "e" chama-se "expoente" e o número

$$f = d_1/\beta + d_2/\beta^2 + \ldots + d_t/\beta^t$$

chama-se "parte fracionária".

Alguns autores chamam "mantissa" ao número

$$m = \pm f.$$

É fácil ver que o sistema está normalizado se e somente se

$$1/\beta \leq |m| < 1.$$

EXEMPLO 5.1: O sistema $F(2,3,2,-1)$ é

$$
\begin{array}{l}
+1/4,\ +(1/4+1/16),\ +(1/4+1/8),\ +(1/4+1/8+1/16),\\
+1/2,\ +(1/2+1/8),\ +(1/2+1/4),\ +(1/2+1/4+1/8),\\
+1,\ +(1+1/4),\ +(1+1/2),\ +(1+1/2+1/4),\\
+2,\ +(2+1/2),\ +(2+1),\ +(2+1+1/2),\\
-1/4,\ -(1/4)+1/16),\ -(1/4+1/8),\ -(1/4+1/8+1/16),\\
-1/2,\ -(1/2+1/8),\ -(1/2+1/4),\ -(1/2+1/4+1/8),\\
-1,\ -(1+1/4),\ -(1+1/2),\ -(1+1/2+1/4),\\
-2,\ -(2+1/2),\ -(2+1),\ -(2+1+1/2),
\end{array}
$$

DEFINIÇÃO 5.2: Seja $F = F(\beta,t,U,L)$. Diz-se que um número real x está na imagem de F se

$$y^- \leq |x| \leq y^+$$

onde $y^+ = \max F$ e $y^- = \min\{y \in F : y > 0\}$.

EXEMPLO 5.2: (Continuação 5.1). Se $F = F(2,3,2,-1)$ então: $y^- = 1/4$ e $y^+ = 7/2$, daí que imagem de F é $[-7/2,-1/4] \cup [1/4,7/2]$.

DEFINIÇÃO 5.3: Seja x na imagem de F, denota-se por $fl(x)$ um número em F tal que:

$$|x - fl(x)| = \min\{|x-y| : y \in F\}.$$

Pode-se ver facilmente que

$$|x - fl(x)| / |x| \leq 0.5 * \beta^{1-t}.$$

Neste caso poremos: $\Delta x = fl(x) - x$.

EXEMPLO 5.3: (Continuação do Exemplo 5.2). Neste caso temos:

$$fl(3.1) = 3\ fl(3.25) = 3 \text{ ou } fl(3.25) - 3.5,\ fl(3.3) = 3.5.$$

Operações elementares em F

Sejam:

$$D+^* = \{(x,y) \in F \times F : x + y \text{ está na imagem de } F\},$$
$$D-^* = \{(x,y) \in F \times F : x - y \text{ está na imagem de } F\},$$
$$D*/^* = \{(x,y) \in F \times F : x * y \text{ está na imagem de } F\},$$
$$D/^* = \{(x,y) \in F \times F : x/y \text{ está na imagem de } F\}.$$

Chama-se "soma em ponto flutuante" à $+^*: D+^* \to F$ dada por

$$+^*(x,y) := x +^* y := fl(x+y).$$

Chama-se "diferença em ponto flutuante" à $-^*: D-^* \to F$ dada por

$$-^*(x,y) := x -^* y := fl(x-y).$$

Chama-se "produto em ponto flutuante" à $*^*: D*^* \to F$ dada por

$$*^*(x,y) := x *^* y := fl(x*y).$$

Chama-se "divisão em ponto flutuante" à $/^*: D/^* \to F$ dada por

$$/^*(x,y) := x /^* y := fl(x/y).$$

Daqui em diante vamos usar o símbolo *op* e *op** para indicar qualquer uma das operações acima. E *Dop* para indicar o correspondente domínio.

DEFINIÇÃO 5.4: Sejam x e y na imagem de F. Chama-se erro de arredondamento da operação *op* em (x,y) ao:

$$\Delta op(x,y) = (fl(x) op^* fl(y)) - (x\, op\, y).$$

Sem entrar em muito detalhe pode-se dizer que um limitante superior da certeza relativa da aritmética de ponto flutuante de um sistema computacional com sistema de ponto flutuante dado por $F = F(\beta,t,U,L)$ é dada por β^{1-t}; quer dizer:

$$|\Delta op(*x,y)/op(x,y)| \leq \beta^{1-t}.$$

DEFINIÇÃO 5.5: Sejam x e y na imagem de F.

Se $|x\ op\ y| > y^+$, diz-se que o intento de realizar $x\ op\ y$ dá um erro de "overflow".
Se $|x\ op\ y| < y^-$, diz-se que o intento de realizar $x\ op\ y$ dá um erro de "underflow".

EXEMPLO 5.4: (Continuação do Exemplo 5.3)

$$x = 5/2,\ y = 3 \quad \Rightarrow \quad x + y \text{ dá "overflow".}$$
$$x = 1/2,\ y = 1/4 \Rightarrow x * y \text{ dá "underflow".}$$

Análise de erro levando em consideração a possibilidade de erros de "overflow" ou "underflow" é muito complicada e entediante. Além disso para cada cálculo deve-se efetuar uma análise diferente. Existem técnicas que resolvem mais ou menos satisfatoriamente certas dificuldades e muitos algoritmos foram desenhados para tratar com este tipo de erros. Nestas notas não abordaremos esta questão e daí vamos assumir que não se apresentarão problemas de "overflow" e/ou "underflow". Isto é equivalente a supor que $L = -\infty$ e $U = +\infty$. Quem desejar mais informação sobre este tema pode ler o excelente livro de WILKINSON (1963).

DEFINIÇÃO 5.6: Chama-se épsilon do computador ao número dado por

$$\varepsilon_{\text{mach}} = \inf\{\varepsilon \in F : 1 +^* \varepsilon > 1\}.$$

Sendo F o sistema de ponto flutuante usado pelo sistema computacional.

Notemos que este número não é (em geral) y^-. No sistema que viemos analisando nos exemplos anteriores, ambos os valores coincidem. Porém não acontece a mesma coisa em geral. Pode-se provar que:

$$0.5 * \beta^{1-t} \leq \varepsilon_{mach} \leq \beta^{1-t}.$$

Mesmo que o $\varepsilon_{\text{mach}}$ esteja bem definido pela fórmula acima, a maioria dos programas usam como "épsilon do computador" um número, digamos eps, que difere do anterior num certo fator "potência de 2", mas tal parece que essa diferença não causa graves dificuldades nas aplicações. Um método para calcular eps poderia

usar um programa em FORTRAN que incluisse o seguinte trecho:

$$
\begin{aligned}
&EPS = 1. \\
10&EPS = 0.5 * EPS \\
&EPSP1 = EPS + 1. \\
&IF(EPSP1.GT.1.)GOTO10.
\end{aligned}
$$

Suposição básica para análise de erros (nestas notas).

Deixando de lado, como já dissemos, a possibilidade de erros de "overflow" e/o "underflow" e certos detalhes quanto ao "hardware" que efetivamente implementa a aritmética de ponto flutuante num determinado computador vamos assumir que se o nosso sistema de ponto flutuante é dado por $F = F(\beta, t, +\infty, -\infty)$, então:

$$
\begin{aligned}
x \in F,\ y \in F \qquad & x\ op^*\ y = (x\ op\ y)(1 + \varepsilon(x\ op\ y)), \\
\text{com} \quad & \varepsilon(x\ op\ y) \leq eps.
\end{aligned} \tag{3}
$$

Para denotar os resultados de operações em ponto flutuante tem sido adotada uma certa notação que é conveniente mas pouco precisa. Nós seguiremos este uso. Se fica claro do contexto como calcular uma certa expressão aritmética E, então $fl(E)$ denota o valor dessa expressão obtida usando aritmética de ponto flutuante de nosso sistema. Por exemplo:

$$
\begin{aligned}
fl(x + y) &:= fl(x) +^* fl(y) \\
fl * x + (y + z)) &:= fl(x) +^* fl(y + z) \\
&= fl(x) +^* (fl(y) +^* fl(z)), \\
fl((x + y) + z) &:= fl(x + y) +^* fl(z).
\end{aligned}
$$

Se o nosso sistema computacional tem já embutidas certas funções como seno, coseno, raiz quadrada, etc. usamos a notação $\sin^*(x), \cos^*(x), \sqrt{}^*x$, etc. para denotar $fl(\sin(x))$, $fl(\cos(x)), fl(\sqrt{x})$, etc., que são as aproximações calculadas pelo computador dos valores $\sin(x), \cos(x), \sqrt{x}$, etc.

6. Propagação dos erros.

Uma mesma quantidade pode-se calcular de diversas maneiras, isto é através de diferentes algoritmos.

EXEMPLO 6.1: Seja $\phi: \mathbf{R}^3 \to \mathbf{R}$ definida por:

$$\phi((a,b,c)') = a + b + c.$$

Algoritmo 1:

- $u_1 := a + b, \quad u_2 := c$
- $v := u_1 + u_2$

Algoritmo 2:

- $u_1 := a, \quad u_2 := b + c$
- $v := u_1 + u_2$

EXEMPLO 6.2: Seja $\phi: \mathbf{R}^2 \to \mathbf{R}$ definida por:

$$\phi((a,b)') = a^2 - b^2$$

Algoritmo 1:

- $u_1 := a^2, \quad u_2 := b$
- $v_1 := u_1, \quad v_2 := u_2^2$
- $w := v_1 - v_2$

Algoritmo 2:

- $u_1 := a + b, \quad u_2 := a - b$
- $v := u_1 u_2$

Segue-se que devemos ter um critério para escolher o algoritmo mais adequado para ser usado em cada situação particular. Para elaborar um tal critério vamos examinar brevemente como poderia-se analisar propagação de erros (principalmente dos erros de arredondamento) na execução de um certo algoritmo.

Vejamos o que se passa no Exemplo 6.1. Aí se usarmos o algoritmo 1 obtemos

$$\varepsilon_y = \text{erro de } \tilde{y} \text{ relativo a } y = (\tilde{y} - y)/y \cong \frac{a+b}{a+b+c}\varepsilon_1 + 1\varepsilon_2,$$

se usarmos o algoritmo 2 obtemos

$$(\tilde{y} - y)/y \cong \frac{b+c}{a+b+c}\varepsilon_1 + 1\varepsilon_2$$

com $|\varepsilon_i| \leq eps$. Os fatores que magnificam os erros são $\frac{a+b}{a+b+c}$, $\frac{b+c}{a+b+c}$ e 1.
Logo, os valores $\frac{a+b}{a+b+c}$ e $\frac{b+c}{a+b+c}$ são críticos. Com efeito, suponhamos:

$$a = .23371258 * 10^{-4},$$
$$b = .33678429 * 10^{2} \quad \text{e}$$
$$c = -.33677811 * 10^{2}.$$

Então:

$$\frac{a+b}{a+b+c} = 0.33... * 10^{2}/(0.64... * 10^{-3}) \cong 0.5 * 10^{5}, \text{e}$$
$$\frac{b+c}{a+b+c} = 0.618... * 10^{-3}/(0.64... * 10^{-3}) \cong 0.97.$$

O método que usamos para analisar a propagação dos erros de arredondamento: começar supondo que cada dado tem associado um certo erro e seguir passo a passo um algoritmo calculando em cada etapa o erro de arrendondamento que se vai obtendo e descartando os termos com erro de ordem grande, é chamado "forward error asnalysis". Assim, usando esta técnica, podemos calcular os erros absolutos e/ou relativos para cada operação e usá-los nas operações a seguir até chegar ao valor final y.

Porém este método apresenta dois defeitos importantes: 1) a maioria das vezes obtém-se cotas muito grandes; 2) esta análise é freqüentemente muito tediosa e complicada. De qualquer maneira pode-se aplicar a todos os cálculos e obter resultados de utilidade.

Uma outra forma de efetuar análise de erros, chamada "backward error analysis", é também usada com bastante freqüência. Essencialmente, consiste na determinação do conjunto de valores iniciais de dados que poderiam ter conduzido ao mesmo resultado final. Por exemplo: suponhamos que os dados iniciais foram x_1, x_2 e x_3 e o resultado do cálculo: $x_1 + x_2 + x_3$ foi y, o "backward error analysis" consiste na determinação de cotas para os erros de entrada dos x_i's, digamos e_i's, tais que

$$y = (x_1 + e_1) + (x_2 + e_2) * (x_3 + e_3).$$

Fórmulas aproximadas dos erros absolutos e relativos.

DEFINIÇÃO 6.3: Sejam z e $\tilde{z}$ dois números ou vetores. Chama-se "erro absoluto da aproximação de z por $\tilde{z}$" ao

$$\Delta(\tilde{z}, z) := \tilde{z} - z.$$

"Erro relativo da aproximação de z por $\tilde{z}$ ao

$$\varepsilon(\tilde{z}, z) := (\tilde{z} - z)/z, \quad \text{se} \quad z \neq 0.$$

Suponhamos que D é um subconjunto aberto do $\mathbf{R}^n$ e que ϕ é uma função definnida sobre D com valores em $\mathbf{R}^m$ continuamente diferenciável sobre D. Seja $x \in D$ e $\tilde{x} \in D$ um valor "aproximado" de $x, \tilde{y} = \phi(\tilde{x})$, definimos:

$$\tilde{y}_i = \phi_i(\tilde{x}), \quad \forall\, i = 1, \ldots, m.$$

onde $\phi_1, \ldots, \phi_m$ são as componentes da ϕ. Então, pode-se verificar expandindo em série de Taylor e descartando os termos com erro de ordem alto, que:

$$\Delta(\tilde{y}_1, y_1) \cong \sum_{j=1}^{n} \partial_k \phi_1(x) \Delta(\tilde{x}_j, x_j)$$

$$\Delta(\tilde{y}, y) \cong D\phi(x) \Delta(\tilde{x}, x)$$

onde ∂_j é o operador "derivada parcial com respeito a j-ésima variável" e $D\phi$ é a matriz Jacobiana da ϕ. Daí:

$$\varepsilon(\tilde{y}_i, y_i) \cong \sum_{j=1}^{n} (x_j/\phi_i(x)) \partial_j \phi_i(x) \varepsilon(\tilde{x}_j, x_j).$$

O fator $\partial_j \phi_i(x)$ mede a sensibilidade com que y_i reage à perturbação $\Delta(\tilde{x}_j, x_j)$. Analogamente, o fator $((x_j/\phi_i(x))\partial_j \phi_i(x))$ indica quanto o erro relativo $\varepsilon(\tilde{x}_j, x_j)$ afeta o erro relativo $\varepsilon(\tilde{y}_i, y_i)$. Estes fatores de amplificação dos erros relativos têm a vantagem de não depender das escalas dos x's nem dos y's. São chamados também "números de condição". Se algum cálculo apresenta números de condição "grandes", diz-se que o tal problema de

cálculo está "mal condicionado", caso contrário, diz-se que o problema está "bem condicionado".

Problemas bem e mal condicionados

Numa linguagem não tão precisa, pode-se dizer que um problema está mal condicionado, quando pequenas perturbações nos dados de entrada ocasionam perturbações grandes nos resultados.

EXEMPLO 6.3: Problema mal condicionado.
Suponhamos querer encontrar os zeros de

$$f(z) = 0,$$

onde

$$f(z) = (z+1) * (z+2) * \ldots * (z+20).$$

É óbvio que esses zeros são: $-1, -2, \ldots, -20$.
Agora modificamos ligeiramente os dados de entrada. Suponhamos querer encontrar os zeros de

$$F(z) = 0,$$

onde

$$F(z) = f(z) + 2^{-23} z^{19}.$$

Pode-se ver que esta última equação tem 5 pares de zeros complexos (um deles é $-19.502 \pm 1.940i$) e entre os 10 zeros reais está o número 20.847.... Vemos então que, uma mudança da ordem de 10^{-7} em um dos coeficientes modificou totalmente a solução.

A experiência diz que o cálculo de zeros de polinômios de grau elevado constitui um exemplo típico de problema mal condicionado.

EXEMPLO 6.4: Problema bem condicionado.
Suponhamos querer resolver a equação diferencial:

$$y' = -y,$$

sua condição inicial:

$$y(0) = 1.$$

A solução é $y = e^{-x}$ e constitui um problema bem condicionado no sentido de que se trocamos $y(0) = 1$ por $y(0) = 1 + \varepsilon$, então a solução mudará de $y = e^{-x}$ para $y = (1+\varepsilon)e^{-x}$.

EXEMPLO 6.5: Continuação do Exemplo 6.1. Seja

$$y = \phi((a,b,c)') = a + b + c.$$

Então

$$\varepsilon(\tilde{y}, y) = \frac{a}{a+b+c}\,\varepsilon(\tilde{a}, a) + \frac{a}{a+b+c}\,\varepsilon(\tilde{b}, b) + \frac{c}{a+b+c}\,\varepsilon(\tilde{c}, c).$$

Daí que o problema vai estar bem condicionado se cada somando é pequeno quando comparado com a soma $a + b + c$.

EXEMPLO 6.6: Seja

$$y = \phi((p,q)') = -p + (p^2 + q)^{1/2}.$$

Então:

$$\begin{aligned}\varepsilon(\tilde{y}, y) = {} & (-p/(p^2+q)^{1/2})\,\varepsilon(\tilde{p}, p) \\ & + (p + (p^2+q)^{1/2})/(2(p^2+q)^{1/2})\,\varepsilon(\tilde{q}, q).\end{aligned}$$

Agora: $\left|-p/(p^2+q)^{1/2}\right| \leq 1$ e $\left|(p + (p^2+q)^{1/2})/(2(p^2+q)^{1/2})\right| \leq 1$ quando $q > 0$, logo, o problema estará bem condicionado se $q > 0$ e mal condicionado se $q \approx -p^2$.

Erros relativos nas operações básicas.

PROPOSIÇÃO 6.1: Vale que: (x e y são não nulos)

(i) $z = x * y \Rightarrow \varepsilon(\tilde{z}, z) \cong \varepsilon(\tilde{x}, x) + \varepsilon(\tilde{y}, y)$.
(ii) $z = x/y \Rightarrow \varepsilon(\tilde{z}, z) \cong \varepsilon(\tilde{x}, x) - \varepsilon(\tilde{y}, y)$.
(iii) $z = x \pm y \Rightarrow \varepsilon(\tilde{z}, z) \cong \frac{x}{x \pm y}\,\varepsilon(\tilde{x}, x) \pm \frac{y}{x \pm y}\,\varepsilon(\tilde{y}, y)$ se $x \pm y \neq 0$.
(iv) $z = \sqrt{x} \Rightarrow \varepsilon(\tilde{z}, z) \cong 0.5 * \varepsilon(\tilde{x}, x)$.

DEMONSTRAÇÃO: Fácil.

Desta proposição segue-se que as operações de multiplicação, divisão e raiz quadrada não são perigosas. Mas não acontece o mesmo com a soma, pois vai depender do valor de $x \pm y$.

7. Aritmética de Intervalos - Estimação Estatística do Erro de Arredondamento.

A técnica conhecida com o nome de "Aritmética de Intervalos" fornece uma maneira de determinar cotas entre as quais

pode-se assegurar que está o erro absoluto dum algoritmo, levando em consideração os erros de arredondamento e de entrada nos dados. Essa técnica baseia-se no fato de que quase todos os números reais que entram num algoritmo, tanto como dados de entrada ou como resultados intermediários, são impossíveis de ser representados exatamente num computador. O que melhor se pode saber é que para cada $a \in \mathbf{R}$, existem dois números consecutivos, digamos a' e a'' representáveis exatamente no computador tais que $a \in [a', a'']$. Daí que a técnica consiste na realização dos cálculos em termos desses intervalos em vez de fazê-los em termos de um número só. As operações aritméticas entre intervalos, digamos, $\circ \in \{\oplus, \ominus, \otimes, \oslash\}$, são definidas de forma tal que sejam consistentes com a interpretação acima. Isto é, se $\tilde{a}$ e $\tilde{b}$ são intervalos, então: $\tilde{c} = \tilde{a} \circ \tilde{b}$ é definido como um intervalo (o menor possível) tal que

$$\tilde{c} \supseteq \{a \circ b : a \in \tilde{a} \text{ e } b \in \tilde{b}\},$$

e com extremos representáveis exatamente no computador.

Substituindo todo número por um intervalo e toda operação aritmética por sua correspondente operação entre intervalos, obtemos algoritmos de cálculo sobre intervalos que vão produzir, como resultados, intervalos que possuam a solução exata.

Porém, uma manipulação pouco cuidadosa desta técnica pode conduzir a cotas de erros que sejam confiáveis mas muito pessimistas e daí de pouco utilidade prática. De fato, não é suficiente substituir num certo algoritmo às operações numéricas por operações entre intervalos sem considerar a forma particular em que se introduzem os erros de arredondamento e/ou entrada de dados num determinado resultado. Assim por exemplo, pode acontecer que um certo erro de arredondamento afete severamente os resultados intermediários e contudo, o resultado final não seja muito afetado.

Vamos ver um exemplo prático dessa situação. Os detalhes podem-se estudar em STOER e BULIRSCH (1980).

EXEMPLO 7.1: Calcular $y = \phi(x) = x^3 - 3x^2 + 3x$. Primeiramente vamos usar o chamado esquema de Horner, isto é usamos a identidade

$$x^3 - 3x^2 + 3x = ((x - 3)x + 3)x.$$

O algoritmo é:

$$u := x - 3,$$
$$v := u * x,$$
$$w := v + 3,$$
$$y := w * x.$$

Suponhamos agora que do valor x apenas sabe-se que

$$x \in \tilde{x} = [0.9,\ 1.1].$$

Se no algoritmo substituimos simplesmente os valores numéricos por intervalos e as operações pelas correspondentes operações entre intervalos, obtemos como resultado final o intervalo

$$\tilde{y} = [0.621,\ 1.419].$$

Mas este intervalo é bem maior que o intervalo que realmente da conta do efeito do possível erro em x, isto é:

$$\{\phi(x) : x \in \tilde{x}\} = [0.999,\ 1.001].$$

Suponhamos agora que o nosso sistema de ponto flutuante tem apenas 2 dígitos. Então, aplicando o algoritmo separadamente a $x = 0.9$ e $x = 1.1$, obtemos:

$$\text{para} \quad x = 0.9,\ y = 0.99, \quad \text{e para} \quad x = 1.1,\ y = 0.99.$$

Neste exemplo vemos que obtemos uma resposta mais útil sem usar aritmética de intervalos.

Muitas vezes, antes de começar a fazer contas, convém arrumar as coisas de forma tal que sugerem algoritmos mais adequados. Vejamos:

$$y = \phi(x) = x^3 - 3x^2 + 3x = 1 + (x-1)^3.$$

O algoritmo é:

$$u := x - 1,$$
$$v := u * u,$$
$$w := v * u,$$
$$y := w + 1.$$

Aplicando agora a este esquema a técnica da aritmética de intervalos, obtemos a partir do $x \in \tilde{x} = [0.9, 1.1]$, o intervalo $\tilde{y} = [0.999, 1.001]$.

A moral do exemplo é que para aplicar com sucesso a técnica de intervalos é necessário desenvolver novos algoritmos que produzam os mesmos resultados finais mas elaborando um esquema de tratamento dos erros nos passos intermediários bem melhorado.

Para acabar estas notas sobre análise de erros computacionais, vamos ver apenas muito superficialmente uma outra técnica chamada de "análise estatística (ou probabilística) de erros". Não entraremos a estudar detalhes formais. Para tanto seria necessário assumir que se conhecem os elementos matemáticos da Teoria de Probabilidades, o que não é o nosso caso. Quem desejar um estudo apenas um pouco mais detalhado deste assunto, com uma razoável introdução à matemática das probabilidades, pode consultar, por exemplo o livro de PIZER (1975).

A Teoria de Probabilidades é a parte da Matemática que é desenvolvida para analisar situações nas quais o aleatório tem um papel importante. Quer dizer, o resultado dum certo experimento ou fenômeno não pode ser conhecido exatamente. Por exemplo, o erro nos dados de entrada num computador devido a que o computador trabalha apenas com um conjunto finito de números, o erro final no cálculo duma certa $y = \phi(x)$ quando usamos um certo algoritmo. Obviamente, a maioria dos fenômenos da vida real tem resultados predizíveis mais ou menos aproximadamente. É claro que a Teoria de Probabilidades é aplicada também em muitos casos onde uma maior informação poderia ser coletada para obter resultados exatos ou mais aproximados aos verdadeiros. Por exemplo, se estudarmos com todo detalhe o hardware dum determinado computador, e como os números estão nele representados, e como são realmente efetuadas as diversas operações aritméticas, e como ele transmite os resultados para nosso conhecimento, e como... quiça possamos saber o que realmente fez nosso computador quando solicitamos-lhe calcular $y = \phi(x)$. Porém, ao usar probabilidades estamos supondo que essa informação toda não está disponível ou que o seu manejo é muito complicado como para ser de utilidade prática, dado que queremos uma resposta a nosso problema num tempo razoável. Neste caso justifica-se bem o emprego da Teoria de Probabilidades para melhor entender o problema sob

estudo.

Pois bem. Uma das formas de usar Teoria de Probabilidades ou Estatística para analisar erros de computação, seria supor que em vez de trabalhar com números exatos, estamos trabalhando com certos conjuntos de números (variáveis aleatórias) que tem associada uma certa distribuição sobre a reta (ou sobre um espaço vetorial caso estejamos trablhando com vetores). Daí aplicar o que essa Teoria ensina sobre como conhecer certos parâmetros das distribuições das variáveis obtidas através de transformações de outras variáveis. Dessa forma poder-se-ia obter um certo conhecimento da distribuição do erro total num cálculo, e daí uma medida da bondade do algoritmo usado.

Bibliografia

1. Pizer, Stephen, M. (1975), "Numerical Computing and Mathematical Analysis," Science Research Associates, Inc., Chicago.
2. Ralston, Anthony and RABINOWITZ, Philip (1978), "A First Course in Numerical Analysis," 2nd. ed., McGraw Hill, New York.
3. Stoer, Josef, and Bulirsch, Roland (1980), "Introduction to Numerical Analysis," Springer-Verlag, New York.
4. Wilkinson, J.H. (1963), "Rounding Erros in Algebraic Processes," Englewood Cliffs, Prentice-Hall, N.J..

IMPA
Estrada Dona Castorina, 110
22460 Rio de Janeiro, RJ