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

Curso:	Estatística
Data:	17/07/2018
Disciplina:	Cálculo Numúmerico
Pasta Resposta:	Prova_II
Prova:	Substituitiva

1. Arquivo resposta: N01.py. Considere a função:

$$f(x) = \cos x - \frac{1}{3}$$

Estamos a procura da raíz, $f(x_0) = 0$, no intervalo $I_0 =]0, \pi[$. Temos:

$$x_0 = \text{Arccos}\, x$$

(a) *1 pt*. Usando como intervalo inicial I_0, encontre estimativos para a raíz x_0, usando o algoritmo de Bisecção, e $\varepsilon = 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}$, gerando uma tabela exibindo a ε, o número de iterações (i), o valor analítica, x_a, o valor estimado, x_n, alem dos erros absolutos e relativos:

ε	i	x_a	x_n	r_a	r_r
1.00e-03	9	1.230959417341	1.230252591884	7.068255e-04	5.742070e-04
1.00e-04	11	1.230959417341	1.231019582278	6.016494e-05	4.887646e-05
1.00e-05	17	1.230959417341	1.230959661154	2.438129e-07	1.980674e-07
1.00e-06	17	1.230959417341	1.230959661154	2.438129e-07	1.980674e-07
1.00e-07	23	1.230959417341	1.230959473900	5.655941e-08	4.594742e-08
1.00e-08	25	1.230959417341	1.230959427087	9.746033e-09	7.917429e-09

(b) *1 pt*. Repita a questão anterior, mas usando o algoritmo de Posição Falsa.

(c) *1 pt*. Repita a questão anterior, mas usando o algoritmo de Newton Raphson com ponto inicial $\pi/4$.

2. Arquivo resposta: N02.py.

Para as Séries de Potência ou Polinômios de Taylor abaixo, encontramos as seguintes funções soma:

$$(1): \quad \sum_1^{+\infty} \frac{2^n}{n} x^n = -\log(1-2x), \ |x| < \frac{1}{2}$$

$$(2): \quad \sum_1^{+\infty} \frac{1}{(n+3)!} x^n = \frac{1}{x^3}\left(e^x - \frac{1}{6}x^3 - \frac{1}{2}x^2 - x - 1\right), \ |x| \in \mathbb{R}$$

$$(3): \quad \sum_1^{+\infty} (-1)^n x^{2n} = -\frac{x^2}{1+x^2}, \ |x| < 1$$

Gostariamos verificar numericamente estes resultados. Veja o código abaixo:

```
def Sum(a,n,x):
    summ=0.0

    xi=x
    for i in range(1,n):
        summ+=a(i)*xi
        xi*=x

    return summ
```

```
def Factorial(n):
    if (n<2): return 1.0

    return n*Factorial(n-1)
```

(a) *1 pt*. Implemente como funções Python as funções soma acima (`def f_1(x)`, etc).

(b) *1 pt*. Implemente como funções Python as coeficientes dos polinômios de Taylor (`def a_1(n)`, etc).

(c) *2 pts*. Para cada uma dos polinômios acima e $x = 0.25$, calcule os valores do polinômio de Taylor de ordem $n = 1, 2, 5, 10, 20, 50, 100$. Para cada um deles, gerar uma tabela tendo colunas: x, a ordem do polinômio (n), o valor das funções soma ($f_n(x)$), o valor do polinômio de Taylor ($P_n(x)$), e os erros absolutos, relativos e percentuais. Veja a tabela abaixo.

ματεματικα
Made in LaTeX

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

t	$x_i(t)$	$x'_{ana}(t)$	$x'_{num}(t)$	r_a	r_r	$x''_{ana}(t)$	$x''_{num}(t)$	r_a	r_r
1.000000	0.841471	1.381773	1.381773	1.996181e-13	1.444652e-13	0.239134	0.239142	8.412576e-06	3.517939e-05
2.000000	1.818595	0.077004	0.077004	2.461331e-11	3.196378e-10	-2.650889	-2.650880	9.010848e-06	3.399180e-06
5.000000	-4.794621	0.459387	0.459387	6.505962e-11	1.416228e-10	5.361946	5.361933	1.262451e-05	2.354465e-06
10.000000	-5.440211	-8.934736	-8.934736	6.914931e-09	7.739379e-10	3.762068	3.761436	6.324433e-04	1.681105e-04

3. Arquivo resposta: N03.py.

   Considere a matriz:

$$\underline{\underline{\mathbf{A}}} = \begin{pmatrix} -1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -1 \end{pmatrix}$$

   (a) *1pt.* Resolver as equações lineares, $\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}_i = \underline{\mathbf{e}}_i$, para os lados direitos:

$$\underline{\mathbf{e}}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \underline{\mathbf{e}}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \underline{\mathbf{e}}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \underline{\mathbf{e}}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

   Exibe (print) *somente* os vetores, x_i!

   Hint! Lembre-se de chamar `Matrix_Copy` dentro do laço.

   (b) *1pt.* Encontrar a matriz inversa, $\underline{\underline{\mathbf{A}}}^{-1}$.
   Exibe (print) *somente* a matriz inversa!

   (c) *1pt.* Implemente um trecho de código python, que calcula as matrices $\underline{\underline{\mathbf{A}}}^5$, $\underline{\underline{\mathbf{A}}}^{10}$ e $\underline{\underline{\mathbf{A}}}^{15}$.
   Exibe (print) *somente* as matrices citadas!

4. Arquivo resposta: N04.py.

   Considere as funções:

$$x_1(t) = t\cos t, \qquad x_1(t) = t\sin t, \qquad x_3(t) = t^4.$$

   (a) *1 pt.* Implementar como funções Python, as funções $x_1(t)$, $x_2(t)$ e $x_3(t)$. Implemente também suas primeira e segunda derivadas.

   (b) *2 pts.* Para $t = 1, 2, 5, 10$, calcular aproximações numéricas para as primeira e segunda derivadas das no item anterior. Calcule, para ambas, os resíduos absoluto e relativo, gerando uma tabela como abaixo:

t	$x_1(t)$	$x'_{ana}(t)$	$x'_{num}(t)$	r_a	r_r	$x''_{ana}(t)$	$x''_{num}(t)$	r_a	r_r
1.000000	0.841471	1.381773	1.381773	1.996181e-13	1.444652e-13	0.239134	0.239142	8.412576e-06	3.517939e-05
2.000000	1.818595	0.077004	0.077004	2.461331e-11	3.196378e-10	-2.650889	-2.650880	9.010848e-06	3.399180e-06
5.000000	-4.794621	0.459387	0.459387	6.505962e-11	1.416228e-10	5.361946	5.361933	1.262451e-05	2.354465e-06
1.000000	0.841471	1.381773	1.381773	1.996181e-13	1.444652e-13	0.239134	0.239142	8.412576e-06	3.517939e-05

Em seus arquivos respostas, exibir somente os dados solicitados!

Somente será considerado provas de quem assinou a lista de frequência!!!

Responder no máximo 10.0 pontos!

ματεματικα
Made in LaTeX