Tarea 1

Fecha de publicación: Septiembre 3, 2020

Fecha de entrega: Domingo 13 de septiembre de 2020.

Indicaciones para el envío de las respuestas:

1. Crea un archivo en el que puedas agregar texto e imágenes, y que después lo puedas exportar a PDF.
2. Comienza escribiendo tu nombre y el número de la tarea.
3. Antes de poner la respuesta, escribe a qué ejercicio corresponde.
4. Para los ejercicios teóricos, puedes escribir directamente la respuesta en el archivo o puedes poner una foto de las hojas en donde escribiste tu respuesta. Sólo cuida que se vean claras las imágenes.
5. Para los ejercicios de programación:
   (a) Escribe el nombre del archivo que se tiene que pasar al interprete de Python, o en el caso de C/C++, el archivo que se tiene que compilar. Si requiere alguna instrucción especial para compilarlo, agrega el comando.
   (b) Si el programa recibe parámetros, indica en que orden se tienen que proporcionar.
   (c) Da al menos un ejemplo de la manera en que se tiene que ejecutar tu programa y agrega la salida del programa, poniendo directamente el texto que imprime tu programa o poniendo una imagen.
   (d) Si se te pide generar una gráfica de algún resultado, agrégala al archivo de respuestas.
   (e) Si tiene algun comentario, agrégalo también al documento de respuestas.
6. Trata de dejar algún espacio en blanco entre los ejercicios, por si el revisor quiere poner comentarios.

7. Si hay ejercicios de programación, sube los códigos en un archivo ZIP para que no tengas que subir archivo por archivo. Recuerda no adjuntar los executables y los códigos de cada ejercicio deben estar en carpetas diferentes.

8. Convierte el archivo de respuestas a un PDF y súbelo a la tarea que corresponda. Revisa que estén los archivos que quieres enviar antes de presiona el botón para enviar tu trabajo.

Ejercicio 1. (2 puntos)
Considere la función $f(x) = x + \ln\sqrt{x} - 2.5$. Verique que la función $g(x) = 2.5 - \ln\sqrt{x}$ cumple con las condiciones del teorema de punto fijo en el intervalo $I = [1, 3]$, de modo que puede usarse para encontrar una raíz de $f(x)$. Estime el número de iteraciones que requiere el algoritmo de punto fijo para aproximar a la raíz de $f(x)$ con un error menor a 10^{-6}, partiendo de cualquier punto en el intervalo I.

Ejercicio 2. (1 punto)
Explique la manera en que se puede usar el siguiente teorema para usarlo junto con algún método de cálculo de raíces visto en clase para tratar de encontrar todas las raíces de un polinomio, y la mencione la conveniencia de usar esa estrategia dependiendo de η.

Theorem 1. *Todos los ceros del polinomio $p_n(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$ de grado n están contenidas en el círculo Ω del plano complejo definido por*

$$\boxed{\Omega = \{z \in \mathbb{C} : \; |z| \leq 1 + \eta\},}$$

donde

$$\boxed{\eta = \max_{0 \leq k \leq n-1} \left|\frac{a_k}{a_n}\right|.}$$

Ejercicio 3. (3 puntos)
Programar y probar el método de la secante de acuerdo al algoritmo visto en la clase 5.

1. Escriba la función que reciba como argumentos
   - el apuntador a la función $f(x)$.
   - el valores de los dos puntos iniciales x_0, x_1.
   - el valor de la tolerancia τ.
   - el máximo número de iteraciones N.

2. La información que debe devolver la función es

- el último punto x_k que genera el algoritmo,
- el valor $f(x_k)$,
- el número k de iteraciones realizadas,
- un variable res que indique la condición en la que terminó el algoritmo: $res = 0$ si $|f(x_k)| < \tau$; $res = 1$ si el denominador $f(x_k) - f(x_{k-1})$ es casi cero; $res = 2$ si no se cumplió ninguna de las condiciones anteriores.

3. Escriba un programa que pruebe la función anterior en los siguiente casos, usando una tolerancia $\tau = \sqrt{\epsilon_m}$:

   (a) $f(x) = e^{2x} - x - 6$, con $x_0 = 3$ y $x_1 = 15$.

   (b) $f(x) = 3\cos(3\pi x) - 4x$, con $x_0 = -0.75$ y $x_1 = -0.8$.

   (c) $f(x) = \arctan x$, con $x_0 = 0.5$ y $x_1 = 0.6$.

   (d) $f(x) = \arctan x$, con $x_0 = 2$ y $x_1 = 2.8$.

4. En cada caso haga que el programa imprima los datos:

$$x_0, \quad x_1, \quad f(x_1), \quad k, \quad x_k, \quad f(x_k), \quad res$$

Ejercicio 4. (2 puntos)

1. Usando la implementación del método secante del Ejercicio 3 y su respuesta del Ejercicio 2, escriba un programa que trate de obtener todas las raíces reales del polinomio

$$p(x) = 6 \cdot x^5 - 25x^4 - 24x^3 + 110x^2 - 72x + 320$$

2. Haga que el programa imprima las raíces diferentes que encuentra con este procedimiento, y el valor del polinomio en esos puntos.

3. Denotando por $r_{\min}$ a la raíz más pequeña y por $r_{\max}$ la raíz más grande, eenere la gráfica del polinomio en el intervalo $[r_{\min} - 2, r_{\min} + 2]$.

Para los que programan en C/C++, haga que el programa genere un archivo de texto que tenga una tabla con dos columnas, la primera columna corresponde a una secuencia de puntos $r_{\min} - 2 = x_0 < x_1 < ... < x_m = r_{\min} + 2$; y la segunda columna tiene los valores $p(x_k)$. Use alguna herramienta que lea este archivo y genere la gráfica del polinomio.

Para los que programan en Python, pueden generar directamente la gráfica, si almacenan los valores x_k y $p(x_k)$ en arreglos que los usen en las instrucciones para generar la grafica.

En cualquier caso, agregue la gráfica al archivo de respuestas.

Ejercicio 5. (2 puntos)
Dado un polinomio $p_n(x)$ de grado n, el método de Horner permite evaluar al polinomio realizando n sumas y n multiplicaciones ($2n$ FLOPs), en lugar de los n^2 FLOPs (aproximadamente) que se realizan si se evalua el polinomio tal y como indica su expresión:

$$\begin{aligned} p_n(x) &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n \\ &= a_0 + x\bigg(a_1 + x\Big(a_2 + x\big(a_3 + \cdots + x(a_{n-1} + x\, a_n)\cdots\big)\Big)\bigg) \end{aligned}$$

Para evaluar el polinomio $p_n(x)$ en un punto x_0 con este algoritmo, podemos ir generando la siguiente secuencia, que corresponden a los valores que se van obteniendo en la expresión anterior al hacer las operaciones anidadas.

$$\begin{array}{rcl} b_n &=& a_n \\ b_{n-1} &=& a_{n-1} + b_n x_0 \\ &\vdots& \\ b_1 &=& a_1 + b_2 x_0 \\ b_0 &=& a_0 + b_1 x_0 \end{array} \tag{1}$$

Al final, b_0 es igual a $p_n(x_0)$. Ahora, si en las expresiones anteriores despejamos los coeficientes a_i y hacemos la sustitución en el polinomio $p_n(x)$ obtenemos:

$$\begin{array}{rcl} p_n(x) &=& a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n \\ &=& (b_0 - b_1 x_0) + (b_1 - b_2 x_0)x + (b_2 - b_3 x_0)x^2 + ... + (b_{n-1} - b_n x_0)x^{n-1} + b_n x^n \\ &=& b_0 + b_1(x - x_0) + b_2 x(x - x_0) + b_3 x^2 (x - x_0) + ... + b_n x^{n-1}(x - x_0) \\ &=& b_0 + q_{n-1}(x; x_0)(x - x_0) \end{array}$$

donde $q_{n-1}(x;\, x_0) = b_1 + b_2 x + b_3 x^2 + ... + b_n x^{n-1}$ es un polinomio de grado $n-1$. El valor x_0 se pone como parámetro del polinomio q_{n-1} para indicar que los coeficientes b_i dependen de x_0. Así, el método de Horner también nos da el polinomio $q_{n-1}(x;\, x_0)$ que resulta al dividir el polinomio $p_n(x)$ entre $x - x_0$:

$$p_n(x) = q_{n-1}(x;\, x_0)(x - x_0) + b_0 \tag{2}$$

Esto permite calcular el valor de la derivada de $p_n(x)$ en el punto x_0:

$$p_n'(x) = \frac{d}{dx}\left\{(x - x_0)q_{n-1}(x;\, x_0)\right\} = q_{n-1}(x;\, x_0) + (x - x_0)q_{n-1}'(x;\, x_0)$$

Entonces la derivada de p_n en el punto x_0 se obtiene al evaluar q_{n-1} en x_0:

$$p_n'(x_0) = q_{n-1}(x_0;\, x_0). \tag{3}$$

Por lo anterior, conviene que al aplicar el método de Horner para evaluar al polinomio $p_n(x)$ se obtenga tanto el valor del polinomio como los coeficientes de $q_{n-1}(x;\, x_0)$.

1. Escriba una función que implemente el método de Horner, como se describe en el Algoritmo 1.
   Note que para representar al polinomio sólo se requiere sus coeficientes, por lo que al algoritmo de Horner se le pasa un arreglo con sus coeficientes:

   $$\mathtt{a} = \{a_n,\ a_{n-1},\ ...,\ a_1,\ a_0\}.$$

   Note que por el orden en que se almacenan los coeficientes en el arreglo, la entrada $\mathtt{a}[i]$ corresponde al coeficiente a_{n-i}.

2. Use el polinomio $p(x)$ del Ejercicio 4 y escriba la función que corresponde a su derivada $p'(x)$.

3. Escriba un programa que genere 20 puntos equidistantes del intervalo, $-5 = x_0 < x_1 < ... < x_{19} = 5$, y que con el siguiente procedimiento calcule los errores promedio en la evaluación del polinomio $p(x)$, que tiene sus coeficientes en el arreglo `a`, y de su derivada $p'(x)$:

   Fijar $E_p = 0$ y $E_d = 0$
   Para $k = 0, ..., 9$:
   - Calcular $px, \mathtt{b} \leftarrow \text{horner}(n, \mathtt{a}, x_k)$
   - Calcular $dpx, \mathtt{b2} \leftarrow \text{horner}(n-1, \mathtt{b}, x_k)$
   - $E_p = E_p + |p(x_i) - px|$
   - $E_d = E_d + |p'(x_i) - dpx|$
   - Imprimir $k,\ px,\ p(x_i) - px,\ dpx,\ p'(x_i) - dpx$

   Imprimir $E_p/20$ y $E_d/20$.

Algoritmo 1: Método de Horner

Entrada: El grado n del polinomio, el arreglo a con los coeficientes del polinomio $p_n(x) = a[0]x^n + a[1]x^{n-1} + ... + a[n-1]x + a[n]$ y el punto x_0.

Resultado: El valor de la evalución $y = p_n(x_0)$ y el arreglo `b` con los coeficientes del polinomio asociado $q_{n-1}(x; x_0)$.

Reservar memoria para el arreglo `b` con n elementos;

$\texttt{b}[0] = \texttt{a}[0]$;

for $k = 1, 2, ..., n-1$ **do**

  $\texttt{b}[k] = \texttt{a}[k] - \texttt{b}[k-1]x_0$;

end

$y = \texttt{a}[n] - \texttt{b}[n-1]x_0$;