Instituto Tecnológico de Costa Rica
Ingeniería en Computadores
CE 3102: Análisis Numérico para Ingeniería

Semestre I-2020
Puntaje Total: 100 pts
Tiempo: 3 horas

EXAMEN DE PROGRAMACIÓN 1

Instrucciones Generales

- **Este examen se realizará en un archivo con extensión `ipynb`, utilizando el programa Jupyter**. El nombre del archivo debe ser `Apellido1-Apellido2-Nombre-ANPI-EP1`. Dentro de este archivo, deben existir dos partes bien identificadas con nombre `Pregunta1` y `Pregunta2`, donde cada parte dará solución a cada una de las preguntas de este examen. Este archivo debe contener todas las funciones creadas con el propósito de responder a cada pregunta (funciones y *script*).
- Debe enviar el archivo al correo `jusoto@tec.ac.cr`.
- El asunto del correo deber ser `Examen Programación 1 - I 2020`. En el cuerpo del correo debe indicar su nombre completo y número de carnet.
- Fecha y hora límite de entrega: **Lunes 15 de junio del 2020 a las 5:30 pm.** No se aceptarán exámenes entregados después de la fecha y hora indicada.

Pregunta 1 - Método de Newton-Steffensen de Tercer Orden

- Esta pregunta consiste en la implementación de un método iterativo para aproximar una solución de una ecuación no lineal de la forma $f(x) = 0$. Este método se explica y desarrolla en el artículo científico *A composite third order Newton-Steffensen method for solving nonlinear equations* elaborado por el investigador J.R. Sharma (ver método NSM en la ecuación (9)).
- **Esta pregunta desarrollará usando el núcleo de GNU Octave.**

Pregunta

1. [**Valor: 50 pts**] Implementar un *script* que produzca una tabla similar a la Tabla 1 presentada en el artículo científico mencionado antes (ver Figura 1). Para eso, deben utilizar el comando `dataframe` (ver archivo `ejemplo_tabla.m` como ejemplo para crear una tabla en GNU OCTAVE). En ese caso, también debe implementar los método de Newton (NM) y el método de Steffensen (SM) (ver ecuación (2) del artículo científico) para comparar el método desarrollado en el artículo científico. Utilice una condición de parada de $|f(x_k)| \leq 10^{-10}$. Además, cada método implementado debe detenerse cuando el denominador respectivo sea cero (para evitar la división entre cero). En este caso, el método muestra los resultados obtenidos hasta ese punto.

Table 1
Numerical Examples

$f(x)$	x_0	Root (α)	Iteration (n) by		
			NSM	NM	SM
$\tan^{-1}(x)$	2	0.00000000000000	4	Failure	Failure
$\sin(x) - x/2$	2	1.89549426703398	4	Not converges to required root	4
$10x\exp(-x^2) - 1$	1	1.67963061042845	3	5	Failure
$x^6 - 36x^5 + 450x^4 - 2400x^3 + 5400x^2 - 4320x + 720$	15	15.98287398060170	4	7	Failure
$x\log 10(x) - 1.2$	2	2.74064609597369	3	5	5

Figura 1: Imagen de la Tabla 1 del artículo científico *A composite third order Newton-Steffensen method for solving nonlinear equations* (x_0 es el valor inicial).

Pregunta 2 - Algoritmo de Thomas

- Esta pregunta consiste en el desarrollo del algoritmo de Thomas para resolver un sistema de ecuaciones tridiagonal.
- **Esta pregunta se desarrollará usando el núcleo de Python.**

Matriz Tridiagonal y Algoritmo de Thomas

Matriz Tridiagonal: Una matriz se llama matriz tridiagonal si todos los elementos que están fuera de la diagonal principal y las diagonales adyacentes por encima y por debajo de esta, son igual a cero. Por ejemplo

$$A = \begin{bmatrix} 1 & 4 & 0 & 0 & 0 \\ 3 & 4 & 1 & 0 & 0 \\ 0 & 2 & 3 & 1 & 0 \\ 0 & 0 & 1 & 3 & 4 \\ 0 & 0 & 0 & 3 & 4 \end{bmatrix}.$$

Algoritmo de Thomas: El algoritmo para matrices tridiagonales o algoritmo de Thomas es un algoritmo del álgebra lineal numérica para resolver matrices tridiagonales de forma eficiente. Considere el sistema de ecuaciones

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

El primer paso del método es modificar los coeficientes como sigue:

$$p_i = \begin{cases} \dfrac{c_i}{b_i} & ; \quad i = 1 \\ \dfrac{c_i}{b_i - p_{i-1} \cdot a_i} & ; \quad i = 2, 3, \dots, n-1 \end{cases} \qquad \text{y} \qquad q_i = \begin{cases} \dfrac{d_i}{b_i} & ; \quad i = 1 \\ \dfrac{d_i - q_{i-1} \cdot a_i}{b_i - p_{i-1} \cdot a_i} & ; \quad i = 2, 3, \dots, n. \end{cases}$$

Luego, la solución del sistema se obtiene a partir de las siguiente fórmula:

$$x_n = q_n \quad \text{y} \quad x_i = q_i - p_i \cdot x_{i+1}, \qquad \text{para } i = n-1, n-2, \dots, 1.$$

Pregunta

1. [**Valor: 50 pts**] Implementar un *script* el Algoritmo de Thomas para resolver el sistema de ecuaciones $Ax = d$, donde $A \in \mathbb{R}^{100\times 100}$ y $d \in \mathbb{R}^{100}$ tal que

$$A = \begin{bmatrix} 5 & 1 & & & \\ 1 & 5 & 1 & & \\ & 1 & 5 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 5 \end{bmatrix} \quad y \quad d = \begin{bmatrix} -12 \\ -14 \\ -14 \\ \vdots \\ -14 \\ -14 \\ -12 \end{bmatrix}$$

Observación: El nombre de las variables a utilizar deben coincidir con las letras que se utilizan en la fórmula matemática del algoritmo de Thomas presentadas en este examen. **Si no se cumple esta indicación, se restarán 30 puntos del puntaje total del examen.**