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

CE-3102: Análisis Numéricos para Ingeniería
Semestre: I - 2020
Valor Porcentual: 8 %

Tarea 3

Instrucciones generales

- La tarea se realiza en grupos de 3 ó 4 personas.
- Los archivos de esta tarea se encuentran en el TEC- Digital, en la sección de `Documentos`, en la carpeta `Tareas->Tarea 3`.
- Los archivos computacionales implementados en GNU Octave y Pyhton deben estar correctamente documentados. 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.**

Parte 1: Paquete Computacional *NumInt* en GNU Octave

Descripción General

Esta parte de la tarea consiste en desarrollar un paquete computacional en GNU Octave que permita aproximar el valor numérico de la integra definida

$$\int_a^b f(x)\,dx, \tag{1}$$

donde $f : A \to \mathbb{R}$, $A \subseteq \mathbb{R}$, es una función continua en A y $[a, b] \in A$.

Preguntas

1. [**Valor: 16 puntos**] Implemente computacionalmente en GNU Octave los siguientes métodos para aproximar el valor de la integral definida expresada en (1).
   - Regla del Trapecio:
     - Nombre de la función: `trapecio`.
     - Parámetros iniciales: `f` = función $f(x)$; `a, b` = intervalo $[a, b]$.
     - Parámetro de Salida: `I` = Aproximación numérica de (1).

- Regla de Simpson:
  - Nombre de la función: `simpson`.
  - Parámetros iniciales: `f` = función $f(x)$; `a, b` = intervalo $[a, b]$.
  - Parámetro de Salida: `I` = Aproximación numérica de (1).
- Regla del Trapecio Compuesta:
  - Nombre de la función: `trapecio_compuesto`.
  - Parámetros iniciales: `f` = función $f(x)$; `a, b` = intervalo $[a, b]$; `N` = número de puntos en los que se divide el intervalo $[a, b]$.
  - Parámetro de Salida: `I` = Aproximación numérica de (1).
- Regla de Simpson Compuesta:
  - Nombre de la función: `simpson_compuesto`.
  - Parámetros iniciales: `f` = función $f(x)$; `a, b` = intervalo $[a, b]$; `N` = número de puntos en los que se divide el intervalo $[a, b]$.
  - Parámetro de Salida: `I` = Aproximación numérica de (1).
- Cuadratura Gaussiana:
  - Nombre de la función: `gaussiana`.
  - Parámetros iniciales: `f` = función $f(x)$; `a, b` = intervalo $[a, b]$; `M` = número de puntos a utilizar en la cuadratura.
  - Parámetro de Salida: `I` = Aproximación numérica de (1).
- Cuadratura Gaussiana Compuesta:
  - Nombre de la función: `gaussiana_compuesta`.
  - Parámetros iniciales: `f` = función $f(x)$; `a, b` = intervalo $[a, b]$; `M` = número de puntos a utilizar en la cuadratura; `N` = número de puntos en los que se divide el intervalo $[a, b]$.
  - Parámetro de Salida: `I` = Aproximación numérica de (1).

Observaciones:

- Cada función implementada en GNU Octave y Python debe tener su propia ayuda. Esta ayuda debe indicar en que consiste la función, cuales son los parámetros iniciales y cuales con los parámetros de salida.
- Cada método debe estar desarrollado como una función, y cada función debe estar en un archivo por aparte. El nombre del archivo debe ser el mismo que el nombre de la función.

2. [**Valor: 14 puntos**] En el libro *Numerical Analysis* de R. Burden y J. Faires, Novena Edición, en la sección 4.5, página 213, es explica el **método de Romberg** para aproximar el valor de la integral definida expresada en (1).

   - [**Valor: 4 puntos**] En un documento con nombre `parte1_p2.pdf`, realice una breve explicación sobre el método Romberg, mostrando su formulación matemática, además de presentar un ejemplo numérico. La formulación matemática debe indicar los valores iniciales y el valor de salida.

- [**Valor: 10 puntos**] Implemente computacionalmente en GNU Octave el método de Romberg, como una función con el nombre `romberg`. Este método debe tener su propia ayuda. Esta ayuda debe indicar en que consiste la función, cuales son los parámetros iniciales y cuales con los parámetros de salida. El nombre del archivo debe ser el mismo que el nombre de la función.

3. [**Valor: 20 puntos**] Utilizando las funciones implementadas en las preguntas 1 y 2, desarrolle un paquete computacional en GNU Octave, basándose en los siguientes indicaciones:

   - El nombre del paquete debe ser `NumInt`.
   - Para crear el paquete computacional, utilice la información que se encuentra en la siguiente dirección electrónica:

     `https://octave.org/doc/v4.2.2/Creating-Packages.html`.

     También pueden usar de referencia el documento `paquetes_octave.pdf` que se encuentra en el TEC Digital.
   - Para este paquete computacional, se debe elaborar un manual de usuario. El manual de usuario debe contener lo siguiente:
     – Portada con nombre del paquete, nombre del TEC, nombre del curso y el nombre de los miembros del grupo.
     – Tabla de Contenidos
     – Una sección donde se explique en que consiste el paquete computacional.
     – Una sección que explique como instalar el paquete computacional y que requisitos se necesitan para su uso.
     – Una sección donde explique el uso de las funciones implementadas, con su formulación matemática y ejemplos ilustrativos. Esta sección debe contener todo lo necesario para saber utilizar las funciones implementadas.

     El nombre del manual deben ser `manual_NumInt.pdf`. La estructura del manual se puede basar en el manual desarrollado para el paquete NumPy de Python, el cual se encuentra en el TEC Digital, con el nombre `userguide_numpy.pdf`. **Se tomará en cuenta la apariencia, aspecto y calidad del manual en el puntaje de esta pregunta.**

4. [**Valor: 15 puntos**] Investigue un problema aplicado a la ingeniería que necesite aproximar el valor de una integral definida como se expresa en (1). Encuentre la solución utilizando **todos** los métodos implementados en el paquete `NumInt`. La parte escrita debe estar desarrollada en un documento con nombre `parte1_p4.pdf`, el cual debe incluir lo siguiente:

   - Problema en el área de la ingeniería a resolver, explicado en forma simple y resumida. Deben incluir la referencia bibliográfica de donde se seleccionó dicho problema.
   - Problema matemático presentado como un problema similar al presentado en 1.
   - Solución del problema usando la implementación computacional de todos los método desarrollados en el paquete `NumInt`. Cada grupo define como escoger los valores iniciales. El nombre del archivo debe ser `parte1_solucion_aplicación.m`.
   - **Cada grupo debe utilizar un problema diferente**. Apenas cada grupo seleccione el problema a utilizar, deben indicarlo al profesor correo electrónico `jusoto@tec.ac.cr`, para verificar que ningún otro grupo lo seleccionó.

Parte 2: Ecuación Diferencial Ordinaria de Segundo Orden

Descripción General

Esta tarea consiste en resolver el siguiente problema, utilizando el lenguaje de programación Python.

Problema A: Sea $y = y(x)$ una función continua en un intervalo $[a, b]$. Considere la ecuación diferencial

$$\begin{cases} y'' = p(x)y' + q(x)y + f(x) \\ \\ y(a) = y_0, \quad y(b) = y_n \end{cases}$$

donde $p(x), q(x), f(x)$ son funciones continuas de variable real. Definimos $x_0 = a$, $x_n = b$, y un conjunto soporte $\mathcal{S} = \{x_0, x_1, ..., x_n\}$, tal que $h = x_{i+1} - x_i$, para todo $i = 0, 1, ..., n-1$. El problema consiste en calcular un conjunto de puntos $\{(x_0, y_0),\ (x_1, y_1),\ ....,\ (x_n, y_n)\}$ que aproximen numéricamente la función y en el intervalo $[a, b]$.

Preguntas

1. [**Valor: 25 puntos**] Implemente computacionalmente en Python el método de diferencias finitas, explicado en el documento `diferencias_finitas.pdf`, que da una solución numérica al **Problema A** . Para eso, elabore una función con nombre `edo2`, cuyos parámetros iniciales son las funciones p, q y f, el tamaño de paso h, los valores a, b del intervalo y los valores iniciales y_0, y_n. Los parámetros de salida son los vectores $x = [x_0, x_1, ..., x_n]^T$ y $y = [y_0, y_1, ..., y_n]^T$.

   Nota: La función `edo2` necesita resolver un sistema de ecuaciones lineal cuya matriz de coeficientes es una matriz tridiagonal. Para resolver dicho sistema, utilice el comando `numpy.linalg.solve`.

2. [**Valor: 10 puntos**] Implemente computacionalmente un *script* para aproximar la solución del problema

   $$\begin{cases} y'' = -\dfrac{1}{x}y' + \left(\dfrac{1}{4x^2} - 1\right) y \\ \\ y(1) = 1, \quad y(6) = 0 \end{cases} \tag{2}$$

   Para esto, debe generar una animación en la cual debe aparecer cada 2 segundo una nueva gráfica que representa una aproximación de la solución a la ecuación diferencial (2) con diferentes valores de h. Para eso, utilicen los valores $h = 10^{-i}$, donde $i = 1, 2, ..., 10$. Adicionalmente, al inicio de la animación debe aparecer la solución exacta del problema

   $$y(x) = \frac{\sin(6-x)}{\sin(5)\sqrt{x}}.$$

   La animación debe indicar una leyenda para cada gráfica. Un ejemplo de la animación solicitada se encuentra en el video `ejemplo_animacion.mp4` (El video no corresponde a este ejercicio).

   Sugerencia: En Python, utilice el comando `time.sleep` del modulo `time`.

Información de la Entrega

- **Fecha y hora límite**: Lunes 3 de Agosto a las 11:59 pm.
- Los documentos deben estar guardados usando la siguiente estructura: Una carpeta principal con nombre `Tarea 3`. Dentro de esta carpeta debe existir dos carpetas con nombres `Pregunta 1` y `Pregunta 2`. En cada una de estas carpetas estarán todos los archivos necesarios para el desarrollo de las preguntas mencionadas anteriormente.
- Deben enviar la carpeta `Tarea 1` en formato `zip` al correo `jusoto@tec.ac.cr`, con el encabezado `Entrega Tarea 3 - ANPI`. En el cuerpo del correo deben indicar el nombre completo de los miembros del grupo.
- **OBSERVACIÓN IMPORTANTE: La entrega tardía de la tarea se penalizará con una reducción del 20% de la nota final, POR CADA HORA DE ATRASO**.

Defensa

- Cada grupo debe defender esta tarea frente al profesor. Para eso deben seleccionar un horario de la siguiente dirección electrónica:

  `https://doodle.com/poll/8y5uu334aketsr4u`

- Deben escribir el nombre (sin apellidos) de todos los miembros del grupo y seleccionar uno de los horarios disponibles.
- Todos los miembros del grupo deben estar presentes para defender cada una de las preguntas. Si un estudiante no esta presente, entonces el estudiante perderá 35 puntos de la nota final.