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

CE-3102: Análisis Numéricos para Ingeniería
Semestre: II - 2021
Valor Porcentual: 15 %

Tarea 1

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/1APyO6iyjGc-dRYafCkImadVOnFOlfgNM`

- Todos los archivos de esta tarea se encuentran en la carptea de One Drive del curso.
- Los archivos computacionales implementados en GNU Octave, Pyhton y C++ 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.**

Parte 1: Librería `FunTras` (C++)

Descripción General

- **Definición**: Una función trascendente es una función que no satisface una ecuación polinomial. Ejemplo de funciones trascendentes son e^x, $\ln(x)$, $\sin(x)$ y $\dfrac{1}{x}$.
- Esta parte de la tarea consiste en desarrollar una libreria estática en **C++** 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 (`**`). **No pueden usar la división (`/`).**
- En esta parte de la tarea se evaluarán los siguientes atributos de egresado, de acuerdo con la definición de la CEAB:
  - **Conocimientos de Ingeniería (Avanzado)**
  - **Investigación (Medio)**

Preguntas

1. **[Valor: 20 puntos]** Implemente computacionalmente en C++ las funciones transcendentes que se encuentran en la siguiente tabla.

Función $f(x)$	Comando en C++	Función $f(x)$	Comando en C++
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)`
$\log_a(x)$	`log_t(x,a)`	a^x	`power_t(x,a)`
$\sin^{-1}(x)$	`asin_t(x)`	$\sqrt{x}$	`sqrt_t(x)`
$\sqrt[a]{x}$	`root_t(x,a)`	π	`pi()`

   - Para realizar dicha implementación, deben leer el documento `fun_tras.pdf` que se encuentra en la carpeta de One Drive del curso. Este documento contiene los métodos iterativos que deben implementar para aproximar las funciones que se encuentran en la tabla anterior.

- Todas las funciones deben estar implementada en un archivo con nombre `funtras.h`.
- Para su implementación, cada método iterativo debe usar una tolerancia de 10^{-8}, además de una cantidad máxima de 5000 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`.
- 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. Cada grupo debe implementar la función factorial.
- Utilizando las funciones implementadas, desarrolle un *script* con nombre `test_funtras.cpp` que realice la operación

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

Parte 2: Generalización del Método de Newton-Rapshon (Python)

Descripción General

- Esta parte de la tarea consiste en el estudio e implementación en **Python** de varios métodos iterativos que representan una variación del método de Newton-Raphson para aproximar una solución de una ecuación no lineal. Al final, se utilizarán estos métodos para resolver un problema del área de la ingeniería.
- En esta parte de la tarea se evaluarán los siguientes atributos de egresado, de acuerdo con la definición de la CEAB:
  - **Conocimientos de Ingeniería (Avanzado)**
  - **Investigación (Medio)**

Preguntas

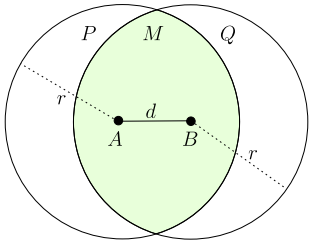
1. **[Valor: 25 puntos]** En el artículo científico *One-point Newton-type iterative methods: A unified point of view*, desarrollado por los investigadores A. Cordero, C. Jordán y J.R. Torregrosa, se desarrolla una generalización del método de Newton-Rapshon para aproximar una solución de la ecuación $f(x) = 0$. Esta generalización se representan en las ecuaciones (4) y (12) del artículo. Las fórmulas iterativas en (4) y (12) utiliza funciones de peso $H(u(x_k))$ y $G(w(x_k))$, respetivamente. Los posibles valores de dichas funciones se encuentran en las Tablas 1 y 2 del artículo. El artículo se encuentra en el TEC Digital en el archivo `one_newton.pdf`.

   (a) Implemente computacionalmente en Python las ecuaciones (4) y (12) del artículo, escogiendo solo 2 funciones de peso de cada una de las Tablas 1 y 2.
   - El nombre de las funciones en Python debe seguir el siguiente formato: `newton_H_m1` y `newton_H_m2` para los métodos basados en la ecuación (4) y la Tabla 1; `newton_G_m1` y `newton_G_m2` para los métodos basados en la ecuación (12) y la Tabla 2.
   - Los parámetros de entrada de las funciones son los siguientes: un *string* `fun` que representan a la función f, un valor inicial `x0`, una tolerancia `tol` > 0 e iteraciones máximas `iterMax`.
   - Los parámetros de salida son los siguientes: `xk` que representa la aproximación a la solución de la ecuación $f(x) = 0$, `k` que representa el número de iteraciones realizadas y `error` que representa al error $|f(x_k)|$.
   - Cada función implementada debe realizar el cálculo de las derivadas. Para eso utilicen el paquete `SymPy`.
   - Los criterios de parada para cada uno de los métodos implementados es que se cumpla $|f(x_k)| < tol$ o $k > iterMax$.

- Algunos de los métodos del artículo tienen parámetros extra (por ejemplo, β, λ, A). Cada grupo debe seleccionar un valor particular de estos parámetros para utilizarlos en las funciones implementadas.
- Todas las funciones deben estar implementadas en un archivo con nombre `metodos_p2.py`.

2. **[Valor: 5 puntos]** En la Sección 5 del artículo científico, se encuentra un conjunto de funciones para probar la eficiencia de los métodos implementados. Entre las funciones f_1, f_2, f_3 que se encuentran en esta sección, seleccione una de ellas para probar todos los métodos implementados en la Pregunta 1. Dicha prueba se debe guardar en un archivo con nombre `prueba_metodo_p2.py`. Al ejecutar cada uno de los métodos, debe aparecer un mensaje indicando: (1) la función a la cual se le desea calcular el cero, (2) el nombre del método, (3) la aproximación del cero y (4) el error. Utilice una tolerancia de 10^{-5} y una cantidad máxima de 500 iteraciones.

3. **[Valor: 10 puntos]** En el artículo científico *Estimating distances via received signal strength and connectivity in wireless sensor networks*, los autores explican una técnica para estimar la distancia en redes de sensores inalámbricos. En general, el problema se puede formular de la siguiente manera: Determinar la distancia d entre dos sensores A y B, el cuál se muestra en la siguiente figura:



En la figura anterior, r representa el radio de cobertura de cada sensor (nodo). Se asume que cada nodo tiene el mismo radio de cobertura. P, Q, M representan los vecindarios (regiones) de interés.

El valor d que calcula la distancia entre dos sensores se obtiene de encontrar la intersección con el eje d de la función

$$F(d) = \frac{\log_{10}(x_1/d)}{\sigma_R^2 \ln(10)} + \frac{d(x_2 - d)}{\sigma_c^2}, \tag{1}$$

donde:

- $\sigma_R^2 = {\sigma_{dB}}^2/(10\alpha)^2$
- $\sigma_c^2 = \frac{g^2(d)}{2\lambda k^2}\left(\frac{1}{g(d)} + \frac{1}{S}\right)$
- $k = 10\alpha/\ln(10)$
- $S = \pi r^2$
- $g(d) = \frac{2S}{\pi}\arccos\left(\frac{d}{2r}\right) - d\sqrt{r^2 - \frac{d^2}{4}}$
- $\alpha, \lambda, r, \sigma_{dB}, x_1, x_2$ son parámetros conocidos.

- Utilizando los valores $r = 10$, $\alpha = 4$, $\sigma_{dB} = 4$, $\lambda = 1$, $x_1 = 7$ y $x_2 = 6$, aproxime en Python el valor d que calcula la distancia entre los sensores usando uno de los métodos implementados en la Pregunta 1. Para esto, modifique dicho método para no tener que ingresar la función `func` como un *string*, sino que acepte una función en formado simbólico.
- Utilice una tolerancia de 10^{-5} y una cantidad máxima de 100 iteraciones.
- El valor inicial x_0 debe definirlo cada grupo.
- La implementación computacional en Python se debe realizar en un archivo con nombre `aplicacion_p2.py`.
- Al ejecutar el método, debe aparecer un mensaje indicando la aproximación del cero y el error calculado.

Parte 3: Optimización de Funciones en Varias Variables (GNU Octave)

Descripción General

- Esta parte de la tarea se basa en el artículo científico *On the global convergence of the BFGS method for nonconvex unconstrained optimization problems*. Este artículo científico presenta un método iterativo, llamado método BFGS, para encontrar el punto que minimiza una función en varias variables. La implementación computacional se debe realizar en **GNU Octave**.
- La parte escrita de esta parte debe estar en un documento con nombre `Tarea 1 - Parte 3.pdf`.

Preguntas

1. [**Valor: 5 puntos**] En la carpeta de One Drive, se encuentra el archivo `funciones_de_prueba.pdf`. Este documento contiene un conjunto de funciones de prueba en varias variables. Seleccione una función de ellas (mínimo de 5 variables). **Nota:** Escribir la función en el enlace donde se escriben los grupos (Ver la sección de Instrucciones Generales de este documento).
2. [**Valor: 20 puntos**] Implemente computacionalmente en GNU Octave el método BFGS que se encuentra en el página 3 del artículo científico para encontrar el mínimo de la función seleccionada en la Pregunta 1.
   - Utilice la ecuación (2.6) del artículo científico del método BFGS para escoger el paso de búsqueda λ_k.
   - Este método utilizará como condición de parada el criterio $\|\nabla_f(\mathbf{x}_k)\|_2 \leq 10^{-5}$, además de generar una gráfica de iteraciones versus error.
   - El vector inicial $\mathbf{x}_0$ debe ser generado de forma aleatoria.
   - El nombre del archivo debe ser `p3_bfgs.m`.
   - En el documento escrito debe indicar los resultados obtenidos, incluyendo la aproximación del punto mínimo, el número de iteraciones y el error final $\|\nabla_f(\mathbf{x}_k)\|_2$.
3. [**Valor: 15 puntos**] Investigue un problema aplicado a la ingeniería que necesite resolver un problema de minimización en varias variables, y encontrar la solución utilizando el método BFGS. La parte escrita 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 de minimización en varias variables.
   - Solución del problema usando la implementación computacional del método BFGS desarrollado en GNU Octave. Cada grupo define como escoger el valor inicial. El nombre del archivo debe ser `p3_solucion_aplicación.m`.

   Nota: Cada grupo debe utilizar un problema diferente y debe escribir la el problema en el enlace donde se escriben los grupos (Ver la sección de Instrucciones Generales de este documento).

Información de la Entrega

- **Fecha y hora límite**: Lunes 6 de Setiembre del 2021 a las 11:59 pm.
- Los documentos deben estar en una carpeta principal con nombre `Tarea 1 - Grupo #`, donde `#` es el número de cada grupo. Dentro de esta carpeta debe existir dos 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.
- Deben enviar la carpeta `Tarea 1 - Grupo #` en formato `zip` al correo `jusoto@tec.ac.cr`, con el encabezado `Entrega Tarea 1 - Grupo # - ANPI`. En el cuerpo del correo deben indicar el nombre completo de los miembros del grupo.
- **OBSERVACIÓN IMPORTANTE: Las entregas tardías se penalizarán con una reducción del 25% de la nota máxima por día de atraso. A las tareas que excedan el plazo de entrega en 3 días o más después de la fecha límite, se les asignará la nota de 0**.

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/f37g2r5ap4rwzvtm`

- 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.

Modalidad de la Calificación

- La calificación de la tarea se dividirá en 2 partes:
  - **Parte 1:** El 75% de la nota se obtendrá de la calificación se llevará a cabo a través de la defensa de cada una de las preguntas de la tarea. En esta defensa, el profesor escogerá al azar un estudiante del grupo para que conteste alguna de las preguntas de la tarea. Todos los miembros del grupo deben ser capaces de contestar cualquiera de las preguntas realizadas por el profesor. Si un estudiante no puede contestar correctamente la pregunta realizada por el profesor, entonces el grupo perderá un máximo del 20% del puntaje de dicha pregunta.
  - **Parte 2**: El 25% de la nota se obtendrá de una calificáción realizada entre los miembros del grupo. Antes de iniciar la defensa, el profesor les compartirá un enlace para que todos los miembros del grupo evaluen el trabajo de cada uno de los miembros de su grupo. La rúbrica de la evaluación de esta parte será la siguiente:

Estudiante: _____

Califiación del Trabajo en Grupo	Siempre (4 pts)	Generalmente (3 pts)	A Veces (2 pts)	Muy pocas veces (1 pt)	Nunca (0 pts)
1. Mostró Interes en el Trabajo					
2. Debatió con argumentos sus puntos de vista					
3. Colaboró en el trabajo en equipo					
4. Aportó ideas para la compresion del tema					
5. Escuchó con atención la opinión de los otros compañeros					
6. Realizó el trabajo asignado por el grupo					
7. Expuso a los demás compañeros el trabajo asignado					