Dr. Ole Peter Smith
Instituto de Matemática e Estatística
Universidade Federal de Goiás
ole@ufg.br - https://olepeter.mat.ufg.br

Curso:	Engenharia de Elementos
Data:	06/02/2024
Disciplina:	Cálculo Numérico
Prova:	I

1. Os coeficientes binomiais são dados por, $p = 0, ..., n$:

$$C_{n,p} = \binom{n}{p} = \frac{n!}{p!(n-p)!}$$

Considere a função (`def`) Python:

```
def Factorial(n):
    if (n==0):     return 1
    elif (n==1):   return 1
    else:          return n*Factorial(n-1)
```

(a) *1pt.* Estabelece uma função (`def`) Python, `Binomial(n,p)`, calculando o valor da coeficiente binomial, $C_{n,p}$.

(b) *1pt.* O Triângulo de Pascal lista os coeficientes binomais, linha número n contendo os coeficientes $B_{n,0}, ..., B_{n,n}$.
Desenvolve um trecho de código, gerando o triângulo de Pascal, incluindo até linha $n = 10$.

(c) *1pt.* Escreve uma função Python, `def fn(x,n)`, retornando a soma:

$$f_n(x) = \sum_{p=0}^{n} \frac{x^n}{n!}$$

Para n 'suficientemente grande', $f_n(x)$ aproxima-se a função exponencial, e^x. Escreve um código gerando uma tabela da seguinte forma:

n	$f_n(x)$	$\lvert e^x - f_n(x)\rvert$	$\lvert (e^x - f_n(x))/e^x\rvert$
1	...	...	...
...	...	...	...
10	...	...	...

Gera a tabela para $x = 1, ..., 5$. Para cada destes valores de x, quantos termos é preciso incluir, para o erro relativo ser menor do que $\varepsilon = 1.0E-4$?

2. Estamos procurendo um polinômio interpolador:

$$P(x) = p_4x^4 + p_3x^3 + p_2x^2 + p_1x + p_0,$$

contendo os pontos:

i	0	1	2	3	4
x_i	-2	-1	0	1	2
y_i	1	-3	2	-1	4

Ou seja:

$$y_i = P(x_i), \ i = 0, ..., 4$$

(a) *2pts.* Encontre uma matriz, $\underline{\underline{\mathbf{A}}}$, tal que: $\underline{\underline{\mathbf{A}}}\ \underline{\mathbf{p}} = \underline{\mathbf{y}}$ e resolve o sistema linear do item anterior, usando o método de Gauss com pivotação parcial.

(b) *2pts.* Encontre os polinômios de Lagrange, $H_0(x), ..., H_4(x)$, associado aos valores $x_0, ..., x_4$ e calcule o polinômio interpolador, $P(x) = \sum_{k=0}^{4} y_k H_k(x)$, verificando que coincide com o polinômio obtido no item (a). Verifique também, que: $y_i = P(x_i), \ i = 0, ..., 4$.

(c) *1pt.* Encontre a matrix inversa, $\underline{\underline{\mathbf{A}}}^{-1}$, e o resíduo:

$$r = \frac{||\underline{\underline{\mathbf{A}}}^{-1}\underline{\underline{\mathbf{A}}} - \underline{\underline{\mathbf{I}}}||}{||\underline{\underline{\mathbf{A}}}||}$$

Hint! Em Python, pode-se representar um polinômio de grau n:

$$P(x) = p_nx^n + p_2x^2 + p_1x + p_0,$$

com uma lista (`list`): [p_0,p_1,...,p_n,]. Por exemplo: $P(x) = 3$ é representado por [3], $P(x) = x^2 + 2x + 3$ é representado por [3,2,1].

Dr. Ole Peter Smith
Instituto de Matemática e Estatística
Universidade Federal de Goiás
ole@ufg.br - https://olepeter.mat.ufg.br

3. Considere a função e seu primitivo:

$$f(x) = x^{3/2}, \qquad F(x) = \int_0^x f(t)\,dt$$

   (a) *1pt.* Escreve funções (def) Python, retornando o valor de $f(x)$, respectivamente $F(x)$.

   (b) *2pts.* Para $x = 3$, estime $F(x)$ usando o Método de Trapézios e de 1/3 de Simpson, calculando os erros absolutos e relativos com $n = 10, 20, ..., 100$ intervalos.

Instruções:

- Salve suas respostas numa pasta nova, em arquivos respostas: 01.py, 02.py e 03.py.
- Responstas literais, deviam ser incluídas no código como comentários (#) ou `print`'s.
- Pode-se utilizar os códigos disponibilizado no site abaixo, salvos como arquivos auxiliares (incluídos via `import`'s) ou, alternativamente, copiando os códigos relevantes para os arquivos respostas.
- Entregar um arquivo ZIP de todo os arquivos deste pasta por email: ole@ufg.br.
- Incluir como assunto do email: CN, P1: seu nome completo.

http://www.olesmith.com.br/SmtC?ModuleName=Texts&Action=Root&Text=356

Destacamos também o link:
http://www.olesmith.com.br/SmtC?ModuleName=Texts&Action=Codes&Text=356