

Manual de Usuario de los Métodos

Prerrequisitos

Capítulo 1

1. Búsquedas incrementales:
 - a. Entradas:
 - i. FuncionX: función continua a evaluar
 - ii. Xi: punto de valor numérico donde inicia la búsqueda
 - iii. DeltaX: incremento en cada iteración
 - iv. MaxIteracion: la máxima cantidad de iteraciones que puede realizar
 - b. Condiciones:
 - i. La función debe estar escrita en sintaxis de operaciones de Python sin librerías
 - ii. Xi debe ser un valor entero
 - iii. DeltaX en caso de no ser entero, el decimal debe ser separado con “.”
 - iv. MaxIteracion solo puede ser un valor entero positivo
 - c. Salida:
 - i. Entrega la raíz o un rango donde está la raíz o un mensaje de que no existe una raíz
2. Bisección y Regla Falsa:
 - a. Entrada:
 - i. Xi: el límite inferior del intervalo en el que se busca la raíz
 - ii. Xf: el límite superior del intervalo en el que se busca la raíz
 - iii. FuncionX: función a evaluar
 - iv. Tol: margen de error permitido del valor de la raíz
 - b. Condiciones:
 - i. Tanto Xi como Xf debe ser un valor entero
 - ii. La función debe estar escrita en sintaxis de operaciones de Python sin librerías
 - iii. Tol en caso de no ser entero, el decimal debe ser separado con “.”
 - c. Salidas:
 - i. El valor de la raíz exacta o el valor de la raíz con un margen de error o en mensaje informando que el intervalo es incorrecto
3. Newton:
 - a. Entrada:
 - i. FuncionX: función continua a evaluar
 - ii. DerivadaX: derivada de la función a evaluar
 - iii. X0: valor inicial en que realizara el método
 - iv. Tol: margen de error permitido del valor de la raíz
 - v. MaxIter: la máxima cantidad de iteraciones que puede realizar
 - b. Condiciones:
 - i. Tanto funcionX continua como derivadaX deben estar escritas en sintaxis de operaciones de Python sin librerías
 - ii. X0 debe ser un valor entero
 - iii. Tol en caso de no ser entero, el decimal debe ser separado con “.”
 - iv. MaxIteracion solo puede ser un valor entero positivo
 - c. Salida:

- i. Valor de la raíz con un margen de error o un mensaje de que la derivada es cero o debido a que supero la cantidad de iteraciones y el error es menor a la tolerancia entrega mensaje de que no encontró raíz
- 4. Punto Fijo:
 - a. Entrada:
 - i. FuncionGX: función después de despejar X que se usará para evaluar escrita en sintaxis de operaciones de Python sin librerías
 - ii. X0: valor inicial en que realizara el método
 - iii. Tol: margen de error permitido del valor de la raíz
 - iv. MaxIter: la máxima cantidad de iteraciones que puede realizar
 - b. Condiciones:
 - i. funcionGX debe estar escrita en sintaxis de operaciones de Python sin librerías
 - ii. X0 debe ser un valor entero
 - iii. Tol en caso de no ser entero, el decimal debe ser separado con "."
 - iv. MaxIteracion solo puede ser un valor entero positivo
 - c. Salida:
 - i. Valor de la raíz con una tolerancia especificada en las entradas o un mensaje de que no se encuentra una raíz debido a que el error es mayor que la tolerancia
- 5. Secante:
 - a. Entrada
 - i. FuncionX: función continua a evaluar
 - ii. X0: valor inicial a evaluar la función
 - iii. X1: valor siguiente del valor inicial para evaluar la función
 - iv. Tol: margen de error permitido del valor de la raíz
 - v. MaxIter: la máxima cantidad de iteraciones que puede realizar
 - b. Condiciones
 - i. La función debe estar escrita en sintaxis de operaciones de Python sin librerías
 - ii. X0 y X1 debe de ser un valor entero
 - iii. Tol en caso de no ser entero, el decimal debe ser separado con "."
 - iv. MaxIter solo puede ser un valor entero positivo
 - c. Salida:
 - i. Valor de la raíz con un margen de error o un mensaje de que no se encontró raíz
- 6. Raíces Múltiples:
 - a. Entrada:
 - i. FuncionX: función continua a Evaluar
 - ii. DerivateX: derivada de la función a evaluar
 - iii. Derivate2X: segunda derivada de la función a evaluar
 - iv. X0: valor inicial en que realizara el método
 - v. Tol: margen de error permitido del valor de la raíz
 - vi. MaxIter: la máxima cantidad de iteraciones que puede realizar
 - b. Condiciones:
 - i. Tanto funcionX, derivateX y derivate2X deben estar escritas en sintaxis de operaciones de Python sin librerías
 - ii. X0 y X1 debe de ser un valor entero
 - iii. Tol en caso de no ser entero, el decimal debe ser separado con "."

- iv. MaxIter solo puede ser un valor entero positivo
- c. Salida:
 - i. Valor de la raíz con un margen de error especificado con en las entradas

Capítulo 2

1. Eliminación Gaussiana Simple:

- a. Entradas:
 - i. A =Matriz Aumentada de una matriz invertible y un vector constante
 - ii. N =número de Incógnitas
- b. Condiciones:
 - i. En la diagonal no debe haber algún valor cero
 - ii. El número de incógnitas debe ser correcto, de lo contrario el método fallará
 - iii. la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
- c. Salida:
 - i. Solución del sistema de ecuaciones

2. Eliminación Gaussiana con pivoteo Parcial:

- a. Entradas:
 - i. A =Matriz Aumentada de una matriz invertible y un vector constante.
 - ii. N =número de Incógnitas
- b. Condiciones:
 - i. En cada columna debajo del pivote debe de haber al menos un elemento diferente de cero incluyendo al pivote
 - ii. El número de incógnitas debe ser correcto, de lo contrario el método fallará
 - iii. la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
- iv.
- c. Salida:
 - i. Solución del sistema de ecuaciones

3. Eliminación Gaussiana con pivoteo Total:

- a. Entradas:
 - i. A =Matriz Aumentada de una matriz invertible y un vector constante
 - ii. N =número de Incógnitas
- b. Condiciones:
 - i. Debe ser posible solucionarlo con pivoteo Total
 - ii. El número de incógnitas debe ser correcto, de lo contrario el método fallará
 - iii. la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
- c. Salida:
 - i. Solución del sistema de ecuaciones

4. Factorización LU:

- a. Entradas:
 - i. A : Matriz Invertible
 - ii. $_b$: Vector constante
- b. Condiciones:

- i. La matriz debe ser cuadrada, la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
 - ii. El vector debe ser escrito de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - c. Salida:
 - i. Solución del sistema de Ecuaciones
- 5. Factorización LU con pivoteo Parcial:
 - a. Entradas:
 - i. A: Matriz invertible
 - ii. _b: Vector Constante
 - b. Condiciones:
 - i. La matriz debe ser cuadrada, la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
 - ii. El vector debe ser escrito de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - c. Salida:
 - i. Solución del sistema de Ecuaciones
- 6. Método de Doolittle:
 - a. Entradas:
 - i. A: Matriz Invertible
 - ii. _b: Vector constante
 - iii. _n: Tamaño de la Matriz A
 - b. Condiciones:
 - i. La matriz debe ser cuadrada, la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
 - ii. El vector debe ser escrito de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - iii. _n debe ser el tamaño correcto de la matriz, de lo contrario, el método fallará y generará error
 - c. Salida:
 - i. Solución al sistema de ecuaciones
- 7. Método de Crout:
 - a. Entradas:
 - i. A: Matriz Invertible
 - ii. _b: Vector constante
 - iii. _n: Tamaño de la Matriz A
 - b. Condiciones:
 - i. La matriz debe ser cuadrada, la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2

- ii. El vector debe ser escrito de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - iii. `_n` debe ser el tamaño correcto de la matriz, de lo contrario, el método fallará y generará error
 - c. Salida:
 - i. Solución al sistema de ecuaciones
- 8. Método de Cholesky:
 - a. Entradas:
 - i. A: Matriz Invertible
 - ii. `_b`: Vector constante
 - iii. `_n`: Tamaño de la Matriz A
 - b. Condiciones:
 - i. La matriz debe ser cuadrada, la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
 - ii. El vector debe ser escrito de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - iii. `_n` debe ser el tamaño correcto de la matriz, de lo contrario, el método fallará y generará error
 - c. Salida:
 - i. Solución al sistema de ecuaciones
- 9. Método de Jaccobi:
 - a. Entradas:
 - i. A: Matriz invertible
 - ii. `_b`: Vector Constante
 - iii. X0: Vector de la aproximación inicial
 - iv. Tol: Tolerancia
 - v. Nmax: número máximo de iteraciones
 - b. Condiciones:
 - i. matriz debe ser cuadrada, la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
 - ii. El vector “b” y “x0” deben ser escritos de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - iii. Tol en caso de no ser entero, el decimal debe ser separado con “.”
 - iv. MaxIteracion solo puede ser un valor entero positivo
 - c. Salida:
 - i. Solución al sistema de forma iterativa
- 10. Método de Gauss-Seidel:
 - a. Entradas:
 - i. A: Matriz invertible
 - ii. `_b`: Vector Constante
 - iii. X0: Vector de la aproximación inicial
 - iv. Tol: Tolerancia
 - v. Nmax: número máximo de iteraciones

- b. Condiciones:
 - i. matriz debe ser cuadrada, la notación de la matriz debe ser escrita diferenciando las filas con “;” u las columnas con “,”. No debe tener espacios ni otros caracteres, de lo contrario el algoritmo no podrá leer la matriz. un ejemplo puede ser: 2,2;2,2
 - ii. El vector “b” y “x0” deben ser escritos de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - iii. Tol en caso de no ser entero, el decimal debe ser separado con “.”
 - iv. MaxIteracion solo puede ser un valor entero positivo
- c. Salida:
 - i. Solución al sistema de forma iterativa

Capitulo 3

- 1. Vandermonde:
 - a. Entrada:
 - i. X: Vector constante con los puntos x
 - ii. Y: Vector constante con los puntos y
 - b. Condiciones:
 - i. El vector “x” y “y” deben ser escritos de forma traspuesta, por ejemplo: en vez de 2;2;2;2.... se debe escribir 2,2,2,2. De lo contrario, puede que falle el método
 - c. Salida:
 - i. Solución del método con un pequeño margen de error