Computação Científica

SEMANA 2

ERROS EM REPRESENTAÇÕES NUMÉRICAS E

ARITMÉTICA EM PONTO FLUTUANTE

PARTE I

ERROS E ERROS DE ARREDONDAMENTO

(FINAL)

Computação Científica

prof. Marco Villaça

Aritmética de ponto flutuante
Cancelamento subtrativo

- É o arredondamento induzido quando se subtrai dois números de ponto flutuante muito próximos.

- Pode ocorrer, por exemplo, quando se calcula as raízes de uma equação quadrática com a fórmula de Baskara:

$$x_{1,} x_2 = \frac{-b \pm \sqrt{(b^2 - 4ac)}}{2a}$$

- Quando $b^2 >> 4ac$, as magnitudes do discriminante e de b podem ser praticamente as mesmas, isto é

$$b \cong \sqrt{(b^2 - 4ac)}$$

Aritmética de ponto flutuante
Regra 5

- Nessa situação, o cancelamento subtrativo certamente levará a um erro significativo em uma das raízes. Assim, uma quinta regra a ser considerada é

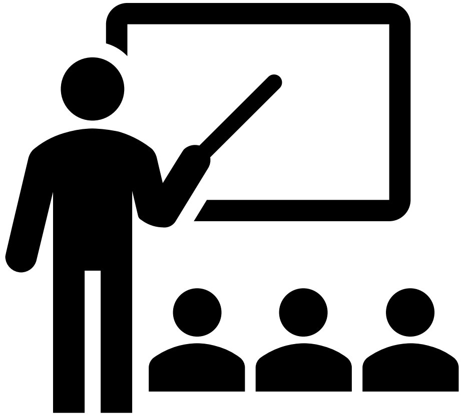
  Evitar a subtração de números muito próximos.

Aritmética de ponto flutuante
Cancelamento subtrativo

- Para minimizar o problema
  - ✓ Usar dupla precisão
  - ✓ Alterar a fórmula de Báskara:

$$x_1, x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \times \frac{-b \mp \sqrt{b^2 - 4ac}}{-b \mp \sqrt{b^2 - 4ac}}$$

$$x_1, x_2 = \frac{b^2 - b^2 - 4ac}{2a \times \left(-b \mp \sqrt{b^2 - 4ac}\right)} \Rightarrow x_1, x_2 = -\frac{2c}{b \pm \sqrt{b^2 - 4ac}}$$



- Calcule os valores das raízes de uma equação quadrática com $a = 1$, $b = 3000.001$ e $c = 3$.

  Compare os valores computados em precisão simples em Linguagem C com as raízes reais $x_1 = -0.001$ e $x_2 = -3000$:

  - Utilizando *Baskara*;
  - Utilizando a equação alternativa (1.10);
- Explique os resultados.

EXEMPLO 4

EXEMPLO 4 Código C

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float a = 1, b = 3000.001, c = 3, x1, x2,delta;
    delta = sqrt(b*b-4*a*c);
    x1 = (-b + delta)/(2*a);
    x2 = (-b - delta)/(2*a);
    printf("As raizes pela eq.  de Baskara sao x1 =  %f e x2 = %f\n",x1,x2);
    x1 = (-2*c)/ (b+delta);
    x2 = (-2*c)/ (b-delta);
    printf("As raizes pela eq. alternativa sao x1 =  %f e x2 = %f\n",x1,x2);
    return 0;
}
```

```
As raizes pela eq.  de Baskara sao x1 =  -0.000977 e x2 = -3000.000000
As raizes pela eq. alternativa sao x1 =  -0.001000 e x2 = -3072.000000

Process returned 0 (0x0)   execution time : 0.328 s
Press any key to continue.
```

EXEMPLO 4
Explicação

- Empregando Báskara, embora os resultados para x_2 são adequados, o percentual de erro relativo para x_1 é alto, $\varepsilon_t = 2{,}4\%$. Este nível poderia ser inadequado para muitos problemas de engenharia aplicada.
- Este resultado é particularmente surpreendente porque se está empregando uma fórmula analítica para obter a solução.
- A perda de significância ocorre onde duas quantidades muito próximas são subtraídas, o que não ocorre quando os mesmos número são adicionados.

EXEMPLO 4
Explicação

- Com base no exposto, podemos deduzir a conclusão geral de que a fórmula de Báskara será suscetível ao cancelamento subtrativo sempre que $b^2 >> 4ac$.
- Uma forma de contornar o problema seria usar dupla precisão, minimizando o cancelamento subtrativo.
- Outra, seria reformular a fórmula quadrática no formato alternativo para calcular x_1, evitando o cancelamento subtrativo.

Aritmética de Ponto Flutuante Regra 6

- Já que as operações aritméticas podem potencialmente contribuir para o erro total de arredondamento, então faz sentido tornar os procedimentos mais eficientes. Isso nos leva a formular uma sexta regra:

Minimize o número de operações aritméticas.

Regra de Horner

- Suponha que deva ser calculado o seguinte polinômio para um valor específico de n e x:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

- Pouco eficiente, pois o número total de multiplicações e adições realizadas será, respectivamente n(n+1)/2 e n. Alternativamente, a regra de Horner pode ser utilizada para reescrever o polinômio como:

$$p(x) = a_0 + x(a_1 + x(a_2 + \ldots x(a_{n-2} + x(a_{n-1} + x a_n))\ldots))$$

reduzindo para $2n$ o número total de operações de multiplicação e adição combinadas.

1. Considere a função

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

- Reescreva a função racionalizando o numerador, isto é, removendo a raiz quadrada do numerador.
- Usando aritmética decimal de 5 dígitos com corte, calcule f(0,001) utilizando a função original e a função racionalizada.

  Respostas: 0 e $0,5 \times 10^{-6}$.
- Utilizando aritmética de dupla precisão (10 dígitos), calcule novamente f(0,001) utilizando as duas formas da função.
- Utilizando aritmética de dupla precisão, calcule $f(1 \times 10^{-6})$ utilizando as duas formas da função.
- Explique os resultados.

Exercício 4

2. Codifique a regra de Horner em linguagem C, para avaliar

$$f(x) = x^3 - 6{,}1\,x^2 + 3{,}2\,x + 1{,}5 \text{ em } x = 4{,}71$$

Exercício 5

3. Conforme foi visto, a perda de exatidão devido a erros de arredondamento pode ser reduzida pelo rearranjo dos cálculos. Pede-se:

- Avalie $f(x) = x^3 - 6{,}1\,x^2 + 3{,}2\,x + 1{,}5$ em $x = 4{,}71$ usando aritmética de 3 dígitos com arredondamento e corte e calcule os respectivos erros relativos.

  `Respostas: -13,4; -13,5; 6%, 5%`

- Reescreva o polinômio empregando a regra de Horner, e avalie em usando aritmética de 3 dígitos com arredondamento e corte e calcule os respectivos erros relativos.

  `Respostas: -14,3; -14,2; 0,25%; 0;45%`

- Discuta os resultados.

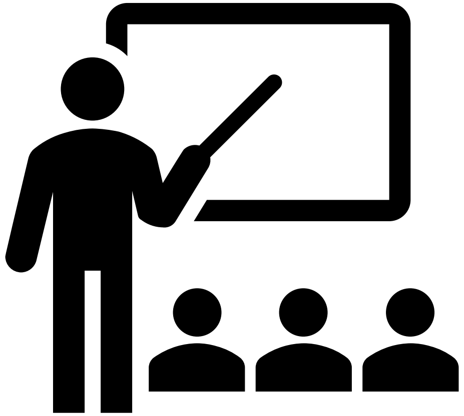
Exercício 5

Teorema da perda de precisão

- Exatamente quantos dígitos binários significativos são perdidos na subtração **x - y** quando x está muito próximo de y?
- A proximidade de x e y é convenientemente medida por **$| 1 - (y / x) |$.**
- **Teorema**

  *Deixe x e y ser números de máquina de ponto flutuante normalizados, onde $x > y > 0$. Se $\mathbf{2^{-p} \leq 1 - (y / x) \leq 2^{-q}}$ para alguns inteiros positivos **p** e **q**, então no máximo p e pelo menos q bits binários significativos são perdidos na subtração $x-y$.*

Na subtração 37,593621 – 37,584216, quantos bits de significância são perdidos?



EXEMPLO 5

EXEMPLO 5

- **Solução:**

Deixe $x = 37{,}593621$ como o primeiro número e $y = 37{,}584216$ como o segundo. Então,

$$1 - y/x = 0{,}0002501754$$

Isto está entre 2^{-12} e 2^{-11}, ou entre 0,000244 e 0,000488.

Portanto, pelo menos 11 mas não mais de 12 bits são perdidos.

ERROS EM REPRESENTAÇÕES NUMÉRICAS E

ARITMÉTICA EM PONTO FLUTUANTE

PARTE II

ERROS DE TRUNCAMENTO
E
ERRO NUMÉRICO TOTAL

Computação Científica

prof. Marco Villaça

ERROS DE TRUCAMENTO
DEFINIÇÃO

- Erros de truncamento são os erros resultantes do uso de uma aproximação no lugar de uma solução matemática exata.

- Conforme foi visto na aula anterior, a expansão em série de Maclaurin para e^x apresenta um número infinito de termos, entretanto, quando a série é utilizada para calcular e^x , somente um número finito de termos pode ser utilizado. Usando três termos para calcular e^x , o erro de truncamento para tal aproximação vale

$$erro\ de\ truncamento = e^x - \left(1 + x + \frac{x^2}{2!}\right) = \frac{x^3}{3!} + \frac{x^4}{4!} \dots + \frac{x^n}{n!}$$

ERROS DE TRUCAMENTO
DEFINIÇÃO

- Os erros de truncamento não se originam apenas do corte de uma parte de uma série, eles podem ocorrer em outros procedimentos matemáticos.

- Um exemplo é o erro que ocorre quando um processo contínuo é substituído por uma aproximação discreta. Ao encontrar a derivada de uma função, definida por

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

- Numericamente não se pode utilizar $\Delta x \to 0$, é necessário utilizar um valor finito de x, resultando em

ERROS DE TRUCAMENTO
DEFINIÇÃO

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

- Nesse caso, portanto, o erro de truncamento é causado pela escolha de um valor finito de Δx.
- Por exemplo, ao se calcular a derivada de $f(x) = x^2$, o erro de truncamento será

$$erro\ de\ truncamento\ = 2x - \frac{(x+\Delta x)^2 - x^2}{\Delta x}$$

$$erro\ de\ truncamento\ = 2x - \frac{x^2 + 2x\Delta x + \Delta x^2 - x^2}{\Delta x} = -\Delta x$$

EXEMPLO 1

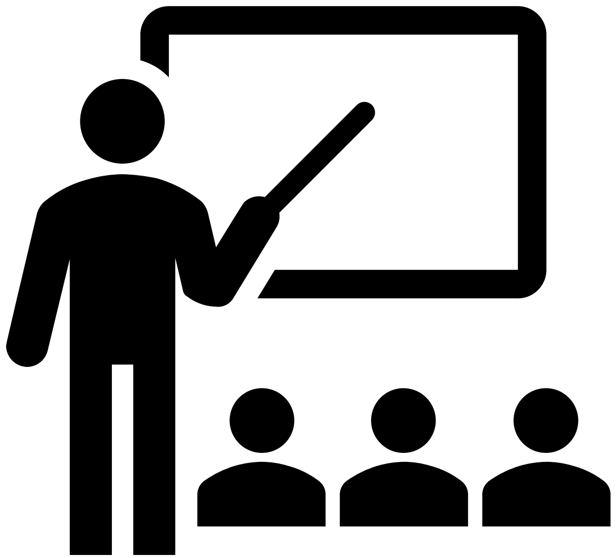
A aceleração a que está submetido um saltador de bungee jumping é descrita pela seguinte equação diferencial:

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$

onde, v é a velocidade em m/s, t é o tempo em s, g é a aceleração da gravidade (9,80665 m/s^2), c_d é o coeficiente de arrasto concentrado em kg/m e m é a massa do saltador em kg.

A solução analítica da equação diferencial é,

$$v(t) = \sqrt{\frac{g \cdot m}{c_d}} \tanh\left(\sqrt{\frac{g \cdot c_d}{m}}\, t\right) \quad (m/s)$$



EXEMPLO 1

- Para resolver numericamente a ED, utiliza-se a aproximação por diferenças finitas:

$$\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} = g - \frac{c_d}{m} v(t_i)^2$$

- Isolando-se $v(t_{i+1})$ na equação acima, resulta

$$v(t_{i+1}) = v(t_i) + \left(g - \frac{c_d}{m} v(t_i)^2\right)(t_{i+1} - t_i)$$

ou

$$v_{i+1} = v_i + \left(g - \frac{c_d}{m} {v_i}^2\right)\Delta t$$

onde Δt é o passo de cálculo

EXEMPLO 1

- Pede-se
  - ✓ Um script Scilab que plote o gráfico de v(t) usando as soluções analíticas e numéricas, do instante t = 0 até t = 12 s, com intervalos de 0,5 s.

    Considere m = 60 kg, g = 9,80665 m/s^2 e c_d = 0,25 kg/m
  - ✓ O erro relativo percentual verdadeiro em t = 12 s.

EXEMPLO 1

```
i = 1; c = 0.25; g= 9.80665; m = 60;
vex(1)=0;  // solução exata
vnum(1)=0; // solução numérica
t(1)=0; dt = 0.5;
while t(i)<12 do
    t(i+1)=t(i)+dt
    vex(i+1)= sqrt(g*m/c)*tanh(sqrt(g*c/m)*t(i+1));
    vnum(i+1)=vnum(i)+(g - (c/m)*(vnum(i)^2))*dt
    i=i+1;
end
```

EXEMPLO 1

```
printf("A soluçao exata de v(%d)= %f\n",t(i),vex(i));
printf("A soluçao numerica de v(%d)= %f\n",t(i),vnum(i));
e = abs((vex(i)-vnum(i))/vex(i))*100;
printf("O erro percentual relativo é  %f %%",e);
plot2d(t,vex,style=[color('blue4')]);
plot2d(t,vnum,style=[color('red4')]);
xgrid
```

Demonstre a solução analítica de:

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$

Exercício 1

Séries de Taylor

- A função suave é uma função que tem derivadas contínuas até alguma ordem desejada sobre algum domínio.
- O Teorema de Taylor estabelece que qualquer função suave pode ser aproximada por um polinômio.
- As séries de Taylor fornece um meio para expressar essa ideia matematicamente.
- A série $\sum_{n \geq 0} \frac{f^{(n)}(a)}{n!}(x-a)^n$ chama-se série de Taylor da função f em torno de a, que no caso de a = 0, é chamada de série de Maclaurin da função f.

Séries de Taylor

$$\sum_{n\geq 0} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

- A aproximação por série de Taylor de uma função, permite estimar o valor da função em um ponto x_{i+1}, conhecido o seu valor em x_i.

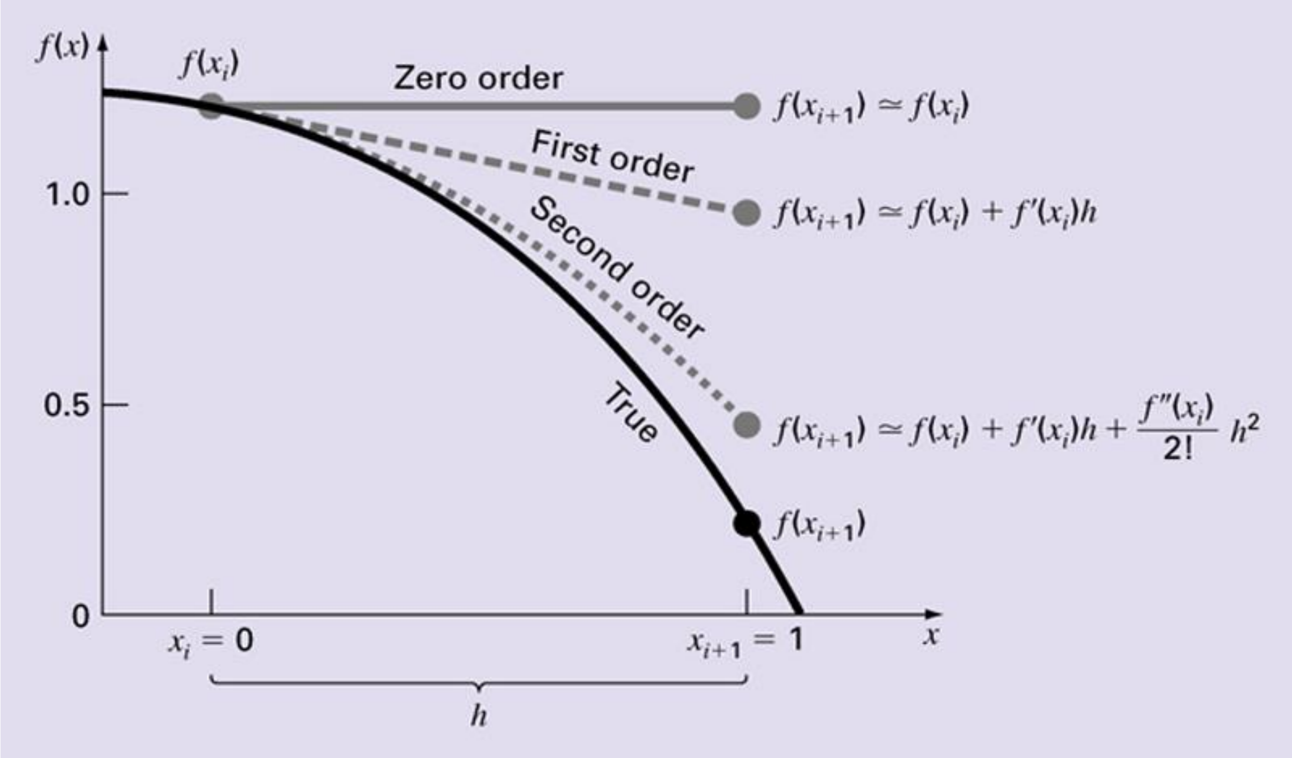
$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

onde

- $h = x_{i+1} - x_i$
- O resto é incluído para representar todos os termos a partir de $n + 1$

Séries de Taylor

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \dots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$



Aproximação de $f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$ em $x = 1$ por expansão em séries de Taylor.
Fonte: Capra, 2013, p. 105

ERROS EM REPRESENTAÇÕES NUMÉRICAS E

ARITMÉTICA EM PONTO FLUTUANTE

PARTE II

ERROS DE TRUNCAMENTO
E
ERRO NUMÉRICO TOTAL

(CONTINUAÇÃO)

Computação Científica

prof. Marco Villaça

Séries de Taylor
Fórmula de Taylor com resto de Lagrange

Se $f : [a, b] \rightarrow R$ *é uma função com n derivadas contínuas até a ordem n+1. Seja* $x_i \in [a, b]$, *então existe um* ξ *entre* x_i *e* x_{i+1} *tal que*

$$f(x_{i+1}) = \sum_{k=0}^{n} \frac{f^{(k)}(x_i)}{k!} h^k + \frac{f^{(n+1)}(\xi)}{n+1!} h^{n+1}$$

sendo

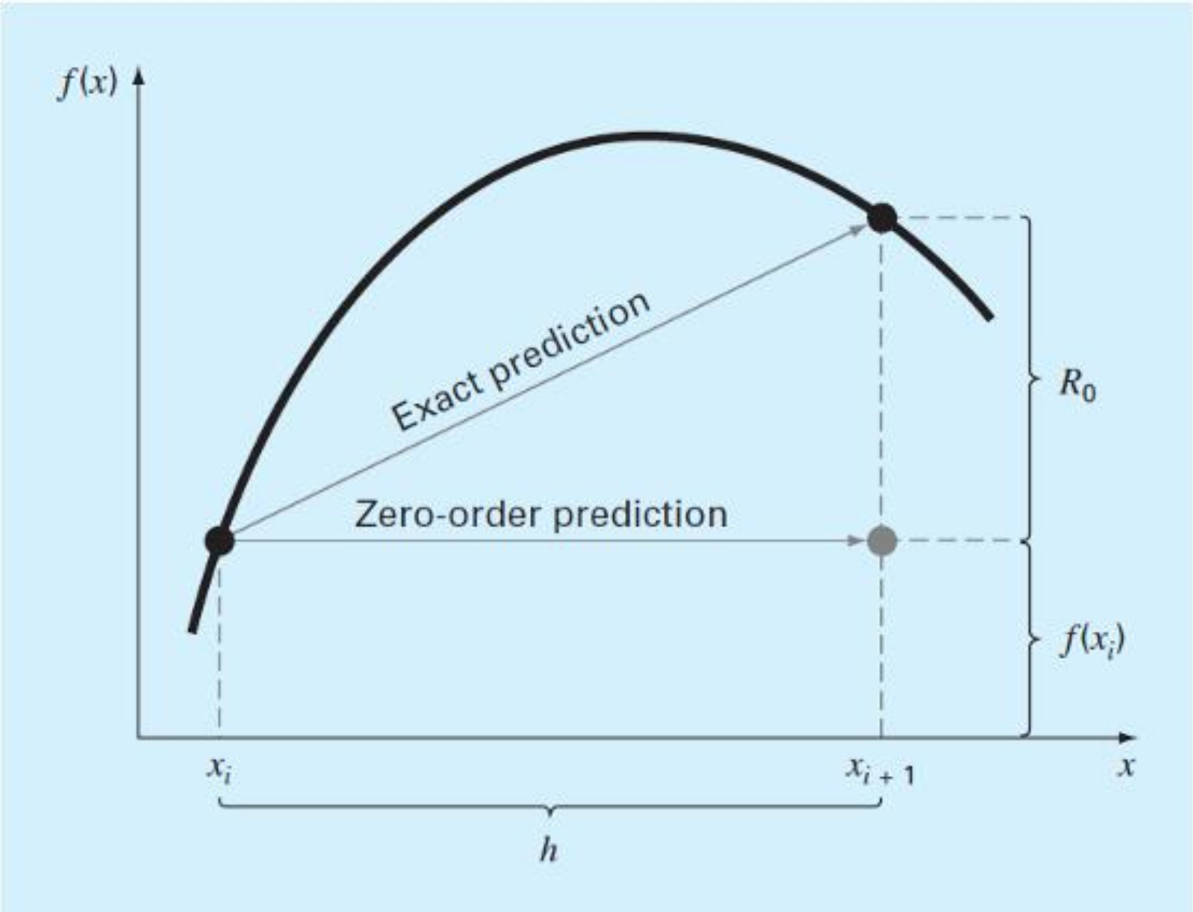
$$\frac{f^{(n+1)}(\xi)}{n+1!} h^{n+1} = R_n$$

Séries de Taylor
Análise do resto

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

- Se $f(x_{i+1}) \approx f(x_i)$
- O resto será

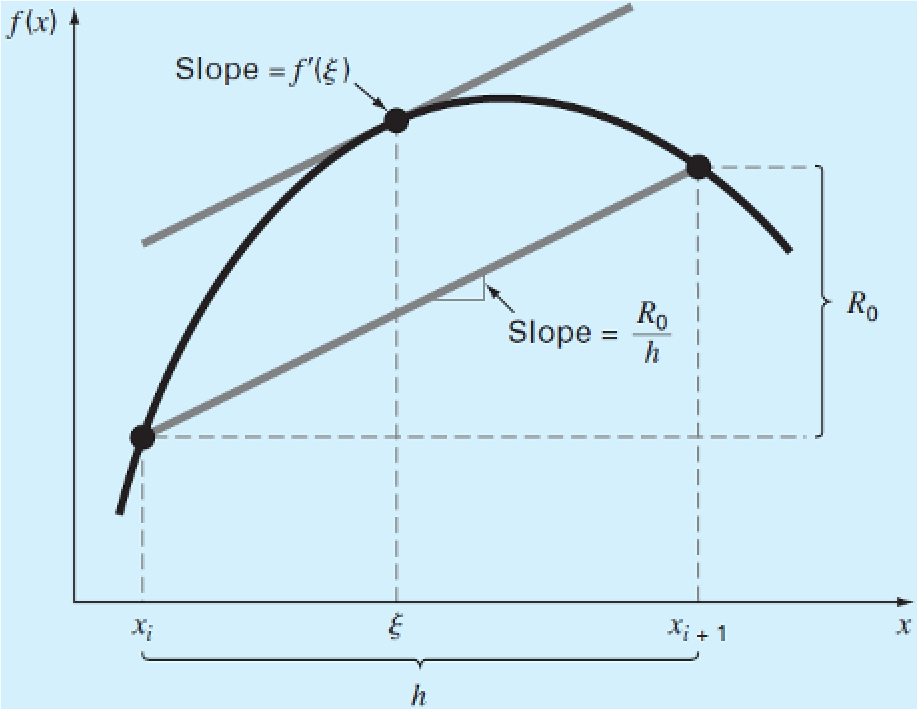
$$R_0 = f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots$$



Séries de Taylor
Análise do resto

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

- Segundo o Teorema do Valor Médio, se uma função $f(x)$ e sua 1ª derivada forem contínuas em um intervalo entre x_i e x_{i+1}, então pelo menos um ponto em f(x), denotado por $f'(\xi)$, tem uma inclinação paralela a reta que une $f(x_i)$ e $f(x_{i+1})$.



- Assim, para a aprox. de ordem 0:

$$f'(\xi) = \frac{R_0}{h} \longrightarrow R_0 = f'(\xi)h$$

- Estendendo para ordens superiores:

$$R_n = \frac{f^{(n+1)}(\xi)}{n+1!}h^{n+1}$$

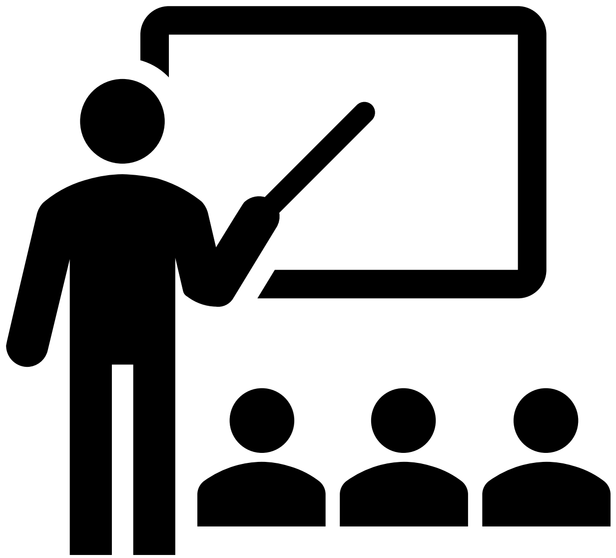
EXEMPLO 2

Na Série de Taylor, se h for suficientemente pequeno, poucos termos serão suficientes para se obter uma estimativa adequada.

Com o auxílio de um script Scilab, use expansões em série de Taylor com n = 0 até 6 para aproximar $f(x) = \cos x$ em $x_{i+1} = \pi/3$ com base no valor de f(x) e suas derivadas em $x_i = \pi/4$

Note que $h = \pi/3 - \pi/4$

Calcule o erro relativo para cada expansão.



EXEMPLO 2

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

```
x1 = %pi/3; x0 = %pi/4;
fx = cos(x0); i = 0; j = 0; vreal = cos(x1);
while i<=6 do
   e = abs((vreal-fx)/vreal)*100
   printf("Ordem = %d, f(x_i+1)= %.10f , erro = %.2e\n",i, fx, e);
   i=i+1;
   j=j+1;
   if j==1 then
      der = -sin(x0);
   elseif j==2 then
      der = -cos(x0);
   elseif j==3 then
      der = sin(x0);
   elseif j==4 then
      der = cos(x0);
      j = 0;
   end
   fx = fx + der*((x1-x0)^i)/factorial(i);
end
```

Exemplo 2 – Execução do script Scilab

```
Ordem = 0, f(x_i+1)= 0.7071067812 , erro = 4.14e+01
Ordem = 1, f(x_i+1)= 0.5219866588 , erro = 4.40e+00
Ordem = 2, f(x_i+1)= 0.4977544914 , erro = 4.49e-01
Ordem = 3, f(x_i+1)= 0.4998691469 , erro = 2.62e-02
Ordem = 4, f(x_i+1)= 0.5000075508 , erro = 1.51e-03
Ordem = 5, f(x_i+1)= 0.5000003040 , erro = 6.08e-05
Ordem = 6, f(x_i+1)= 0.4999999878 , erro = 2.44e-06
```

Observação: erros percentuais.

Séries de Taylor

Estimativa do erro de truncamento

- Seja a expansão em série de Taylor de f(x):

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \ldots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n$$

Truncando-se a partir do termo de primeira ordem:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + R_1$$

Isolando-se f '(x):

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{(x_{i+1} - x_i)} - \frac{R_1}{(x_{i+1} - x_i)}$$

Aproximação de primeira ordem

Erro de truncamento

Séries de Taylor

Estimativa do erro de truncamento

$$R_n = \frac{f^{(n+1)}(\xi)}{n+1!} h^{n+1}$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{(x_{i+1} - x_i)} - \frac{R_1}{(x_{i+1} - x_i)}$$

Como

$$R_1 = \frac{f^{('')}(\xi)}{2!}(x_{i+1} - x_i)^2$$

O erro de truncamento pode ser expresso por

$$\frac{R_1}{x_{i+1} - x_i} = \frac{f^{('')}(\xi)}{2!}(x_{i+1} - x_i)$$

ou

$$\frac{R_1}{x_{i+1} - x_i} = O(x_{i+1} - x_i)$$

Séries de Taylor

Estimativa do erro de truncamento

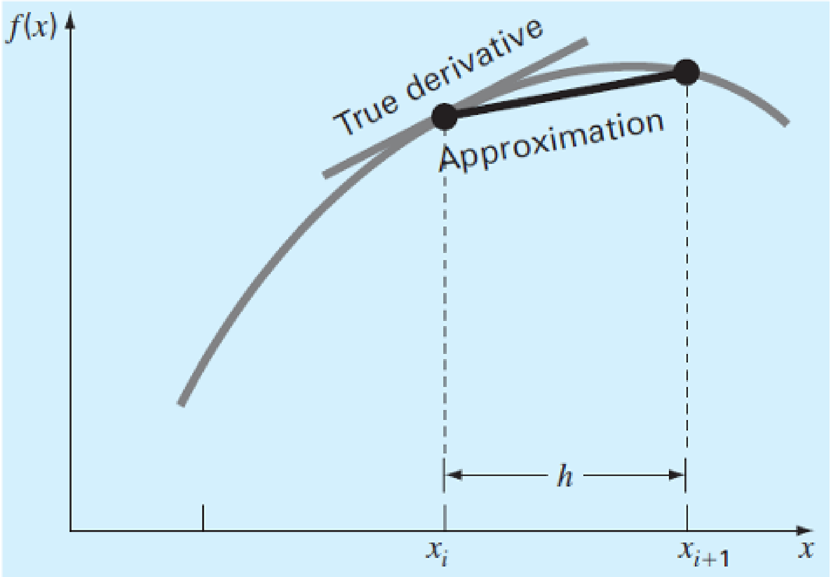
$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{(x_{i+1} - x_i)} - \frac{R_1}{(x_{i+1} - x_i)}$$

- A expressão acima é chamada de *diferença finita dividida* e é representada em geral por

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h)$$

onde *h* é chamado de *tamanho do passo*

e O(h) é a função conhecida como *Big O*

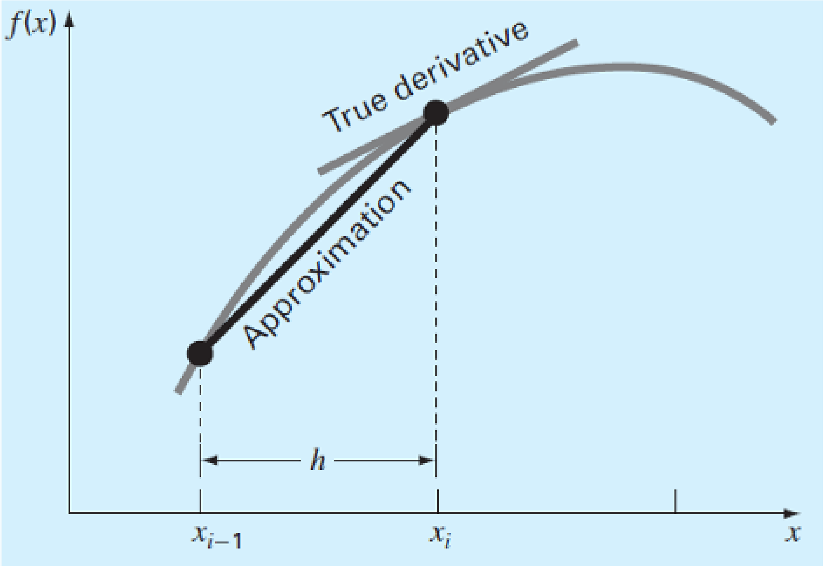


Aproximação da derivada primeira
por diferenças progressiva

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h)$$

Aproximação da derivada primeira
por diferenças regressiva

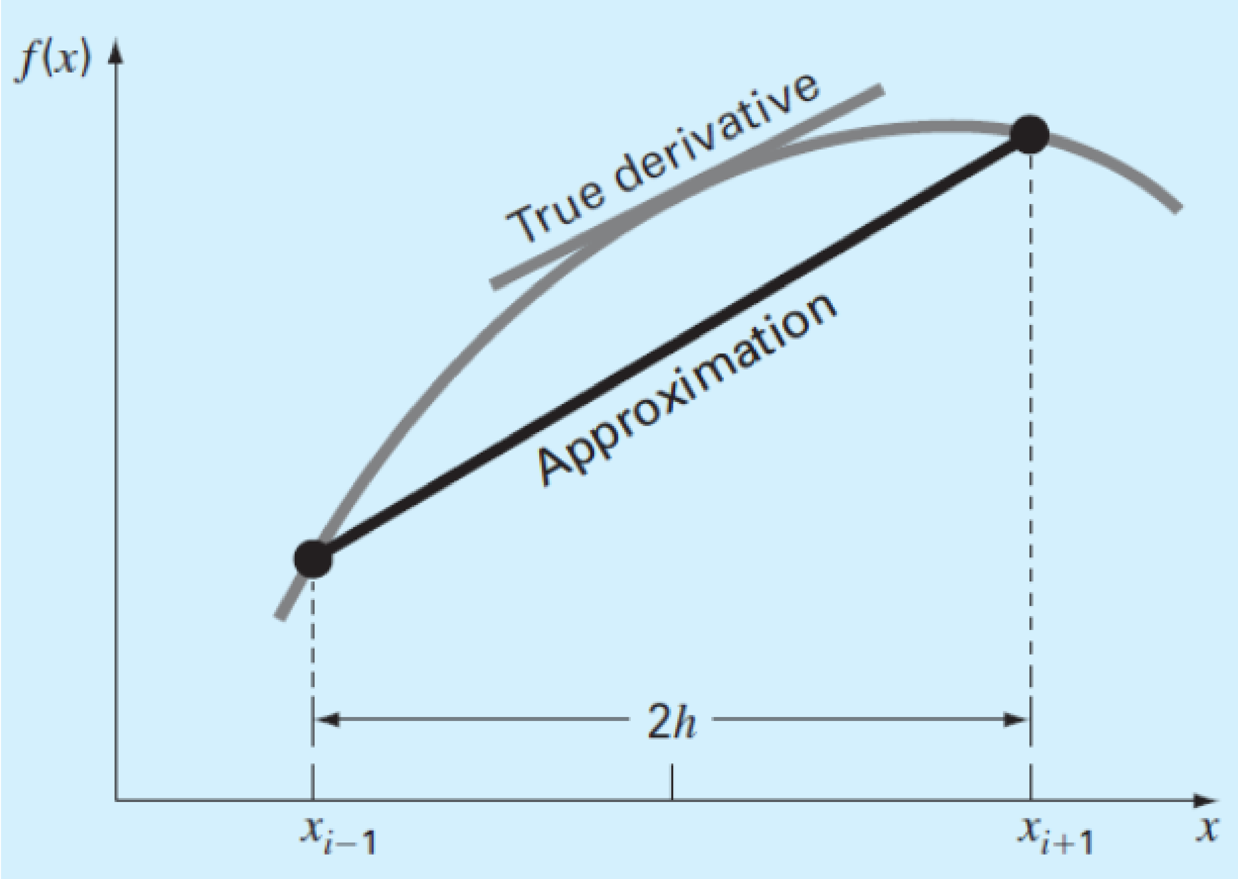
$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + O(h)$$



Séries de Taylor

Derivação numérica

Aproximação da primeira derivada por diferença centrada



Séries de Taylor

Derivação numérica

Séries de Taylor
Aproximação da 1ª derivada por diferença centrada

- Expansão da série de Taylor progressiva: $$f(x_i+h)=f(x_i)+f'(x_i)h+\frac{f''(x_i)}{2!}h^2+\frac{f^{(3)}(x_i)}{3!}h^3+\ldots$$
- Expansão da série de Taylor regressiva: $$f(x_i-h)=f(x_i)-f'(x_i)h+\frac{f''(x_i)}{2!}h^2-\frac{f^{(3)}(x_i)}{3!}h^3+\ldots$$

$-$

$$f(x_i+h)-f(x_i-h)=2f'(x_i)h\ +2\frac{f^{(3)}(x_i)}{3!}h^3+\ldots$$

Séries de Taylor
Aproximação da 1ª derivada por diferença centrada

$$f(x_i+h) - f(x_i-h) = 2f'(x_i)h + 2\frac{f^{(3)}(x_i)}{3!}h^3 + \dots$$

- Como $f(x_i+h) = f(x_{i+1})$ e $f(x_i-h) = f(x_{i-1})$, isolando $f'(x_i)$, obtém-se:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{f^{(3)}(x_i)}{6}h^2 + \dots$$

ou:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - O(h^2)$$

- Nesse caso, o erro de truncamento é da ordem de h^2 em oposição as aproximações regressiva e progressiva que eram da ordem de h, sendo assim mais exata.

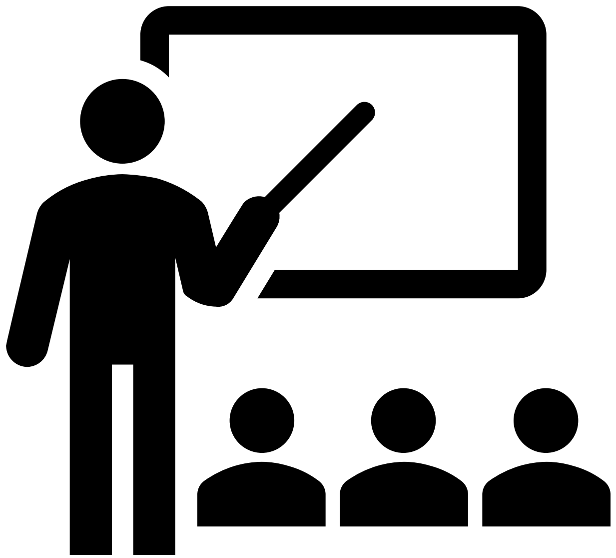
EXEMPLO 3

Use a aproximação por diferenças progressiva e regressiva de O(h) e uma aproximação por diferença centrada de O(h^2) para estimar a derivada primeira de

$$f(x) = 25x^3 - 6x^2 + 7x - 88$$

Avalie a derivada em x = 2 usando o passo de cálculo de 0,5 e 0,25. Compare as suas estimativas com o valor real da derivada. Interprete seus resultados com base no resto da expansão em séries de Taylor.

Calcule manualmente e com o auxilio de um script Scilab



EXEMPLO 3

- Solução:

$$f(x) = 25x^3 - 6x^2 + 7x - 88$$
$$f'(x) = 75x^2 - 12x + 7$$
$$f'(2) = 75 \cdot 2^2 - 12 \cdot 2 + 7 = 283$$

- Passo de cálculo h = 0,5
  - ✓ Diferença progressiva

$$f(2{,}5) = 25 \cdot 2{,}5^3 - 6 \cdot 2{,}5^2 + 7 \cdot 2{,}5 - 88 = 282{,}625$$
$$f(2) = 25 \cdot 2^3 - 6 \cdot 2^2 + 7 \cdot 2 - 88 = 102$$

$$f'(2) = \frac{282{,}625 - 102}{0{,}5} = 361{,}25 \qquad e_t = \left|\frac{283 - 361{,}25}{283}\right| x100 = 27{,}65\%$$

EXEMPLO 3

$$f(2,5) = 25 \cdot 2,5^3 - 6 \cdot 2,5^2 + 7 \cdot 2,5 - 88 = 282,625$$
$$f(2) = 25 \cdot 2^3 - 6 \cdot 2^2 + 7 \cdot 2 - 88 = 102$$

- Passo de cálculo h = 0,5
  - ✓ Diferença regressiva:

$$f(1,5) = 25 \cdot 1,5^3 - 6 \cdot 1,5^2 + 7 \cdot 2,5 - 88 = -6,625$$

$$f'(2) = \frac{102 + 6,625}{0,5} = 217,25 \qquad e_t = \left| \frac{283 - 217,25}{283} \right| x\,100 = 23,23\,\%$$

  - ✓ Diferença centrada

$$f'(2) = \frac{282,625 + 6,625}{2 \cdot 0,5} = 289,25 \quad e_t = \left| \frac{283 - 289,25}{283} \right| x\,100 = 2,21\,\%$$

EXEMPLO 3

- Respostas para passo de cálculo h = 0,25
  - ✓ Diferença progressiva: $f'(2) = 320{,}56248$ $e_t = 13{,}27\%$
  - ✓ Diferença regressiva: $f'(2) = 248{,}5625$ $e_t = 12{,}17\%$
  - ✓ Diferença centrada: $f'(2) = 284{,}56249$ $e_t = 0{,}55\%$
- **Conclusão:** Para ambos os tamanhos de passo, **a aproximação por diferença centrada é mais exata que as outras.**
- Além disso, conforme previsto pela análise por Séries de Taylor, dividir o passo de cálculo por 2, divide o erro por aproximadamente 2 para as diferenças progressiva e regressiva e por 4 na diferença centrada.

EXEMPLO 3
Scilab

```
xi=input('Entre com o valor de xi: ');
h=input('Entre com o passo de cálculo: ');
disp('Entre com os coef da f. polinomial entre ');
vetor=input(' no formato [a0 a1 . . . an]:');
f = poly(vetor,'x','c');
disp( 'f(x)’, f);
flinha = derivat(f);
disp('f´(x)’, flinha);
vreal = horner(flinha,xi);
printf(“Val. real da der. em x = %f é %f\n",xi,vreal);
a = horner(f,xi-h)
b = horner(f,xi)
c = horner(f,xi+h)
diferenca = 'progressiva'
dfdt = (c - b)/h;
```

EXEMPLO 3
Scilab

```
for i=1:3
    printf("Diferença %s :\n",diferenca);
    et = 100*abs((vreal - dfdt)/vreal) ;
    printf("Val. aprox. em x = %f é %f \n",xi,dfdt);
    printf("Com erro relativo de %f %%\n\n",et);
    if i==1 then
        diferenca = 'regressiva';
        dfdt = (b - a)/h;
    elseif i==2 then
        diferenca = 'centrada';
        dfdt = (c - a)/(2*h);
    end
end
```

Séries de Taylor

Aproximação por diferenças finitas de derivadas superiores

- Expansão da série de Taylor progressiva para $f(x_{i+2})$:

$$f(x_{i+2}) = f(x_i) + f'(x_i)2\mathrm{h} + \frac{f''(x_i)}{2!}(2\mathrm{h})^2 + \ldots$$

- Expansão da série de Taylor progressiva:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \ldots \qquad \times\ \mathbf{2}$$

$-$

$$f(x_{i+2}) - 2f(x_{i+1}) = -f(x_i) + f''(x_i)h^2 + \ldots$$

Séries de Taylor
Aproximação por diferenças finitas de derivadas superiores

$$f(x_{i+2}) - 2f(x_{i+1}) = -f(x_i) + \quad f''(x_i)h^2 + \ldots$$

- Truncando-se a partir do termo de 2ª ordem e isolando $f''(x_i)$, obtém-se:

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h)$$

relação chamada de segunda diferença dividida finita progressiva

Séries de Taylor
Aproximação por diferenças finitas de derivadas superiores

- Segunda diferença dividida finita regressiva

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2} + O(h)$$

- Segunda diferença dividida finita centrada

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} + O(h^2)$$

que pode ser expressa pela diferença de 2 diferenças divididas da 1ª derivada

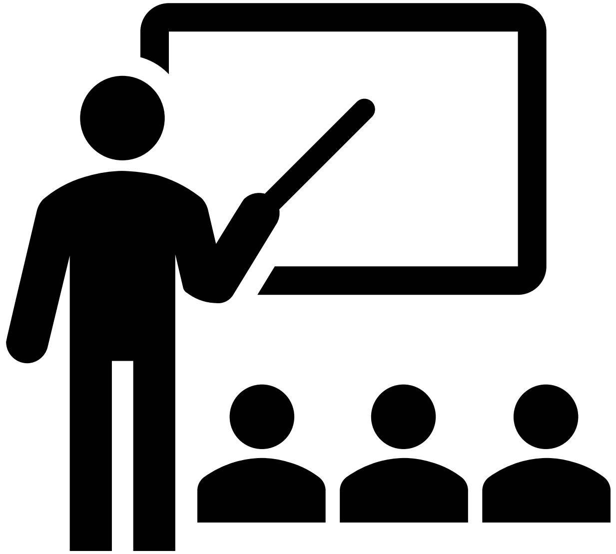
$$f''(x_i) \approx \frac{\dfrac{f(x_{i+1}) - f(x_i)}{h} - \dfrac{f(x_i) - f(x_{i-1})}{h}}{h}$$

EXEMPLO 4

Use uma aproximação por diferença centrada de O(h2) para estimar a derivada segunda da função

$f(x) = -0{,}1x^4 - 0{,}15x^3 - 0{,}5x^2 - 0{,}25x + 1{,}2$

Faça o cálculo em x = 0,5, utilizando os passos de cálculo h = 0,5 e 0,25 e compare as estimativas com o valor verdadeiro da derivada. Interprete os resultados com base no termo do resto da expansão em série de Taylor.



EXEMPLO 3

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} + O(h^2)$$

- Solução:

$$f(x) = -0{,}1x^4 - 0{,}15x^3 - 0{,}5x^2 - 0{,}25x + 1{,}2$$

$$f'(x) = -0{,}4x^3 - 0{,}45x^2 - 1x - 0{,}25$$

$$f''(x) = -1{,}2x^2 - 0{,}9x - 1$$

$$f''(0{,}5) = -1{,}2 \cdot 0{,}5^2 - 0{,}9 \cdot 0{,}5 - 1 = -1{,}75$$

- h = 0,5

$$f(1) = -0{,}1 \cdot 1^4 - 0{,}15 \cdot 1^3 - 0{,}5 \cdot 1^2 - 0{,}25 \cdot 1 + 1{,}2 = 0{,}2$$

$$f(0{,}5) = -0{,}1 \cdot 0{,}5^4 - 0{,}15 \cdot 0{,}5^3 - 0{,}5 \cdot 0{,}5^2 - 0{,}25 \cdot 0{,}5 + 1{,}2 = 0{,}925$$

$$f(0) = 1{,}2$$

$$f''(0{,}5) = \frac{0{,}2 - 2 \cdot 0{,}925 + 1{,}2}{0{,}5^2} = -1{,}8 \qquad e_t = \left| \frac{-1{,}75 + 1{,}8}{-1{,}75} \right| x\,100 = 2{,}86\,\%$$

EXEMPLO 3

- Respostas para passo de cálculo h = 0,25

$$f''(2) = -1{,}7625 \qquad e_t = 0{,}714\%$$

- Conclusão: Conforme previsto pela análise por Séries de Taylor, dividir o passo de cálculo por 2, divide o erro por aproximadamente 4 na diferença centrada.

Bibliografia e crédito das figuras

CHAPRA, Steven. **Applied numerical methods with MATHLAB for engineers and scientists.** McGrawHill, 2012.

CHAPRA, Steven e CANALE, Raymond. **Numerical methods for engineers.** McGrawHill, 2010.