

Capítulo 4.

ESTIMAÇÃO DE ERROS NUMÉRICOS

Conforme visto no capítulo 1, três tipos de métodos podem ser empregados na solução de um problema: experimentais, analíticos e numéricos. Os erros envolvidos nestes métodos, mostrados na Fig. 4.1, são:

- nos resultados experimentais: erros experimentais;
- nas soluções analíticas e numéricas: erros de modelagem; e
- nas soluções numéricas: erros numéricos.

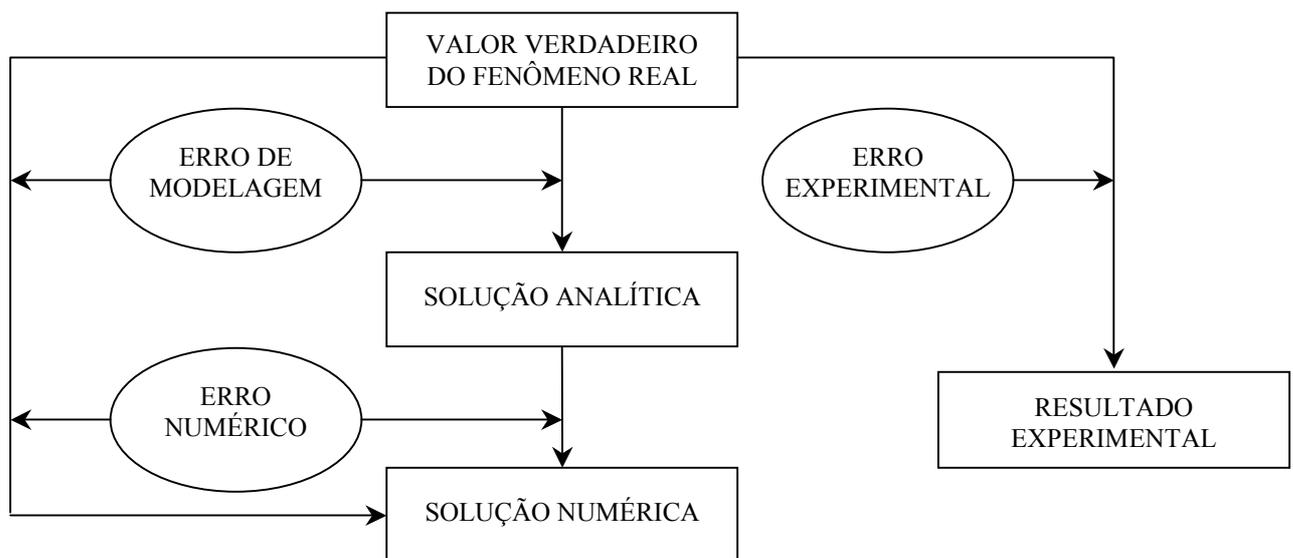


Figura 4.1 Erros envolvidos nos métodos da engenharia.

Erro experimental é a diferença entre o valor verdadeiro de uma variável de interesse e o seu resultado experimental (ABNT, 1997).

Erro de modelagem é a diferença entre o valor verdadeiro de uma variável de interesse e a sua solução analítica exata (Ferziger & Peric, 1999). Ele é causado pelas simplificações feitas sobre o fenômeno real na concepção do modelo matemático. Os erros de modelagem afetam

tanto as soluções analíticas quanto as numéricas porque ambas se baseiam em modelos matemáticos.

Erro numérico (E) é a diferença entre a solução analítica exata (Φ) de uma variável de interesse e a sua solução numérica (ϕ), isto é, (Ferziger & Peric, 1999)

$$E(\phi) = \Phi - \phi \quad (4.1)$$

A solução numérica ideal é igual à solução analítica exata do problema, ou seja, é aquela em que o erro numérico é nulo.

Na prática, o valor verdadeiro é desconhecido e, portanto, consegue-se apenas estimar o valor do erro experimental ou do erro de modelagem. Este valor estimado é denominado de incerteza (U). A mesma situação ocorre com relação ao erro numérico. Normalmente, em situações práticas uma solução numérica é obtida porque a solução analítica é desconhecida. Assim, o erro numérico também tem que ser estimado. Os processos usados para estimar cada tipo de erro da Fig. 4.1 são:

- análise de incerteza (ABNT, 1997), para erros experimentais;
- validação (AIAA, 1998), para erros de modelagem; e
- verificação (AIAA, 1998), para erros numéricos.

Este capítulo é dedicado ao processo de verificação de soluções numéricas. Em outras palavras, como estimar o erro numérico ou calcular a incerteza numérica, que é definida por

$$U(\phi) = \phi_{\infty} - \phi \quad (4.2)$$

onde

ϕ = solução numérica

ϕ_{∞} = solução analítica estimada

$U(\phi)$ = incerteza numérica de ϕ

A incerteza de uma solução numérica é calculada com os chamados estimadores de erro.

A magnitude aceitável para o erro numérico depende:

- da finalidade da solução numérica (exercício acadêmico, pré-projeto, projeto);
- dos recursos financeiros disponíveis;
- do tempo disponível para realizar as simulações; e
- dos recursos computacionais (memória e velocidade de processamento) disponíveis.

Importância de se estimar o erro numérico:

- $E >$ aceitável: compromete-se a confiabilidade do uso da solução numérica;

- $E <$ necessário: há desperdício de recursos;
- para validar e desenvolver modelos matemáticos; por exemplo, escoamentos turbulentos; neste caso o erro numérico deve ser muito menor de que o erro de modelagem; e
- para otimizar o uso da malha através da homogeneização do erro, diminuindo as necessidades computacionais.

O erro numérico é causado por quatro fontes (Ferziger & Peric, 1999):

- 1) erros de truncamento;
- 2) erros de iteração;
- 3) erros de arredondamento; e
- 4) erros de programação.

4.1 ERROS DE TRUNCAMENTO

Os erros de truncamento resultam das aproximações numéricas feitas na discretização de um modelo matemático (Ferziger & Peric, 1999; Tannehill *et al.*, 1997; Roache, 1998). Quando o erro numérico é causado apenas por erros de truncamento, admite-se que (Ferziger & Peric, 1999; Roache, 1998):

$$E(\phi) = C_1 h^{p_1} + C_2 h^{p_2} + C_3 h^{p_3} + \dots \quad (4.3)$$

(equação geral do erro de discretização)

onde

C_1, C_2, C_3, \dots = coeficientes que independem de h

p_1, p_2, p_3, \dots = ordens verdadeiras de $E(\phi)$; números inteiros positivos

p_L = ordem assintótica de $E(\phi)$; $p_L \geq 1$; inclinação da curva do erro

num gráfico $\log(E)$ versus $\log(h)$ para $h \rightarrow 0$

ϕ = variável de interesse

h = tamanho dos elementos da malha

Neste caso, o erro numérico também é denominado de erro de discretização. Existe uma nítida correspondência entre as Eqs. (4.3) e (2.44). Há dois tipos de estimativas do erro de discretização: *a priori* e *a posteriori* das soluções numéricas.

4.1.1 Estimativas *a Priori*

No caso de $h \rightarrow 0$, a Eq. (4.3) se reduz a

$$E(\phi) = C_1 h^{p_L} \quad (4.4)$$

Assim, antes de se obter qualquer solução numérica, é possível prever o comportamento assintótico do erro de discretização com relação a h e p_L , conforme esquematizado na Fig. 4.2. Não é possível obter o valor de $E(\phi)$ porque C_1 é desconhecido. Tanto C_1 quanto p_L dependem das aproximações numéricas empregadas. Da Eq. (4.4), $E(\phi) \rightarrow 0$ para $h \rightarrow 0$.

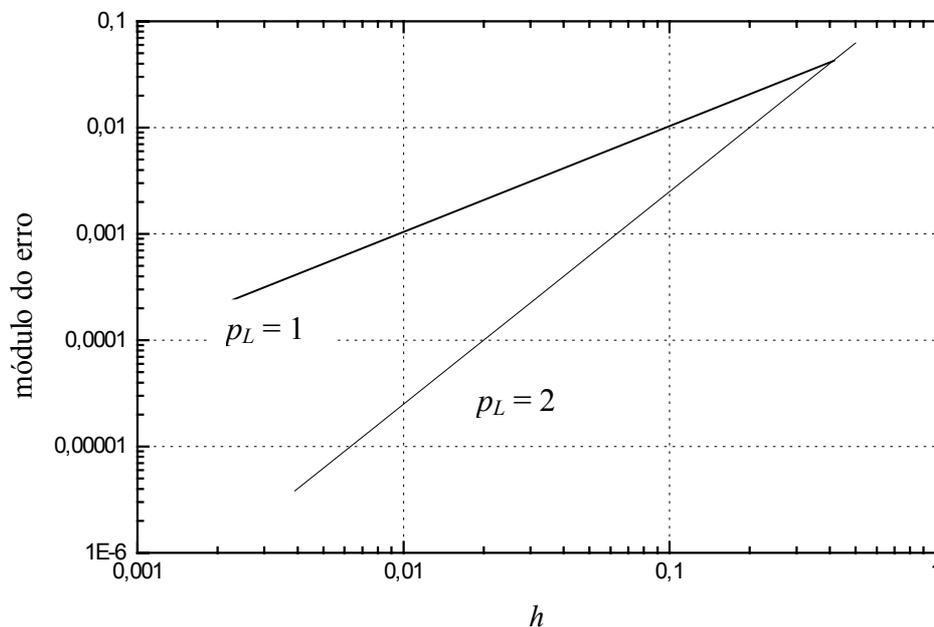


Figura 4.2 Efeito da ordem assintótica (p_L) sobre o erro (E).

Exemplo: $E(\phi_1) = C_1 h_1^{p_L}$ (malha fina)

$E(\phi_2) = C_1 h_2^{p_L}$ (malha grossa)

Portanto,

$$\frac{E(\phi_1)}{E(\phi_2)} = \frac{C_1 h_1^{p_L}}{C_1 h_2^{p_L}} = \left(\frac{h_1}{h_2} \right)^{p_L}$$

Isto é, quanto maior é p_L , maior é a redução do erro com h . Para o caso de $h_1 = \frac{h_2}{2}$ e $p_L = 2$,

temos

$$\frac{E(\phi_1)}{E(\phi_2)} = \left(\frac{h_2}{\frac{h_2}{2}} \right)^2 = \left(\frac{1}{2} \right)^2 = \frac{1}{4}$$

Ou seja, a redução à metade do tamanho dos elementos da malha reduziu a $\frac{1}{4}$ o erro. Quanto maior o p_L , mais rapidamente o erro cai com a redução de h .

4.1.2 Estimativas *a Posteriori*

As estimativas de erro *a posteriori* são usadas para estimar efetivamente a magnitude de erro de discretização. Existem vários métodos que podem ser empregados. Eles podem ser divididos em dois grandes conjuntos:

1. $E(\phi)$ é baseado na solução numérica obtida numa única malha. Em geral o método de elementos finitos (Zhu & Zienkiewicz, 1990) se enquadra neste conjunto.
2. $E(\phi)$ é baseado na soluções numéricas obtidas em duas ou mais malhas diferentes. Em geral os métodos de diferenças finitas e de volumes finitos se enquadram neste conjunto. Alguns estimadores deste tipo são (Marchi, 2001): delta, de Richardson, *GCI*, multicoeficientes e convergente.

4.1.3 Estimador de Richardson

A incerteza (U) de uma solução numérica (ϕ) segundo o estimador de erro de Richardson (Roache, 1994) é obtida admitindo-se que

$$U_{Ri}(\phi) = K_U h^{p_L} \quad (4.5)$$

onde

K_U = coeficiente que é admitido independe de h

h = tamanho dos elementos da malha

p_L = ordem assintótica dos erros de truncamento

Com a Eq. (4.2) em (4.5),

$$\phi_\infty - \phi = K_U h^{p_L} \quad (4.6)$$

Aplicando-se esta equação a duas malhas diferentes ($h_1 =$ fina e $h_2 =$ grossa) cujas soluções numéricas são ϕ_1 e ϕ_2 , obtém-se

$$\phi_{\infty} - \phi_1 = K_U h_1^{p_L} \quad (4.7)$$

$$\phi_{\infty} - \phi_2 = K_U h_2^{p_L} \quad (4.8)$$

Nestas duas equações são conhecidos os valores de ϕ_1, ϕ_2, h_1, h_2 e p_L , e as incógnitas são ϕ_{∞} e K_U . A solução para ϕ_{∞} é

$$\phi_{\infty}(p_L) = \phi_1 + \frac{(\phi_1 - \phi_2)}{(q^{p_L} - 1)} \quad (4.9)$$

onde a razão de refino de malha (q) é dada por

$$q = \frac{h_2}{h_1} \quad (4.10)$$

A Eq. (4.9) representa a extrapolação de Richardson (Roache, 1994). Com sua substituição na Eq. (4.2), a incerteza ou a estimativa do erro de discretização, segundo o estimador de Richardson, para a solução numérica ϕ_1 na malha fina (h_1) é

$$U_{Ri}(p_L) = \frac{(\phi_1 - \phi_2)}{(q^{p_L} - 1)} \quad (4.11)$$

A Eq.(4.6) pode ser reescrita da seguinte forma:

$$\phi_{\infty} - \phi = K_U h^{p_U} \quad (4.12)$$

onde p_U é a ordem aparente (Marchi, 2001) da incerteza. Para malhas grossas, ou práticas, $p_U \neq p_L$, mas para $h \rightarrow 0$, $p_U \rightarrow p_L$. Aplicando-se a Eq. (4.12) a três malhas diferentes ($h_1 =$ fina, $h_2 =$ grossa e $h_3 =$ supergrossa) cujas soluções numéricas são ϕ_1, ϕ_2 e ϕ_3 , obtém-se

$$\phi_{\infty} - \phi_1 = K_U h_1^{p_U} \quad (4.13)$$

$$\phi_{\infty} - \phi_2 = K_U h_2^{p_U} \quad (4.14)$$

$$\phi_\infty - \phi_3 = K_U h_3^{p_U} \quad (4.15)$$

Nestas três equações são conhecidos os valores de ϕ_1 , ϕ_2 , ϕ_3 , h_1 , h_2 e h_3 , e as incógnitas são ϕ_∞ , K_U e p_U . No caso de razão de refino de malha (q) constante entre as malhas h_1 , h_2 e h_3 , isto é,

$$q = \frac{h_3}{h_2} = \frac{h_2}{h_1} \quad (4.16)$$

a solução das Eqs. (4.13) a (4.15) resulta em

$$\phi_\infty(p_U) = \phi_1 + \frac{(\phi_1 - \phi_2)}{(q^{p_U} - 1)} \quad (4.17)$$

onde

$$p_U = \frac{\log\left(\frac{\phi_2 - \phi_3}{\phi_1 - \phi_2}\right)}{\log(q)} \quad (4.18)$$

A Eq. (4.17) representa a extrapolação de Richardson com base na ordem aparente (p_U). Com sua substituição na Eq. (4.2), tem-se

$$U_{Ri}(p_U) = \left(\frac{\phi_1 - \phi_2}{q^{p_U} - 1} \right) \quad (4.19)$$

que é a incerteza da solução numérica (ϕ_1) na malha fina (h_1).

O estimador de Richardson só fornece o valor correto do erro, isto é,

$$U_{Ri}(p_L) = U_{Ri}(p_U) = E(\phi_1) \quad (4.20)$$

em duas situações:

1. no caso-limite de $h \rightarrow 0$; e
2. quando a equação geral do erro de discretização, Eq. (4.3), é composta apenas pelo primeiro termo.

Em geral, nenhuma destas duas situações ocorre na prática, e assim,

$$U_{Ri}(p_L) \neq U_{Ri}(p_U) \neq E(\phi_1) \quad (4.21)$$

Apesar disto, pode-se demonstrar (Marchi e Silva, 2002) que

$$\frac{U_{Ri}(p_L)}{E} < 1 < \frac{U_{Ri}(p_U)}{E} \quad (4.22)$$

quando $p_U \rightarrow p_L$ monotonicamente com valores inferiores a p_L . No caso inverso, isto é, quando $p_U \rightarrow p_L$ monotonicamente com valores superiores a p_L ,

$$\frac{U_{Ri}(p_U)}{E} < 1 < \frac{U_{Ri}(p_L)}{E} \quad (4.23)$$

Além disso, só faz sentido empregar U_{Ri} quando $p_U > 0$.

4.1.4 Estimador *GCI*

De acordo com o estimador *GCI* (*Grid Convergence Index*), a incerteza de uma solução numérica (ϕ_1) é calculada através de (Roache, 1994)

$$U_{GCI}(p_L) = F_s \frac{|\phi_1 - \phi_2|}{(q^{p_L} - 1)} \quad (4.24)$$

onde F_s é um fator de segurança com valor igual a três para aplicações em geral. Este estimador também pode ser empregado com a ordem aparente (p_U). Neste caso,

$$U_{GCI}(p_U) = F_s \frac{|\phi_1 - \phi_2|}{(q^{p_U} - 1)} \quad (4.25)$$

Pode-se demonstrar (marchi, 2001) que

$$\frac{U_{GCI}}{|E(\phi_1)|} \rightarrow F_s \quad \text{para } h \rightarrow 0 \quad (4.26)$$

A representação correta da solução numérica (ϕ) e sua respectiva incerteza (U_{GCI}) obtida com o estimador GCI é

$$\phi = \phi_1 \pm U_{GCI} \quad (4.27)$$

4.2 ERROS DE ITERAÇÃO

A discretização de uma equação diferencial através de um método numérico resulta em

$$[A][\lambda] = [B] \quad (4.28)$$

onde $[A]$ é a matriz dos coeficientes, $[\lambda]$ é o vetor incógnita e $[B]$ é o vetor do termo fonte. No caso do problema abordado na seção 3.1,

$$A_{k,j-1} \lambda_{j-1} + A_{k,j} \lambda_j + A_{k,j+1} \lambda_{j+1} = B_k \quad (\text{para } j = k) \quad (4.29)$$

onde j e k representam cada nó da malha.

O erro de iteração (E_n) da solução numérica (ϕ_n) de uma variável de interesse, na iteração n , é definido por (Fergiger & Peric, 1999)

$$E(\phi_n) = \phi_{n \rightarrow \infty} - \phi_n \quad (4.30)$$

onde $\phi_{n \rightarrow \infty}$ representa a solução exata do sistema (4.28), isto é, sem erros de arredondamento e de programação. A variável ϕ pode representar λ no sistema (4.28) ou qualquer outra variável obtida a partir de λ . Quando o sistema (4.28) tem coeficientes $[A]$ constantes e se usa um método direto para resolvê-lo, como por exemplo eliminação de Gauss, a solução de λ não contém erros de iteração. Algumas causas de erros de iteração são:

- emprego de métodos iterativos para resolver o sistema (4.28);
- problemas não-lineares, isto é, quando os coeficientes $[A]$ dependem de λ ;
- modelos matemáticos constituídos por mais de uma equação, sendo cada uma resolvida separadamente; e
- utilização de métodos *multigrid* para resolver o sistema (4.28).

Três características fundamentais dos erros iterativos são:

1. em geral eles diminuem com o aumento do número de iterações;
2. para $n \rightarrow \infty$, $E(\phi_n) \rightarrow 0$; e
3. para $n \rightarrow \infty$,

$$E(\phi_n) = \frac{C}{10^{n p_L}} \quad (4.31)$$

com

C = coeficiente que independe de n

p_L = ordem assintótica de erro de ϕ

O valor da ordem assintótica (p_L) é sempre maior do que zero e representa a inclinação da curva do erro iterativo num gráfico $\log(E_n)$ versus n para $n \rightarrow \infty$. Essas características podem ser ilustradas pelo exemplo a seguir. Considere a seguinte equação:

$$(\lambda - 2)(\lambda - 3) = 0 \quad \text{ou} \quad \lambda^2 - 5\lambda + 6 = 0 \quad (4.32)$$

A solução numérica de uma das raízes desta equação pode ser obtida pelo método da iteração linear (Barroso *et al.*, 1987); por exemplo, através de

$$\lambda_n = \frac{\lambda_{n-1}^2 + 6}{5} \quad (4.33)$$

Para $\lambda_0 = 0$, a curva de erro obtida é mostrada na Fig. 4.3. A solução exata neste caso é 2 e $p_L = \log(5/4) \cong 0,0969$.

Também há dois tipos de estimativas do erro de iteração: *a priori* e *a posteriori* das soluções numéricas.

4.2.1 Estimativas *a Priori*

As estimativas *a priori* de $E(\phi_n)$ se baseiam na Eq. (4.31). A partir dela temos

$$\frac{E(\phi_{n+1})}{E(\phi_n)} = \frac{1}{10^{p_L}} \quad (4.34)$$

Isto é, quanto maior a ordem assintótica, mais rapidamente o erro de iteração é reduzido com o aumento do número de iterações. Mas, em geral, p_L só pode ser obtido *a priori* para problemas

muito simples. Para a maioria das aplicações, p_L é desconhecido. Normalmente esta dificuldade só é superada com o método exposto na próxima seção.

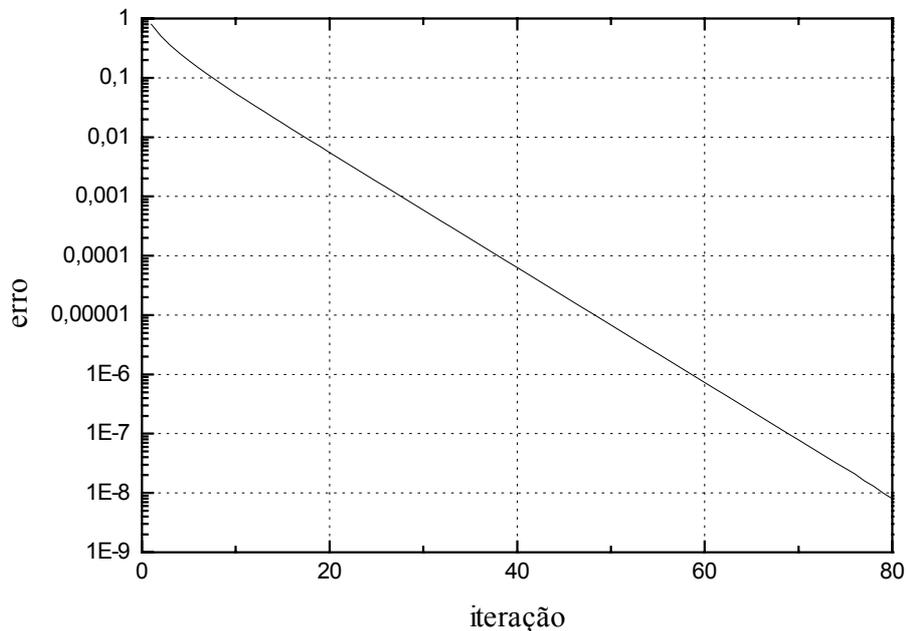


Figura 4.3 Comportamento do erro iterativo com as iterações.

4.2.2 Estimativas *a Posteriori*

Dois métodos para estimar a magnitude do erro iterativo são apresentados por Ferziger & Peric (1999) e Roy & Blottner (2001). Eles são semelhantes ao método apresentado a seguir.

A partir da Eq.(4.31), vamos admitir que

$$U(\phi_n) = \frac{K}{10^{n p_U}} \quad (4.35)$$

onde

K = coeficiente que é admitido independe de n

n = número da iteração

p_U = ordem aparente da incerteza

Com a Eq. (4.2) em (4.35),

$$\phi_\infty - \phi_n = \frac{K}{10^{n p_U}} \quad (4.36)$$

onde, aqui, ϕ_∞ representa uma estimativa da solução exata do sistema (4.28). Escrevendo-se a Eq. (4.36) para soluções numéricas obtidas em três iterações sucessivas ($n-2$, $n-1$ e n), tem-se

$$\phi_\infty - \phi_{n-2} = K 10^{-(n-2) p_U} \quad (4.37)$$

$$\phi_\infty - \phi_{n-1} = K 10^{-(n-1) p_U} \quad (4.38)$$

$$\phi_\infty - \phi_n = K 10^{-n p_U} \quad (4.39)$$

Nestas três equações, ϕ_{n-2} , ϕ_{n-1} , ϕ_n e n são conhecidos. As incógnitas são ϕ_∞ , K e p_U . Com a solução deste sistema, obtém-se

$$\phi_\infty = \phi_n + \frac{(\phi_n - \phi_{n-1})}{(10^{p_U} - 1)} \quad (4.40)$$

onde

$$p_U = \log \left(\frac{\phi_{n-1} - \phi_{n-2}}{\phi_n - \phi_{n-1}} \right) \quad (4.41)$$

A substituição da Eq. (4.40) em (4.2) finalmente fornece a incerteza (U) da solução numérica na iteração n , dada por

$$U(\phi_n) = \frac{(\phi_n - \phi_{n-1})}{(10^{p_U} - 1)} \quad (4.42)$$

Esta estimativa do erro de iteração só deve ser empregada para $p_U > 0$. Note que o erro de truncamento é eliminado na Eq. (4.42) devido à subtração da solução numérica na mesma malha.

Pode-se demonstrar que

$$\frac{U(\phi_n)}{E(\phi_n)} \rightarrow 1 \quad \text{para} \quad n \rightarrow \infty \quad (4.43)$$

4.3 ERROS DE ARREDONDAMENTO

O erro de arredondamento de um variável de interesse (ϕ) é definido por

$$E(\phi_\pi) = \Phi - \phi_\pi \quad (4.44)$$

onde

Φ = solução analítica exata da variável de interesse

ϕ_π = solução numérica sem erros de discretização, de iteração e de programação

Esta definição se aplica mais ao cálculo de uma solução analítica por computação numérica. No caso de soluções numéricas,

$$E(\phi_\pi) = \phi_{n \rightarrow \infty} - \phi_\pi \quad (4.45)$$

onde

$\phi_{n \rightarrow \infty}$ = solução numérica sem erros de iteração, de arredondamento e de programação, mas com erros de discretização

ϕ_π = solução numérica com erros de discretização e de arredondamento, mas sem erros de programação e de iteração.

A causa dos erros de arredondamento é a representação finita das variáveis nas computações numéricas, que está ligada à precisão dos números. Esta precisão depende do compilador (*software*) usado para gerar o código computacional e do computador (*hardware*) empregado em sua execução.

Os erros de arredondamento provocam perda de precisão dos números, que corre basicamente por dois motivos:

- 1) Número de operações nos cálculos: provoca perda de precisão no lado direito dos números. Ocorre por exemplo, na obtenção da solução do sistema (4.28).
- 2) Cancelamento subtrativo nos cálculos: provoca perda de precisão no lado esquerdo dos números. Ocorre, por exemplo, no cálculo de erros e incertezas.

Um modelo grosseiro que permite estimar a perda de precisão (Δp) devido ao número de operações é

$$\Delta p_{\max} = O(N) \text{ a } O(N^2) \quad (4.46)$$

onde N é o número máximo de elementos nas dimensões do problema, e $O(N)$ indica a ordem de grandeza da variável entre parênteses. Alguns exemplos são apresentados a seguir.

Exemplo 1: $N = 100 = 10^2 \quad \rightarrow \quad O(N) = 2$

$$N^2 = (100)^2 = 10^4 \quad \rightarrow \quad O(N^2) = 4$$

Portanto: $\Delta p_{\max} = 2$ a 4 algarismos

Se $\phi = 1,2345678 \times 10^3$, 2, 3 ou 4 algarismos do lado direito podem estar afetados por erros de arredondamento.

Exemplo 2: $N = 65536 = 6,536 \times 10^4 \quad \rightarrow \quad O(N) = 4$

$$N^2 = (65536)^2 \cong 4,29 \times 10^9 \quad \rightarrow \quad O(N^2) = 9$$

Portanto: $\Delta p_{\max} = 4$ a 9 algarismos

Se $\phi = 1,2345678 \times 10^3$, é possível que todos os algarismos deste número estejam afetados por erros de arredondamento e, assim, seu valor não tem utilidade.

Exemplo 3: Malha 2D = 64 x 1024

$$N_{\max} = 1024 = 1,24 \times 10^3 \quad \rightarrow \quad O(N) = 3$$

$$(N^2)_{\max} = (1024)^2 \cong 1,049 \times 10^6 \rightarrow \quad O(N^2) = 6$$

Portanto: $\Delta p_{\max} = 3$ a 6 algarismos

O efeito do número de operações sobre os erros de arredondamento da solução numérica de uma variável de interesse é mostrado na Fig. 4.4. Grosseiramente, quanto menor o tamanho dos elementos da malha (h) ou maior o número de nós (N), maior é o valor do erro de arredondamento.

Um exemplo de cancelamento subtrativo é dado a seguir. Duas soluções numéricas obtidas são $\phi_1 = 1,2345678 \times 10^3$ e $\phi_2 = 1,2344567 \times 10^3$. Ao se calcular a incerteza (U) com as Eqs. (4.11), (4.19), (4.24), (4.25) ou (4.42), no numerador tem-se a diferença entre estas soluções, isto é,

$$\phi_1 = 1,2345678 \times 10^3$$

$$- \phi_2 = 1,2344567 \times 10^3$$

$$\phi_1 - \phi_2 = \quad 1,111 \times 10^{-1}$$

Portanto, neste caso, o cancelamento subtrativo provocou a perda de 4 algarismos do lado esquerdo.

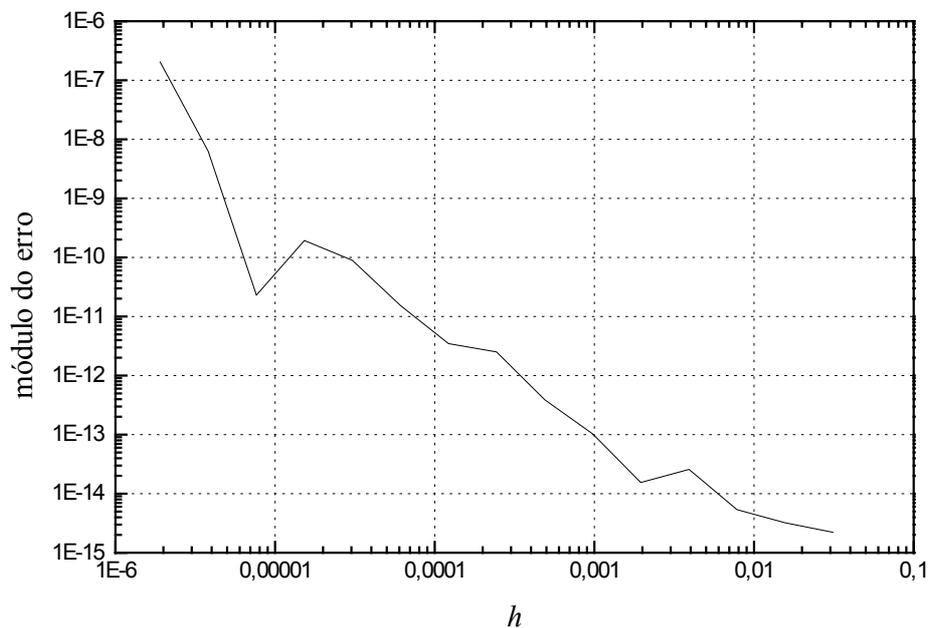


Figura 4.4 Exemplo do comportamento geral dos erros de arredondamento.

4.4 ERROS DE PROGRAMAÇÃO

O erro de programação de uma variável de interesse (ϕ) é definido por

$$E(\phi_p) = \phi_{n \rightarrow \infty} - \phi_p \quad (4.47)$$

onde

$\phi_{n \rightarrow \infty}$ = solução numérica com erros de discretização mas sem erros de iteração, de arredondamento e de programação

ϕ_p = solução numérica com erros de discretização e de programação mas sem erros de iteração e de arredondamento

Os erros de programação podem ser causados (Roache, 1998):

- pelos erros resultantes do uso incorreto de um modelo numérico na aproximação do modelo matemático;

- pelos erros gerados na implementação do modelo numérico num programa computacional;
- pelos erros cometidos no uso do programa computacional durante a obtenção da solução numérica; e
- por qualquer outra eventual fonte de erro.

Nos livros de Ferziger & Peric (1999), Maliska (1995) e Roache (1998b) existem discussões sobre erros de programação: como evitá-los e como detectá-los. Sugestões gerais são:

1. inicialmente, implemente um programa enxuto, específico, depois tente generalizá-lo;
2. implemente o programa em módulos; isso facilita a detecção e eliminação dos erros de programação; use, por exemplo, um módulo para calcular coeficientes e termos fontes e outro para resolver o sistema de equações;
3. teste o *solver* para um sistema de equações simples com resultados exatos conhecidos;
4. usando uma malha pequena, verifique se a solução converge, isto é, se o erro de iteração atinge o nível do erro de máquina ou erro de arredondamento; e
5. resolva um problema “fabricado” para verificar se:

$$\left. \begin{array}{l} E_h \rightarrow 0 \\ p_U \rightarrow p_L \end{array} \right\} \text{ para } h \rightarrow 0$$

Na Fig. 4.5 representa-se o erro de programação detectado no tipo de teste mencionado no item 5 acima.

Shih (1985) e Roache (2002) apresentam procedimentos para “fabricar” problemas com o intuito de verificar erros de programação, em particular, e erros numéricos, em geral. O objetivo é encontrar ou fabricar uma solução analítica exata para um problema similar ao que se deseja resolver numericamente. Exemplo: vamos considerar que o problema a resolver seja

$$L(\Lambda) = \frac{d^2\Lambda}{dX^2} + X \frac{d\Lambda}{dX} + \Lambda^2 = X \quad (4.48)$$

para $0 < X < 1$, com $\Lambda(0) = 0$ e $\Lambda'(1) = 2$. E vamos definir que a solução analítica exata desejada seja

$$\Lambda = X^2 \quad (4.49)$$

Através do método das soluções fabricadas, encontra-se

$$L(\Lambda) = X + S(X)C \quad (4.50)$$

onde

$$S(X) = X^4 + 2X^2 - X + 2 \quad (4.51)$$

O coeficiente C é usado para ativar ($C = 1$) ou desativar ($C = 0$) a solução fabricada na solução numérica. Desta forma, verificando-se o erro numérico para a Eq. (4.50), espera-se que automaticamente o erro numérico também esteja verificado para a Eq. (4.48), a original.

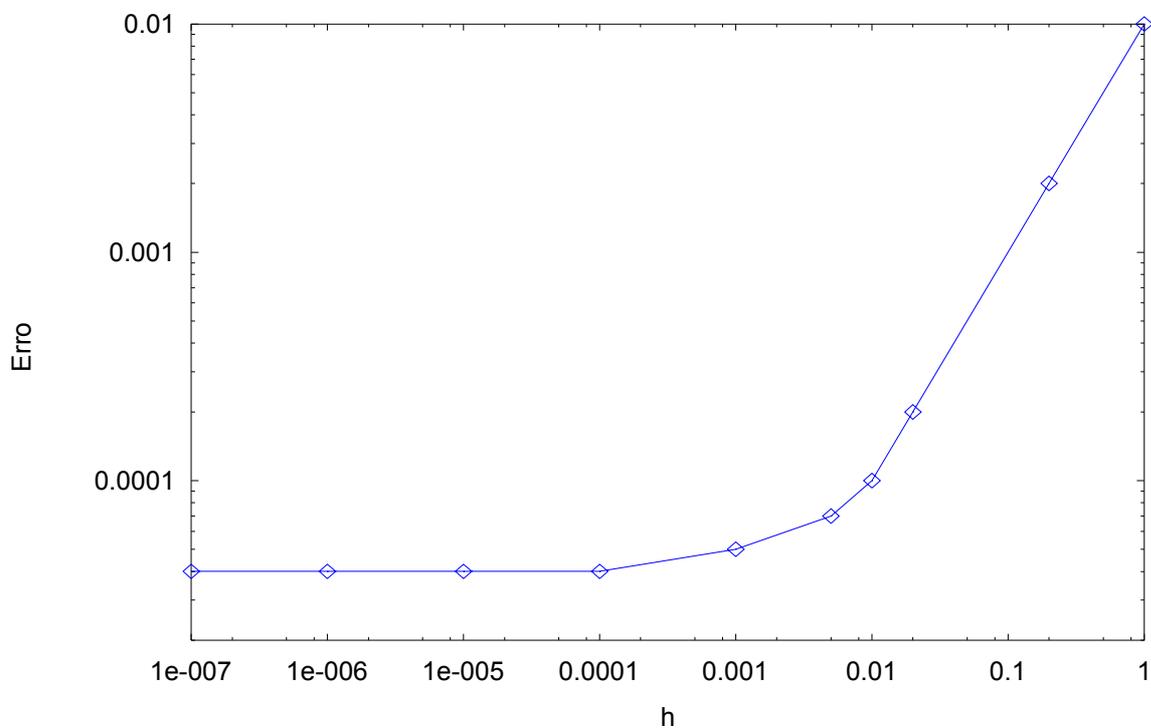


Figura 4.5 Exemplo de erro de programação.

4.5 PROCEDIMENTO PRÁTICO

Para cada problema novo que você queira resolver numericamente, recomenda-se o procedimento a seguir. Primeiro, execute os cinco passos sobre erros de programação descritos na seção anterior. Segundo, para cada caso (conjunto de dados) e variável de interesse (ϕ):

- a) Obtenha três soluções numéricas em três malhas diferentes para o mesmo número de iterações (n): $\phi_{1,n}$, $\phi_{2,n}$ e $\phi_{3,n}$.

- b) Considere as soluções numéricas, na malha mais fina, obtidas nas últimas três iterações: $\phi_{1,n}$, $\phi_{1,n-1}$ e $\phi_{1,n-2}$.
- c) Estime o erro iterativo da malha fina, $U_n(\phi_{1,n})$, através das Eqs. (4.41) e (4.42), lembrando que só é válido $p_U > 0$ na Eq. (4.41).
- d) Estime o erro de truncamento da malha mais fina através das Eqs. (4.24), (4.25) e (4.18), obtendo $U_{GCI}(p_L; \phi_{1,n})$ e $U_{GCI}(p_U; \phi_{1,n})$, lembrando que só é válido $p_U > 0$ na Eq. (4.18); considere o maior dos dois, representando-o por $U_h(\phi_{1,n})$.
- e) Verifique a magnitude relativa entre os erros de truncamento e de iteração através de

$$\frac{U_h(\phi_{1,n})}{U_n(\phi_{1,n})} > 10 \quad (4.52)$$

Se esta relação for satisfeita, os erros de iteração não devem estar comprometendo a estimativa do erro de truncamento. Mas se ela não for satisfeita, deve-se aumentar o número de iterações no item “a” e repetir o procedimento.

- f) Finalmente, o erro numérico estimado é

$$U(\phi_{1,n}) = U_h(\phi_{1,n}) + |U_n(\phi_{1,n})| \quad (4.53)$$

e a representação da solução numérica,

$$\phi = \phi_{1,n} \pm U(\phi_{1,n}) \quad (4.54)$$