Instituto Tecnológico de Costa Rica
Ingeniería en Computadores
Puntaje Total: 100 puntos

CE-1111: Análisis Numéricos para Ingeniería
Semestre: I - 2025
Valor Porcentual: 8 %

Tarea GNU Octave y Python

Instrucciones generales

- La tarea se realiza en grupos de máximo 4 personas. Cada grupo debe escribir el nombre de los integrantes del grupo en la siguiente dirección electrónica:

  `https://docs.google.com/spreadsheets/d/1ZROgDcrQyN1mqv5ViCzNHPbhVfbgnh94`

  El número del grupo está indicado en la primera columna del documento.

- Todos los archivos de esta tarea se encuentran en la carpeta de *One Drive* del curso.

- Los archivos computacionales implementados en GNU Octave y Python deben estar correctamente comentados. Por cada archivo que no este documentado correctamente, se restaran 5 puntos de la nota final. **Si alguna función o archivo computacional está incompleto o genera error al momento de compilar, entonces pierde el 75% del puntaje de la pregunta asignada.**

- Los archivos que dan solución a la tarea deben estar en una carpeta principal con nombre `Tarea GNU Octave y Python - Grupo #`, donde # es el número de cada grupo. Dentro de esta carpeta debe existir tres carpetas con nombres `Parte 1`, `Parte 2` y `Parte 3`. En cada una de estas carpetas estarán todos los archivos necesarios para el desarrollo de las preguntas mencionadas anteriormente.

- La solución de la tarea que se encuentra en la carpeta `Tarea GNU Octave y Python - Grupo #` debe comprimirse en un archivo `.zip` y subirlo al formulario que se encuentra en el siguiente enlace:

  `https://forms.gle/r12f5gBqUy7wrNq96`

  Observaciones:

  – Se necesita tener una cuenta de **gmail** para llenar el formulario.

- **Fecha y hora máxima de entrega: Domingo 16 de marzo del 2023, a las 11:59 pm**

- Las entregas tardías se realizarán al correo penalizarán con una reducción de la nota obtenida con un 10% por cada hora de atraso. A las tareas que excedan el plazo de entrega en 10 horas o más después de la hora límite, se les asignará la nota de 0. **SOLAMENTE las entregas tardías se realizarán al correo** `jusoto@tec.ac.cr`.

Parte 1: Paquete `FunTras` en Python

Descripción General

- **Definición**: Una función trascendente es una función que no satisface una ecuación polinomial. Ejemplo de funciones trascendentes son e^x, $\ln(x)$ y $\sin(x)$.
- Cada grupo debe desarrollar una aplicación en **Python** que permita aproximar el valor numérico de un conjunto de funciones trascendentes de variable real utilizando **únicamente** las operaciones de suma (`+`), resta (`-`), multiplicación (`*`) y potencia de exponente entero positivo ($\wedge$). **No pueden usar la división (`/`).**
- En esta parte de la tarea se evaluarán los siguientes atributos de egresado:
  - **Investigación**

Pregunta

1. [**Valor: 50 puntos**] Implemente computacionalmente en Python las funciones transcendentes que se encuentran en la siguiente tabla.

Función $f(x)$	Comando en Python	Función $f(x)$	Comando en Python
x^{-1}	`div_t(x)`	e^x	`exp_t(x)`
$\sin(x)$	`sin_t(x)`	$\cos(x)$	`cos_t(x)`
$\tan(x)$	`tan_t(x)`	$\ln(x)$	`ln_t(x)`
$\log_y(x)$	`log_t(x,y)`	x^y	`power_t(x,y)`
$\sinh(x)$	`sinh_t(x)`	$\cosh(x)$	`cosh_t(x)`
$\tanh(x)$	`tanh_t(x)`	$\sqrt{x}$	`sqrt_t(x)`
$\sqrt[y]{x}$	`root_t(x,y)`	$\sin^{-1}(x)$	`asin_t(x)`
$\tan^{-1}(x)$	`atan_t(x)`	$\cos^{-1}(x)$	`acos_t(x)`
$\sec(x)$	`sec_t(x)`	$\cot(x)$	`cot_t(x)`
$\csc(x)$	`csc_t(x)`	$n!$	`factorial(n)`

   - Para realizar dicha implementación, deben leer el documento `fun_tras.pdf` que se encuentra en la carpeta de *One Drive*. Este documento contiene los métodos iterativos que deben implementar para aproximar las funciones que se encuentran en la tabla anterior.
   - Todas las funciones debe estar implementada en un archivo con el nombre `pregunta1.py`.
   - Para su implementación, cada método iterativo debe usar una tolerancia de 10^{-8}, además de una cantidad máxima de 2500 iteraciones.
   - Algunas de las funciones que se encuentran en la tabla no están en el documento `fun_tras.pdf`. Para la implementación de estas funciones, utilice propiedades matemáticas para re-escribir dichas funciones en términos de las funciones que se encuentran en el documento `fun_tras.pdf` (por ejemplo, $\cos^{-1}(x) = \pi/2 - \sin^{-1}(x)$ y $\log_a(x) = \ln(x)/\ln(a)$).

- Para calcular $\sqrt[y]{x}$ debe considerar dos casos. Si y es entero positivo, entonces utiliza el método iterativo para raíces que se presenta en el documento `funtras.pdf`. Si y no es entero, entonces utilizar el comando `power_t(x,1/y)`.
- Cuando se realice alguna división a/b en los métodos presentados en el documento `fun_tras.pdf`, no se debe utilizar el comando de división de Python, y en su lugar utilizar el comando `div_t`, es decir, realizar la operación `a·div_t(b)`.
- Cada una de las funciones debe verificar su dominio máximo. En el caso de que el parámetro inicial no se encuentra en el dominio, la función debe enviar un mensaje de error (por ejemplo, la función `sqrt_t(x)` solo debe aceptar parámetros mayores o iguales a 0).
- Algunas de las funciones que se encuentran en el documento `fun_tras.pdf` utilizan la función factorial. Para eso, deben utilizar la función factorial implementada por ustedes.
- Utilizando las funciones implementadas, desarrolle un *script* con nombre `test_funtras.py` que realice la operación

$$\frac{\sqrt[3]{\cos\left(\frac{3}{7}\right)+\ln(2)}}{\sinh(\sqrt{2})}+\tan^{-1}(e^{-1}).$$

Parte 2: Matriz Pentadiagonal en GNU Octave

Descripción General

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

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

- Esta tarea consiste en resolver sistemas de ecuaciones utilizando **GNU Octave**, además de crear matrices pentadiagonales.
- En esta parte de la tarea se evaluarán los siguientes atributos de egresado:
  - **Investigación**

Pregunta

1. [**Valor 25 puntos**]: Implemente en GNU Octave una función con la siguiente estructura:

```
A=pentadiagonal(m,a,b,c,d,e)
```

donde el parámetro de salida A es una matriz pentadiagonal de tamaño $m \times m$, que tiene la siguiente estructura

$$A = \begin{pmatrix} a_1 & c_1 & e_1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ b_1 & a_2 & c_2 & e_2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ d_1 & b_2 & a_3 & c_3 & e_3 & 0 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & b_3 & a_4 & c_4 & e_4 & 0 & \cdots & 0 & 0 \\ 0 & 0 & d_3 & b_4 & a_5 & c_5 & e_5 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & d_{m-5} & b_{m-4} & a_{m-3} & c_{m-3} & e_{m-3} & 0 \\ 0 & 0 & 0 & \cdots & 0 & d_{m-4} & b_{m-3} & a_{m-2} & c_{m-2} & e_{m-2} \\ 0 & 0 & 0 & \cdots & 0 & 0 & d_{m-3} & b_{m-2} & a_{m-1} & c_{m-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & d_{m-2} & b_{m-1} & a_m \end{pmatrix}, \tag{1}$$

y los parámetros iniciales son:

- `m` $=$ un número entero positivo, donde $m \geq 5$.
- `a` $=$ es un vector de tamaño $m \times 1$.
- `b, c` $=$ son dos vectores de tamaño $(m-1) \times 1$.
- `d, e` $=$ son dos vectores de tamaño $(m-2) \times 1$.

Lueg,o desarrolle un *script* en Octave con nombre `pregunta2.m` para resolver el sistema de ecuaciones

$$Ax = h,$$

donde A es una matriz pentadiagonal como se muestra en la ecuación (1). Para eso, utilice los siguientes parámetros:

- `m` $= 2500$.
- `a` $=$ es un vector de tamaño $m \times 1$ tal que $a(i) = 2i$, para todo $i = 1, ..., m$.
- `b` $=$ es un vector de tamaño $m - 1 \times 1$ tal que $b(i) = -(i+1)/3$, para todo $i = 1, ..., m-1$.
- `c` $=$ es un vector de tamaño $m - 1 \times 1$ tal que $c(i) = i/3$, para todo $i = 1, ..., m-1$.
- `d` $=$ es un vector de tamaño $m - 2 \times 1$ tal que $d(i) = -(i+2)/2$, para todo $i = 1, ..., m-2$.
- `e` $=$ es un vector de tamaño $m - 2 \times 1$ tal que $e(i) = i/2$, para todo $i = 1, ..., m-2$.
- h $=$ es un vector de tamaño $m \times 1$ tal que $h(i) = 2i - 5$, para todo $i = 1, ..., m$.

Nota: No se debe imprimir el vector resultante, para eso, impriman el error definido por

$$error = \|Ax - h\|_2,$$

donde $\|\cdot\|_2$ es la norma euclideana.

Parte 3: Análisis de una Función en Python

Descripción General

- Esta tarea consiste en utilizar la librería `sympy` de **Python** para hacer el análisis de funciones reales.
- En esta parte de la tarea se evaluarán los siguientes atributos de egresado:
  - **Investigación**

Pregunta

1. [**Valor 25 puntos**]: En un script con el nombre `pregunta3.m`, y utilizando la librería *sympy* de Python, calcule la siguiente información de la función $f(x) = \dfrac{x^3 - 3x^2 + 3x - 1}{x^2 - 2x}$:

   (a) Dominio de la función.

   (b) Intersecciones en los ejes x y y.

   (c) Asíntotas verticales, horizontales y oblicuas.

   (d) Primera y segunda derivada.

   (e) Gráfica de las funciones f, f' y f''.

   (f) Intervalos donde la función es creciente y decreciente.

   (g) Intervalos donde la función es concava hacia abajo y concava hacia arriba.

   Toda la información anterior debe estar documentada. Para eso, entregue un documento con extensión `.pdf` con la infomación solicitada. El nombre del documento debe ser `pregunta3.py`.