Matemática II (B)
Segundo Cuatrimestre - 2022

Práctica 2: Resolución de Ecuaciones no Lineales

Método de bisección, Newton y secante.

1. Implementar un programa que reciba como input una función f, dos números a, b, un número de iteraciones n y aplique el método de bisección para aproximar una raíz de f en el intervalo $[a, b]$, devolviendo la última iteración.

2. Modificar el programa del ejercio anterior de manera que en vez de una cantidad de iteraciones n, reciba una tolerancia *tol* de manera que el programa frene cuando $|b_n - a_n| < tol$.

3. Determinar una función f y números reales a y b para obtener un valor aproximado de $\sqrt[3]{5}$ usando el método de bisección. ¿Cuántos pasos son necesarios para asegurar que el error es menor que 10^{-10}? Hacer un programa para hallar el valor de dicha aproximación.

4. Elegir un intervalo apropiado y utilizar el método de bisección para calcular una raíz de la ecuación:

$$x + 0.5 = \tan(x)$$

   ¿Cuántos pasos hay que hacer para garantizar que el error sea menor que 10^{-5}?

5. Dada $f(x) = \frac{1}{4}e^{2x} - 1$ mostrar que f tiene una única raíz real y dicha raíz está en el intervalo $[0, 1]$. Probar que el método de Newton-Raphson converge a la raíz cualquiera sea el valor x_0 inicial.

6. Sea $f(x) = e^{x^2+x+1} - e^{x^3} - 2$. Sabiendo que $f''(x) > 0$ para $x \in [0, 1/2]$, determinar un intervalo $I \subset [0, 1/2]$ para el cual se puede asegurar que Newton-Raphson converge a una solución de la ecuación $f(x) = 0$ para todo $x_0 \in I$. ¿Cuántos pasos son necesarios para asegurar que el error cometido con dicha aproximación es menor a 10^{-10}?

7. Implementar un programa que reciba como input una función f, su derivada f', un punto inicial x_0 y una tolerancia ε y aplique el método de Newton-Raphson para buscar una raíz de f a partir de x_0. El programa debe finalizar cuando $|x_n - x_{n-1}| < \varepsilon$.

   Opcional: agregar un parámetro adicional N que indique la cantidad máxima de pasos. Si no se alcanza convergencia luego de N pasos, imprimir un mensaje de error.

8. Considerar la ecuación $\tan(x) - \frac{1}{x} = 0$ en el intervalo $[0.5, 1]$.

a) Determinar un intervalo $I \subset [0.5, 1]$ donde Newton- Raphson converge a la solución. (Ayudarse de los gráficos de f' y f'' para hallar las cotas de la convergencia).

b) Para el intervalo hallado en el ítem anterior determinar el número de pasos necesarios para asegurar que el error cometido con Newton-Raphson es menor que 10^{-5}. Comparar con la cantidad de pasos necesarios para obtener la misma cota del error con el método de bisección.

c) Usando el programa del ejercicio anterior, hallar una aproximación para la solución de la ecuación.

9. a) Implementar un programa que reciba como input una función f y dos puntos x_0 y x_1 y aplique el método de la secante para buscar una raíz de f.

   b) Para la función $f(x) = x^2 - 5$ comparar los resultados obtenidos con el método de la secante y el de Newton-Raphson. Comparar los resultados obtenidos en cada paso.

10. Calcular una solución aproximada de las siguientes ecuaciones

    a) $e^{-x} + x^2 - 3\,\mathrm{sen}(x) = 0$,

    b) $e^{|x|} = \arctan(x) + 2$,

    c) $x^{15} - 2 = 0$,

    utilizando los métodos anteriores. Comparar cómo se comportan los errores y decidir cuál tiene la convergencia más rápida.

Métodos de punto fijo, métodos en dos dimensiones

11. Sea $g(x) = (\frac{x^2+1}{2})^{1/2}$. Mostrar que para cualquier intervalo acotado I, existe $0 \leq \gamma < 1$ tal que $|g'(x)| \leq \gamma < 1$ para $x \in I$. Probar que el método de punto fijo converge a una solución de la ecuación $g(x) = x$ en forma incondicional, es decir, para cualquier punto inicial.

12. Estudiar numéricamente las iteraciones de $g(x) = \lambda x(1 - x)$ para $x_0 = 0.3$, en los casos $\lambda = 0.9, 2.5, 3.2, 3.56, 3.9$.

13. a) Encontrar un intervalo $I \subset \mathbb{R}$ tal que la función $g(x) = \frac{1}{4}\left(\cos(x) + x^2\right)$ tenga un único punto fijo en I y el método converja al punto fijo dado cualquier punto inicial del intervalo. Escribir la fórmula de recurrencia del método.

    b) Determinar cuántas iteraciones del método deben hacerse para asegurar que el error cometido es menor que 10^{-4}.

14. Dada la ecuación $e^{-x} - x = 0$, se definen las funciones $g_1(x) = -\ln(x)$, $g_2(x) = e^{-x}$ y $g_3(x) = \dfrac{e^{-x} + x}{2}$. Para cada $g_i(x)$ hallar, si es posible, $I \subset (0, 1)$ tal que $g_i(x)$ tenga un único punto fijo en I y, dado cualquier punto inicial del intervalo, el método converja a dicho punto fijo. ¿Cuál de las $g_i(x)$ elegiría y por qué?

15. El siguiente sistema de ecuaciones representa la dinámica de un sistema transcripcional en donde la proteina inhibe (reprime) su propia producción. La variable x representa la concentración de mARN y la variable y representa la concentración de proteína:

$$\begin{cases} \dot{x} = \kappa_1 f(y) - \gamma_1 x, \\ \dot{y} = \kappa_2 x - \gamma_2 y, \end{cases}$$

donde $\kappa_1, \kappa_2 > 0$ son constantes que representan la tasa de producción y $\gamma_1, \gamma_2 > 0$ son constantes que representan la tasa de degradación. La no linealidad viene dada por la función $f(y) = \dfrac{\theta^\alpha}{\theta^\alpha + y^\alpha}$, con $\theta > 0$. Considerar $\kappa_1 = 2, \kappa_2 = 1, \gamma_1 = \gamma_2 = 1, \theta = 1$ y $\alpha = 0.5$. Plantear un método de punto fijo convergente y luego aproximar el estado estacionario del sistema implementando dicho método. Recordar que los estados estacionarios son aquellos en los que $\dot{x} = \dot{y} = 0$.

16. Comenzando con $(x_0, y_0) = (1, 2)$, utilizar el método de Newton-Raphson para aproximar una solución del sistema

$$\begin{cases} 3x^2 + (y-2)^2 &= 4 \\ y &= e^x \end{cases}$$

Resolución de sistemas lineales

17. Resolver (con lápiz y papel) el siguiente sistema lineal:

$$\begin{array}{rcrcl} 7x & - & y & = & 16 \\ 3x & + & 2y & = & 19 \end{array}$$

Comparar con el resultado obtenido al resolver numéricamente utilizando los comandos `numpy.linalg.solve` y `scipy.linalg.solve`.

18. Una empresa de fertilizantes para césped produce tres tipos de productos. El primero tiene un 23 % de nitrógeno, un 5 % de fosfato y un 8 % de potasio; los porcentajes del segundo producto son 20 % de nitrógeno, un 7 % de fosfato y un 12 % de potasio; y finalmente para el tercero son 17 %, 2 % y 1 %, respectivamente. Si cada año el césped de una quinta requiere 400 g de nitrógeno, 90 g de fosfato y 140 g de potasio por cada 100 hectáreas, ¿cuánto se necesitará de cada uno de los tres productos para aplicar en un año en 100 hectáreas?

19. Otra forma de resolver sistemas lineales es por medio de métodos iterativos o indirectos. Se comienza con un valor inicial arbitrario $x^{(0)}$ y luego se va mejorando la solución hasta que la diferencia entre $x^{(k-1)}$ y $x^{(k)}$ (los valores $(k-1)$-ésimo y k-ésimo, respectivamente) es menor que una cierta tolerancia prefijada. Por ejemplo, si queremos resolver el sistema $Ax = b$, el método de Gauss-Seidel propone calcular

$$x_i^{(k)} = \frac{1}{a_{i,i}} \left[b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(k-1)} \right].$$

Una condición suficiente (pero no necesaria) para que converja el método de Gauss-Seidel es que la matriz A sea estrictamente diagonal dominante. Es decir que para cada fila de A, el módulo del elemento de la diagonal sea estrictamente mayor que la suma de los módulos de los demás elementos de esa fila ($|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$).

a) Implementar un programa que reciba como input una matriz cuadrada estrictamente diagonal dominante A, un vector b, un vector inicial $x^{(0)}$, una tolerancia *tol* y un número natural N, aplique el método de Gauss-Seidel, se detenga cuando la diferencia entre $x^{(k-1)}$ y $x^{(k)}$ sea menor que la tolerancia *tol* o bien cuando se haya alcanzado el máximo de iteraciones N, y devuelva el último vector calculado.

```
def iterador_Gauss_Seidel(##COMPLETAR##):
  x = ##COMPLETAR
  for iter in range(##COMPLETAR):
      x_original = x.copy()
      for i in range(##COMPLETAR):
          s1 = np.dot(A[i, :i], x[:i])
          s2 = np.dot(A[i, i + 1 :], x_original[i + 1 :])
          x[i] = ##COMPLETAR
      if np.allclose(x, x_original, rtol=tol):
        break
  return(x)
```

b) Aplicar el método de Gauss-Seidel para resolver el siguiente sistema lineal:

$$\begin{array}{rcrcrcr} 8x & - & y & + & 3z & = & 18 \\ x & + & 10y & - & 3z & = & 31.5 \\ 4x & + & 3y & + & 12z & = & 23 \end{array}$$

c) El método de Jacobi propone calcular

$$x_i^{(k)} = \frac{1}{a_{i,i}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{i,j} x_j^{(k-1)} \right].$$

Al igual que en el método de Gauss-Seidel, una condición suficiente (pero no necesaria) para que converja el método de Jacobi es que la matriz A sea estrictamente diagonal dominante. Implementar un programa que reciba como input una matriz cuadrada estrictamente diagonal dominante A, un vector b, un vector inicial $x^{(0)}$, una tolerancia *tol* y un número natural N, aplique el método de Jacobi, se detenga cuando cuando la diferencia entre $x^{(k-1)}$ y $x^{(k)}$ sea menor que la tolerancia *tol* o bien cuando se haya alcanzado el máximo de iteraciones N, y devuelva el último vector calculado. Aplicar el método de Jacobi para resolver el sistema lineal del ítem anterior.