Capítulo 1 - Erros e Aritmética Computacional

Carlos Balsa

balsa@ipb.pt

Departamento de Matemática
Escola Superior de Tecnologia e Gestão de Bragança

2º Ano - Eng. Civil, Química e Gestão Industrial

Outline

Métodos Numéricos

- O que é a computação científica (tradicionalmente análise numérica)?
  - Desenho e análise de algoritmos para a resolução numérica problemas matemáticos oriundos da ciência e da engenharia
  - Lida com quantidades contínuas
  - Tem em conta as aproximações
- Nos métodos numéricos concentramos a nossa atenção nos principais métodos de cálculo usados na computação científica
- Para que serve a computação científica?
  - Simular fenómenos naturais
  - Conceber protótipos virtuais concebidos pela engenharia

Origem das aproximações

- Antes da computação
  - Modelação
  - Medições empíricas
  - Computações anteriores
- Durante a computação
  - Truncatura ou discretização
  - Arredondamentos
- A exactidão dos resultados finais reflecte todas as aproximações
- A incerteza dos dados introduzidos (*input*) pode ser amplificada pelo problema
- Perturbações durante a computação podem ser amplificadas pelo algoritmo

Exemplo: aproximações

- Calcular a superfície terrestre através da formula utilizando a formula $A = 4\pi r^2$ envolve várias aproximações
  - A Terra é modelada como uma esfera, idealizando a sua forma ideal
  - O valor do raio é baseado em medidas empíricas e em computações anteriores
  - O valor de π requer a truncatura de processos infinitos
  - O valor dos *inputs* assim como das operações aritméticas são arredondadas no computador

Erro Absoluto e Erro Relativo

- Erro absoluto = valor aproximado - valor exacto
- Erro relativo = $\frac{\text{erro absoluto}}{\text{valor exacto}}$
- Ou, valor aproximado = valor exacto $\times$ (1+ erro rel.)
- Valor exacto, normalmente desconhecido, pelo que é estimado ou limitado através de um erro máximo
- Erro relativo é geralmente baseado no valor aproximado em vez do valor exacto (desconhecido)

Erros de dados e erros de computação

- Problema típico: calcular o valor da função $f : \mathbb{R} \rightarrow \mathbb{R}$ para os argumentos
  - x valor exacto do argumento
  - $f(x)$ valor pretendido
  - $\hat{x}$ valor aproximado do input
  - $\hat{f}$ valor aproximado da função a calcular
- Erro total = $\hat{f}(\hat{x}) - f(x)$

  $= (\hat{f}(\hat{x}) - f(\hat{x})) + (f(\hat{x}) - f(x))$

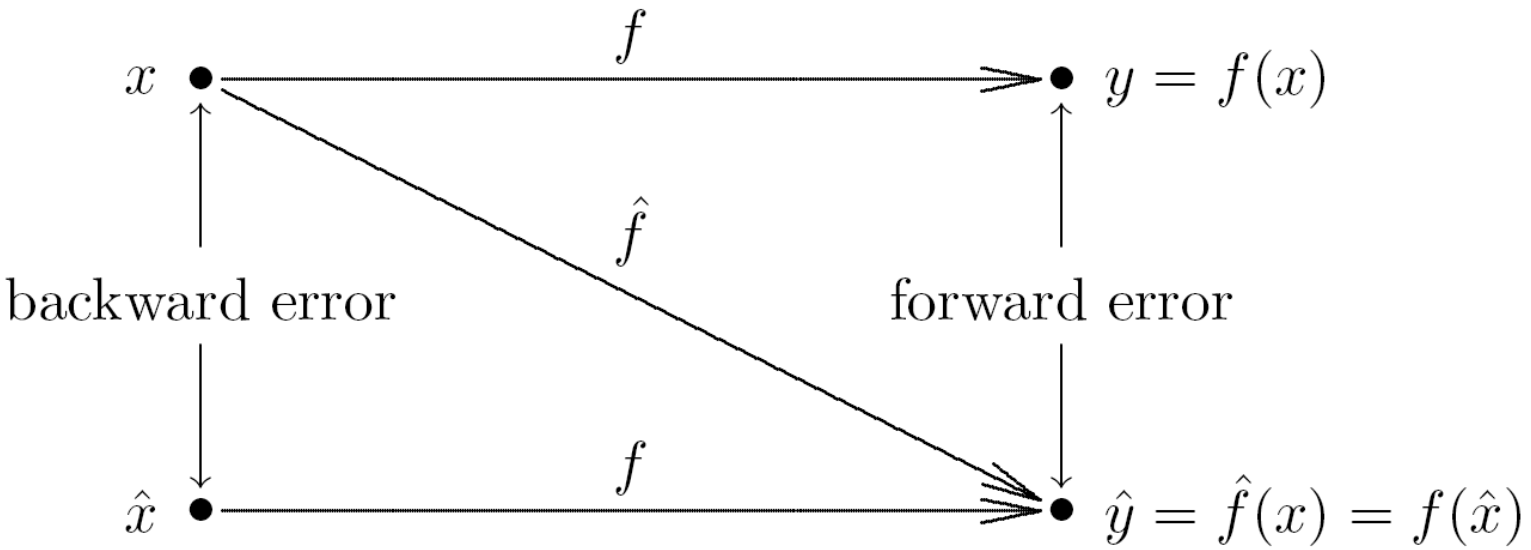
  = erro computacional + erro propagado
- O algoritmo não tem efeito no erro propagado

Erro de truncatura e erro de arredondamento

- Erro de truncatura: diferença entre o resultado exacto (para o input actual) e o resultado produzido pelo algoritmo usando uma aritmética exacta
  - → Devido a aproximações tais como a truncatura de séries infinitas ou fins de processos iterativos antes de se verificar a convergência
- Erro de arredondamento: diferença entre o resultado produzido pelo algoritmo usando aritmética infinita e o resultado produzido pelo mesmo algoritmo usando uma aritmética de precisão limitada
  - → Devido a representação inexacta de números reais e às operações inexactas sobre esses números
- Os erros computacionais são a soma dos erros de truncatura e dos erros de arredondamento, normalmente, um destes é dominante

Erro anterior (backward error) e erro posterior (forward error)

- Supondo que queremos calcular $y = f(x)$, com $f : \mathbb{R} \to \mathbb{R}$, mas obtemos o valor aproximado $\hat{y}$
  - Erro posterior $\Delta y = \hat{y} - y$
  - Erro anterior $\Delta x = \hat{x} - x$, com $f(\hat{x}) = \hat{y}$



Exemplo: erro anterior e erro posterior

- Como aproximação a $\sqrt{2}$, $\hat{y} = 1.4$ tem como erro absoluto posterior

$$|\Delta y| = |\hat{y} - y| = |1.4 - 1.41421| \approx 0.0142$$

que corresponde a um erro relativo posterior de cerca de 1%

- Uma vez que $\sqrt{1.96} = 1.4$, o erro absoluto anterior é

$$|\Delta x| = |\hat{x} - x| = |1.96 - 2| \approx 0.04$$

que corresponde a um erro relativo anterior de cerca de 2%

Análise do erro anterior

- Ideia: solução aproximada é a solução exacta do problema modificado
- De quanto deve ser modificado o problema original para originar o resultado obtido?
- Quanto é que o os erros nos inputs podem explicar todos os erros nos resultados calculados?
- A solução aproximada é boa se for a solução exacta de um problema próximo do original
- O erro anterior é por vezes mais fácil de estimar do que o erro à posterior

Exemplo: análise do erro anterior

- Vamos aproximar a função cosseno $f(x) = \cos(x)$ através da série de Taylor truncada a partir dos 3 primeiros termos

$$\hat{y} = \hat{f}(x) = 1 - x^2/2$$

- O erro posterior é dado por

$$\Delta y = \hat{y} - y = \hat{f} - f = 1 - x^2/2 - \cos(x)$$

- Para determinar o erro anterior, necessitamos do valor $\hat{x}$ tal que $f(\hat{x}) = \hat{f}(x)$
- Para a função cosseno, $\hat{x} = arccos(\hat{f}(x)) = arccos(\hat{y})$

Sensibilidade e Condicionamento

- Um problema é *insensível* ou *bem condicionado* se mudanças relativas no input provocam mudanças relativas semelhantes na solução
- Um problema é *sensível* ou *mal condicionado* se mudanças relativas no input provocam muito maiores mudanças relativas na solução
- Número de condição

$$\text{Cond} = \frac{|\text{Mud. relativa na sol.}|}{|\text{Mud. relativa nos inputs}|}$$

$$= \frac{|[f(\hat{x}) - f(x)]/f(x)|}{|(\hat{x} - x)/x|} = \left|\frac{\Delta y/y}{\Delta x/x}\right|$$

- O problema é sensível ou mal condicionado se $\text{Cond} \gg 1$

Número de Condição

- O número de condição é um factor de ampliação do erro anterior em relação ao erro posterior

$$|\text{Erro relativo posterior}| = \text{cond} \times |\text{Erro relativo anterior}|$$

- Normalmente o numero de condição não é exactamente conhecido e pode variar com o input, pelo que se usa uma aproximação ou um limite máximo para o valor de **Cond**

$$|\text{Erro relativo posterior}| \leq \text{cond} \times |\text{Erro relativo anterior}|$$

Exemplo: Cálculo de uma função

- Calcular uma função para o input aproximado $\hat{x} = x + \Delta x$ em vez de x
- Erro absoluto posterior: $f(x + \Delta x) - f(x) \approx f'(x)\Delta x$
- Erro relativo posterior: $\frac{f(x+\Delta x)-f(x)}{f(x)} \approx \frac{f'(x)\Delta x}{f(x)}$
- Número de condição: $\text{cond} \approx \left| \frac{f'(x)\Delta x/f(x)}{\Delta x/x} \right| = \left| \frac{xf'(x)}{f(x)} \right|$
- Erro relativo no valor da função pode ser muito menor ou muito maior do que o erro no input, dependendo de x e de f

Exemplo: sensibilidade

- A função tangente é sensível para argumentos próximos de $\pi/2$
  - $\tan(1.57079) \approx 1.58058 \times 10^5$
  - $\tan(1.57078) \approx 6.12490 \times 10^4$
- Mudança relativa no output é um quarto de milhão maior do que a mudança relativa no input
  - Para $x = 1.57079$, $cond \approx 2.48275 \times 10^5$

Estabilidade e Exactidão

- Um algoritmo é estável se o resultado produzido for relativamente insensível a perturbações durante a computação
- Estabilidade de um algoritmo é análoga ao condicionamento do problema original
- Exactidão: proximidade da solução calculada da solução exacta do problema
- Exactidão depende do condicionamento do problema assim como da estabilidade do algoritmo
- A inexactidão pode resultar de aplicar algoritmos estáveis a problemas mal condicionados ou algoritmos instáveis a problemas bem condicionados

Notação de Virgula Flutuante

- Nos computadores os números são representados por um sistema de números de vírgula (ou ponto) flutuante da forma

$$x = \pm f_x * b^E$$

em que

f_x: mantissa (fracção)
b: base
E: expoente

- Maior parte dos computadores modernos são concebidos de acordo o sistema de ponto flutuante do IEEE, em que a base é binária ($b = 2$)
- Os computadores convertem os inputs, na base decimal ($b = 10$), para a base binária antes de efectuar as operações pedidas, posteriormente convertem também os resultados para a base decimal antes de serem apresentados

- A forma padrão de representar um numero em computador é através da notação científica

$$x = \pm f_x * 10^E$$

em que $1 \leq f_x < 10$ (todos os dígitos de f_x são significativos)

- Na notação científica normalizada tem-se $0.1 \leq f_x < 1$

  Em análise de erros esta notação é útil pois verifica a relação $-m = E - t$, em que m é o numero de posições décimas, t é o número de dígitos significativos e E é o expoente na base 10 quando x está representado nesta notação

- Por exemplo $x = 0.0003450$

  $x = 3.450 * 10^{-4}$ ou $x = 3.450E - 4$: notação científica
  $x = 0.3450 * 10^{-3}$ ou $x = 0.3450E - 3$: not. normalizada

- A exactidão do sistema de ponto flutuante é caracterizado pela unidade de arredondamento (ou precisão máquina), representada por ϵ_{maq}
- Corresponde ao número de dígitos de precisão com que um número real é representado no sistema de ponto flutuante
- É o erro relativo máximo que se comete ao representar um número real no sistema de ponto flutuante
- No sistema IEEE de precisão simples $\epsilon_{maq} \approx 10^{-7}$ e no sistema IEEE de precisão dupla $\epsilon_{maq} \approx 10^{-16}$

- O valor mínimo (em valor absoluto) que é possível representar no sistema de ponto flutuante é designado por *underflow*
- O valor máximo (em valor absoluto) que é possível representar no sistema de ponto flutuante é designado por *overflow*
- No decorrer da execução de um algoritmo se o *overflow* ocorre verifica-se um erro fatal responsável pelo fim precipitado da execução
- Não confundir *underflow* com ϵ_{maq}, embora ambos sejam pequenos, a precisão máquina depende do número de dígitos na mantissa (f_x) enquanto que o *underflow* é determinado pelo número de dígitos no campo do expoente (E)
- Num sistema de ponto flutuante temos

$$0 < underflow < \epsilon_{maq} < overflow$$