

# FFI 0201 - Introdução à Física Computacional

## Terceiro Projeto

### Instruções

- Crie um diretório **proj3\_#usp** em `/public/IntroFisComp16/projeto3`
- Proteja seu diretório para não ser lido por `g` e `o`
- Deixe no diretório apenas 3 arquivos, de nomes **exerA.f90**, **exerB.f90** e **exerC.f90**
- Os códigos devem seguir rigorosamente os padrões especificados abaixo para **entrada/saída**
- Se deixar de fazer algum exercício não inclua o arquivo correspondente

### Exercícios

A) Considere o problema da **diferenciação numérica**. Dada uma função

$$f_n = f(x_n) = f(x_0 + nh) \quad \text{com } n = 0, \pm 1, \pm 2, \dots$$

pode-se usar a expansão de Taylor ao redor de  $x_0$

$$f(x) = f_0 + (x - x_0)f' + \frac{(x - x_0)^2}{2!}f'' + \frac{(x - x_0)^3}{3!}f''' + \dots$$

onde

$$f' = \left. \frac{df}{dx} \right|_{x=x_0} \quad f'' = \left. \frac{d^2f}{dx^2} \right|_{x=x_0} \quad f''' = \left. \frac{d^3f}{dx^3} \right|_{x=x_0} \quad \dots$$

Usando a relação  $x_n = x_0 + nh$  podemos também escrever

$$f_n = f_0 + nhf' + \frac{n^2h^2}{2!}f'' + \frac{n^3h^3}{3!}f''' + \dots$$

Isso nos permite obter as relações :

- derivada para frente de 2 pontos

$$f' = \frac{f_1 - f_0}{h} + \mathcal{O}(h)$$

- derivada para trás de 2 pontos

$$f' = \frac{f_0 - f_{-1}}{h} + \mathcal{O}(h)$$

- derivada simétrica de 3 pontos

$$f' = \frac{f_1 - f_{-1}}{2h} + \mathcal{O}(h^2)$$

- derivada simétrica de 5 pontos

$$f' = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + \mathcal{O}(h^4)$$

- derivada segunda simétrica de 3 pontos

$$f'' = \frac{f_1 - 2f_0 + f_{-1}}{h^2} + \mathcal{O}(h^2)$$

- derivada segunda simétrica de 5 pontos

$$f'' = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + \mathcal{O}(h^4)$$

- derivada terceira anti-simétrica de 5 pontos

$$f''' = \frac{f_2 - 2f_1 + 2f_{-1} - f_{-2}}{2h^3} + \mathcal{O}(h^2)$$

Escreva um código que forneça os dados da tabela abaixo para as derivadas da função  $f(x) = \exp(4x) \cos^2(x/2)$  para  $x = 1/3$ . Escreva na tabela apenas os desvios (não se preocupe com o sinal do desvio) em relação aos resultados exatos.

Leia a partir de um arquivo de entrada `tabA_in.dat`:

1. o número de valores de  $h$  (na primeira linha)
2. a sequência destes valores de  $h$  (na segunda linha) a serem testados.

O ponto de cálculo da função ( $x = 1/3$ ) deverá ser fixado como parâmetro do programa. A saída deve ser uma tabela, no arquivo `tabA_out.dat`, contendo na primeira coluna o valor de  $h$  e nas colunas seguintes os resultados dos vários desvios, como descrito na tabela abaixo.

**Obs:** a primeira linha do arquivo de saída pode ser reservada para descrição das colunas, desde que seja uma linha só. Deste modo, a tabela para 14 valores de  $h$  poderá ter 15 linhas. Enfim, no terminal, diga em cada caso qual o valor mais apropriado de  $h$  e justifique sua escolha (no formato que achar melhor).

$h$	derivada simétrica 3 pontos	derivada para frente 2 pontos	derivada para trás 2 pontos	derivada simétrica 5 pontos	derivada segunda simétrica 5 pontos	derivada terceira anti-simétrica 5 pontos
0.5						
0.2						
0.1						
0.05						
0.01						
0.005						
0.001						
0.0005						
0.0001						
0.00005						
0.00001						
0.000001						
0.0000001						
0.00000001						

B) Considere o problema da **quadratura numérica**. Dada a integral

$$\int_a^b f(x)dx ,$$

seja  $N = (b - a)/h$  um número inteiro par. Podemos então escrever

$$\int_a^b f(x)dx = \int_a^{a+2h} f(x)dx + \int_{a+2h}^{a+4h} f(x)dx + \int_{a+4h}^{a+6h} f(x)dx + \cdots + \int_{b-2h}^b f(x)dx$$

Para cada intervalo de extensão  $2h$ , i.e. para cada integral do tipo

$$\int_{-h}^h f(x)dx ,$$

podemos usar uma das seguintes regras:

- regra do trapézio

$$\int_{-h}^h f(x)dx = \frac{h}{2} (f_1 + 2f_0 + f_{-1}) + \mathcal{O}(h^3)$$

- regra de Simpson

$$\int_{-h}^h f(x)dx = \frac{h}{3} (f_1 + 4f_0 + f_{-1}) + \mathcal{O}(h^5)$$

onde  $f_{\pm 1} = f(\pm h)$  e  $f_0 = f(0)$ .

No caso de um intervalo de extensão  $4h$ , i.e. para cada integral do tipo

$$\int_{x_0}^{x_4} f(x)dx$$

podemos usar a regra:

- regra de Bode

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45} (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) + \mathcal{O}(h^7)$$

onde  $f_i = f(x_i)$ ,  $x_1 = x_0 + h$ ,  $x_2 = x_0 + 2h$ ,  $x_3 = x_0 + 3h$  e  $x_4 = x_0 + 4h$ . Claramente, neste caso  $N$  deve ser um múltiplo inteiro de 4.

Escreva um código que calcule a integral  $\int_0^1 \exp(x/4) \sin(x/2) dx$  usando os diversos métodos acima e para diferentes números  $N$  de pontos. Estime apenas os desvios (não se preocupe com o sinal dos desvios) em relação ao valor exato.

Leia a partir de um arquivo de entrada `tabB_in.dat`:

1. o número de valores de  $N$  (na primeira linha)
2. a sequência destes valores de  $N$  (na segunda linha) a serem testados.

Seu programa calculará os valores correspondentes de  $h$ . A saída deverá ser uma tabela, no arquivo `tabB_out.dat`, contendo na primeira coluna o valor de  $N$ , na segunda coluna o valor de  $h$  e nas colunas seguintes os resultados dos vários desvios, como descrito na tabela acima.

**Obs:** a primeira linha do arquivo de saída pode ser reservada para descrição das colunas, desde que seja uma linha só. Enfim, no terminal, diga em cada caso qual o valor mais apropriado de  $N$  e justifique sua escolha (no formato que achar melhor).

N	$h$	Regra do trapézio	Regra de Simpson	Regra de Bode
4	0.25000000			
8	0.12500000			
16	0.06250000			
32	0.03125000			
64	0.15625000			
128	0.00781250			
256	0.00399625			
512	0.00195312			
1024	0.00097656			
2048	0.00048828			
4096	0.00024414			

- C) Considere o problema de **encontrar raízes de uma função**  $f(x)$ . Como primeira tentativa, pode-se **chutar** iterativamente valores  $x_i$  ao redor da raiz e verificar quando o valor de  $f(x)$  se aproxima de zero, por exemplo verificando as mudanças de sinal de  $f(x_i)$ . Este método de **busca direta** claramente não será o mais eficiente. No método de **Newton-Raphson** as raízes de  $f(x)$  são calculadas usando as seguintes iterações

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad \text{com } i = 0, 1, 2, \dots$$

Para entender essa relação observe que a equação da tangente à função no ponto  $x_i$  é dada por  $f(x_i) + (x - x_i) f'(x_i)$ . Assim  $x_{i+1}$  representa a intersecção da tangente com o eixo das abscissas. Note que a convergência (ou a não convergência) do método de Newton-Raphson depende da escolha do chute inicial  $x_0$ .

No **método da secante** as iterações podem ser escritas como

$$x_{i+1} = x_i - f(x_i) \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} \quad \text{com } i = 0, 1, 2, \dots$$

Essa relação coincide com a relação do método de Newton-Raphson usando-se para a derivada no ponto  $x_i$  a aproximação

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

Geometricamente, note que

$$f(x_i) + (x - x_i) \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

é a equação da reta que passa nos pontos  $(x_{i-1}, f(x_{i-1}))$  e  $(x_i, f(x_i))$ , i.e. da secante à função  $f(x)$  nesses dois pontos. Assim,  $x_{i+1}$  é a intersecção da secante com o eixo das abscissas. Claramente, o método da secante requer dois valores iniciais, i.e.  $x_0$  e  $x_1$ , que devem ser preferencialmente escolhidos próximos da raiz.

Escreva um código que calcule as (três) raízes (positivas e/ou negativas) da função

$$f(x) = 18x^3 - 9x^2 - 5x + 2,$$

usando os diferentes métodos acima para cálculo de raízes

Leia do terminal o número de iterações (por exemplo, 6) a serem realizadas. A escolha dos valores iniciais será feita dentro do próprio programa. Escreva a saída no arquivo `tabC_out.dat`, contendo 10 colunas, como na tabela abaixo. A primeira linha pode ser usada para descrição das colunas, por exemplo:

```
iter dir1 dir2 dir3 NR1 NR2 NR3 sec1 sec2 sec3
```

**Dica:** para chutar os valores iniciais para busca oriente-se pelo gráfico de  $f(x)$ , obtido com o programa **gnuplot**.

Iteração	Busca Direta <i>r1 r2 r3</i>	Newton-Raphson <i>r1 r2 r3</i>	método da Secante <i>r1 r2 r3</i>
1			
2			
3			
4			
5			
6			