Pontificia Universidad
JAVERIANA
Bogotá

Facultad de Ciencias
Departamento de Matemáticas
eherrera@javeriana.edu.co

Análisis Numérico
Primer parcial
Agosto de 2021

Recomendaciones

• No se resuelven preguntas del contenido a evaluar, toda respuesta debe estar sustentada y justificada.
• Cada estudiante debe subir a su repositorio un documento en pdf con la solución, incluidas gráficas y tablas (recuerde que debe aparecer la fecha y hora en los pantallazos o ImpPt) y debe estar los archivos: R y/o .py (código) en un archivo aparte también en el repositorio.
• Tiene plazo para subir su solución en la carpeta parciales/parcial1 hasta 9:10 am (hora local)-agosto 27 de 2021

Inmediatamente que suba su solución, debe enviar un correo desde su cuenta de la javeriana a: eherrera@javeriana.edu.co con el link del repositorio (esto es obligatorio)

1. En cada uno de los siguientes ejercicios debe implementar en R y/o Python el algoritmo asignado para resolver el problema (incluya en el código comentarios), generar una tabla con los errores relativos, determinar el número de iteraciones realizadas; una gráfica que evidencie el tipo de convergencia del método utilizado y si esta no es cuadrática aplique un método para acelerar la convergencia, para resolver el problema:
Encuentre la raíz de $f(x)$ aplicando el método asignado con la tolerancia deseada y demuestre numéricamente que $f(x) = x^3 + 2x + k$ cruza el eje x exactamente una vez, independientemente del valor de la constante *k*. Si a =Ultimo digito de su documento de identificación (Cédula, Tarjeta de Identidad o Cédula de extranjería) entonces:
+$\sqrt{a+2}$; ii. a -1/6.

a. Método de Müller; $TOL < 10^{-16}$; $k = \sqrt{a+2}$
b. Método de la secante; $TOL < 10^{-32}$; $k = a$ -1/6
c. Método de la posición falsa; $TOL < 10^{-12}$; $k = \sqrt{a+12}$
d. Método del punto fijo; $TOL < 10^{-10}$; $k = \sqrt{a+2}$
e. Método de Newton-Raphson; $TOL < 10^{-8}$; $k = a$ -1/3
f. Newton-Raphson relajado; $TOL < 10^{-16}$; $k = \sqrt{a+2}$

2. En los siguientes ejercicios aplicar Taylor para aproximar $f(x)$, con un polinomio P_i alrededor de x_0; calcular el error hacia adelante y hacia atrás para cada x^*; encuentre un límite superior para el error $|f(x^*) - P_i(x^*)|$ y compárela con el error real; realizar una gráfica que muestre el polinomio de aproximación y la función. Implemente en R y/o Python, con una precisión de 10^{-8}.

a. $f(x) = \ln(x^2+2); x_0 = 1; P_3(1.5111)$
b. $f(x) = (x-1)lnx; x_0 = 1; P_3(1.111)$
c. $f(x) = xe^{x^2}; x_0 = 0; P_3(1/3)$
d. $f(x) = 2\cos(2x) - (x-2)^2; x_0 = 0; P_3(\pi/6)$
e. $f(x) = \sqrt{x+1}; x_0 = 1; P_4(3/2)$

3. Para cada una de las siguientes ecuaciones, determine un intervalo [a, b] en el que la iteración de **punto fijo** converge. Estime el número de iteraciones necesarias para obtener aproximaciones precisas dentro de 10^{-5} y realice los cálculos. Implemente en R y/o Python
a. $2 + sinx - x = 0$
b. $x - cosx = 0$
c. $x^3 - 2x + 5 = 0$
d. $3x + e^x - x^2 = -2$.

4. Sea $f(x) = e^x - x - 1$ (a) Muestre numéricamente que f tiene un cero de multiplicidad 2. (b) Utilizando el método de Newton con $x_0 = 1$demuestre numéricamente que converge a la raíz, pero no cuadráticamente. Implemente en R y/o Python
5. Sea $f(x) = e^x - x - 1$ realice una modificación del método de Newton con $x_0 = 1$ para que converja a la raíz cuadráticamente. Implemente en R y/o Python.
6. Dado $P(x) = 1 - x + x^2 - x^3 + \cdots + x^{98} - x^{99}$ en $x = 1.0001$; encuentre una expresión más sencilla y utilícela para calcular numéricamente el error de la multiplicación anidada para el valor de x ; realice un gráfico donde se muestra la diferencia alrededor de $x = 1.0001$ Implemente en R y/o Python