

Click to verify



Metodo de biseccion ejercicios resueltos metodos numericos

Método de Bisección , Método de Newton , Aplicación del Método de Newton , Método de Montecarlo , Método de Trapecio , Método del Punto Medio , Ejercicio por el método de Bisección ,Métodos Numéricos. Ejemplos del método de Bisección y de Newton Los métodos numéricos son herramientas para la solución de problemas. Son un medio para reforzar la comprensión de las matemáticas ya que permiten convertir matemáticas superiores en operaciones aritméticas simples. Los métodos numéricos se basan en dos conceptos principales: recursión y aproximaciones. Esto significa que utilizan la recursión y las aproximaciones así como la iteración para encontrar una solución.
* Reconocer la importancia de utilizar los métodos numéricos en la resolución de problemas
* Definir, identificar y aplicar recursión y aproximación como base de los métodos numéricos
* Definir dónde se utilizan los métodos numéricos
* Describir los pasos para la programación de métodos numéricos
• Los métodos numéricos se utilizan para:
• Solución de sistemas de ecuaciones lineales
• Solución de ecuaciones no lineales y trascendentales
• Encontrar un valor por medio de tablas: interpolación
• Encontrar un comportamiento (un modelo) a partir de datos ajustando Aritmética del Computador y Errores
Introducción Ejercicios Aritmética del computador. Cancelación Propagación del Error
Ejercicios Interpolación Polinomial. Aspectos Prácticos
Introducción Interpolación polinomial. Forma de Lagrange del polinomio interpolante. Forma modificada y forma baricéntrica de Lagrange. Forma baricéntrica con nodos igualmente espaciados. Ejercicios Forma de Newton para el polinomio interpolante. Diferencias Divididas de Newton. Forma de Newton en el caso de nodos igualmente espaciados. Ejercicios Forma de Lagrange vs Forma de Newton. Estimación del error. Error en interpolación lineal. Error en interpolación cuadrática
Error en interpolación cúbica
una curva: ajuste de curvas
• Integración numérica de una función
• Solución numérica de ecuaciones diferenciales
Error con interpolación con polinomios de grado n. Otros casos. Interpolación Iterada de Neville
2.16.1 Algoritmo Ejercicios Trazadores Cúbicos (Cubic Splines). Ejercicios Algoritmos e implementación con Basic de OpenOffice o de LibreOffice, y Calc.
2.18.1 Forma de Lagrange del polinomio interpolante
Ejercicios Forma modificada y forma baricéntrica de Lagrange. Ejercicios Forma de Newton del polinomio interpolante. Ejercicios Trazadores cúbicos
Ejercicios Interpolación. Aspectos Teóricos. Forma de Lagrange para el polinomio interpolante. Forma de Lagrange modificada y forma baricéntrica de Lagrange. Forma de Newton para el polinomio interpolante. Estimación del error. Polinomios de TChebyshev y convergencia. Ejercicios Ecuaciones no lineales. Orden de convergencia
Ejercicios Método de Punto Fijo
Punto Fijo. Algoritmo e Implementación. Ejercicios Punto Fijo. Aspectos Teóricos. El método de Bisección
Algoritmo e Implementación. Bisección: Criterio de Parada y Número de Iteraciones. Ejercicios Bisección: Teorema de Convergencia. Orden de Convergencia. Acerca del Criterio de Parada en métodos iterativos. El Método de Newton
Método de Newton: Algoritmo e Implementación. Ejercicios Método de Newton: Teorema de Convergencia. Orden de Convergencia. Ejercicios Método de Newton: Estimación del error
Métodos de Orden Cúbico. Método de Euler. Ejercicios Un método h íbrido: NewtonBisección. Algoritmo e Implementación en VBA Excel. Ejercicios El Método de la Falsa Posición
Algoritmo. Teorema de Convergencia. Orden de Convergencia. Ejercicios Método de la Secante
Algoritmo e Implementación en VBA Excel. Secante: Teorema de Convergencia. Orden de Convergencia. Ejercicios Secante: Orden de convergencia. Un Método H íbrido: SecanteBisección Híbrido: Algoritmo e Implementación. Ejercicios Interpolación Inversa. Interpolación Cuadrática Inversa. Algoritmo e Implementación en Excel. Integración Numérica. Introducción
Fórmulas de NewtonCotes. Ejercicios Regla del Trapecio. Ejercicios Trapecio: Algoritmo e Implementación. Regla del Simpson. Simpson: Algoritmo e Implementación. Ejercicios Método de Romberg. Extrapolación de Richardson. Método de Romberg
Algoritmo e Implementación en VBA Excel. Cuadratura Gaussiana. Integrales Impropias. Ejercicios Ecuaciones Diferenciales Ordinarias
Método de Euler
Algoritmo e implementación con Wxmaxima. Métodos de Taylor de orden superior. Algoritmo e implementación con Wxmaxima. Métodos de RungeKutta. Algoritmo e implementación con Wxmaxima. Ejercicios Algunos Detalles Teóricos. Sea f(t) una función continua en el intervalo [(a,b)] tal que f(a)·f(b) < 0. Definimos las siguientes sucesiones: Sucesión {a_n} {n∈mathbb{N}}: Sucesión {b_n} {n∈mathbb{N}}: Sucesión {(r_n)} {n∈mathbb{N}}: Sucesión {(r_n)} {n∈mathbb{N}}: Sucesión {(r_n)} {n∈mathbb{N}}: Sucesión {(r_n)} {n∈mathbb{N}}: Observad que se cumple (a_n)leq r_n leq b_n). La aproximación a alguna raíz (r) en ((a,b)) de f(t) es el valor (r_n) (también, podrían servir (a_n) y (b_n)). Las tres sucesiones convergen a la misma raíz (r) de f(t). Si se escoge (r_n) como aproximación, el error cometido es menor o igual que ((b-a)/2^(n+1)). Nota 1: definimos (b_n+1) con las mismas condiciones que (a_n+1) para tener que hacer sólo una comprobación de signo. Nota 2: no están definidos los términos (n+1)ésimos si ((f(a_n)·f(r_n) = 0)) porque esto ocurre sólo si ((f(r_n)=0)), con lo que el método habrá hallado la raíz exacta y, por tanto, no se requiere construir más intervalos. El método de bisección es una de las soluciones numéricas básicas para hallar la raíz de una ecuación polinómica. Encierra entre paréntesis el intervalo en el que se encuentra la raíz de la ecuación y lo subdivide en mitades en cada iteración hasta encontrar la raíz. Por ello, el método de bisección también se denomina método de entre paréntesis. Sin embargo, como el mecanismo de trabajo es similar al algoritmo de búsqueda binaria, el método de bisección también se conoce como método de búsqueda binaria, método de reducción a la mitad o método de dicotomía. Se basa principalmente en el teorema del valor intermedio. Encontrar raíces de ecuaciones En este ejemplo, solo consideramos ecuaciones con una variable independiente. Puede ser lineal o no lineal. Las ecuaciones lineales se utilizan para representar la gráfica de una línea recta, mientras que las ecuaciones no lineales se utilizan para representar curvas. La raíz de una ecuación significa el valor de la variable independiente que satisface la ecuación. Por ejemplo: la raíz de una ecuación f(x)= 4·x2 = 0 es 2 porque f(2) = 4·22 = 0. Consideremos f(x) como una función continua real. Según el teorema del valor intermedio, la ecuación f(x)=0 tiene al menos una raíz entre a y b si f(a)·f(b) < 0. La función f(x) tiene una raíz, "c", entre A y B. Representación gráfica del método de bisección El siguiente gráfico representa el mecanismo de funcionamiento del método de bisección. En el gráfico podemos ver que la raíz de la ecuación está marcada en rojo. Para empezar: Primero hicimos dos conjeturas iniciales, una1 y B1, para lo cual f(a1)·pensión completa1) < 0. Según el teorema del valor intermedio, la raíz debe estar en [a1, b1]. Podemos encontrar el punto medio de un1 y B1, cual es b2. Por lo tanto, el intervalo inicial ahora se reduce a [a1, b2] porque f(a1)·pensión completa2) = 0, entonces la raíz no se encuentra en este intervalo. Por tanto, no habrá solución. Paso 3) Encuentra el punto medio, c = (a+b)/2 (i) Si el valor de la función del punto medio f(c) = 0, entonces c es la raíz. Vaya al paso 5. (ii) Si f(a)·f(c) < 0, la raíz se encuentra entre a y c. Luego establezca a = a, b = c. (iii) De lo contrario, establezca a = c, b = b. Paso 4) Si el error absoluto es mayor que la tasa de tolerancia o (b-a) > e, vaya al paso 3. Paso 5) Muestre c como la raíz aproximada. Veamos un ejemplo del algoritmo del método de bisección. Tenemos que encontrar la raíz de la siguiente función continua usando la fórmula del método de bisección. f(x) = x3 - X2 + 2
Ejemplo del método de bisección Paso 1) Asumamos, a = -10, b = 10, y e = 1% o 0.01 Paso 2) Ahora comprobaremos si f(a)·f(b) >= 0 o no. f(a) = f(-10) = (-10)3 - (-10)2 + 2 = -1098. f(b) = f(10) = (10)3 - (10)2 + 2 = 902 f(a)·f(b) = f(-10)·f(10) = (-1098)·(902) < 0 Por tanto, la raíz de la función anterior se encuentra en este intervalo [-10, 10]. Paso 3) Entonces se calculará primero el punto medio c. Ahora es necesario comprobar las siguientes condiciones: (i) si f(c) = 0: f(c) = f(0) = (0)3 - (0)2 + 2 = 2 ≠ 0 (ii) si f(a)·f(c) < 0: f(c)·f(a) = 2·(-1098) < 0 La condición se cumple. Para la próxima iteración, los valores serán, una = una = -10 segundo = c = 0 Paso 4) Como (b-a) = (0-(-10)) = 10>0.05, se repetirá el proceso. Las siguientes iteraciones se muestran en la tabla. Iteración a b c licenciado en Letras f(c) 1 -10 0 10 2 2 -5 0 -5 -148 3 -2.5 0 -2.5 2.5 -19.875 4 -1.25 0 -1.25 1.25 -1.52562 5 -1.25 -0.625 -0.625 0.625 1.36523 6 -1.25 -0.9375 -0.9375 0.3125 0.297119 7 -1.09375 -0.9375 -1.09375 0.15625 -0.50473 8 -1.01562 -0.9375 -1.01562 0.078125 -0.0791054 9 -1.01562 -0.976562 -0.976562 0.0390625 0.115003 10 -1.01562 -0.996094 -0.996094 0.0195312 0.0194703 11 -1.00586 -0.996094 -1.00586 0.00976562 -0.0294344 Paso 5) En la undécima iteración, la condición del paso 1) será falsa. Por tanto, la raíz de esta ecuación es -4. Diagrama lógico del método de bisección Pseudocódigo Start Set a, b, e if f(a)*f(b) >= 0 Output("Root does not exist in this interval") Stop while (b-a)>e do c ← (a + b)/2 if f(c) = 0 break end if if f(c)*f(a) < 0 then b ← c else a ← c end while Output(c) Stop
Ejemplo del método de bisección en C/C++
Entrada: #include using namespace std; #define Error 0.01 double value(double x) { return x*x*x - x*x + 2; } void bisection_method(double a, double b) { if (value(a) * value(b) >= 0) { cout <= Error} { c = (a+b)/2; if (value(c) == 0.0) break; else if (value(c)*value(a) < 0) b = c; else a = c; } cout <= 0.01; c = (a+b)/2 if (value(c) == 0.0): break if (value(c)*value(a) < 0): b = c else: a = c print("The root is : ", "%4f%") a =-10 b = 10 bisection_method(a, b) Salida: The root is : -1.0059
Ventajas y limitaciones del método de bisección
Estos son los pros y los contras del método de bisección:
Ventajas
Contras
Método de búsqueda de raíces fácil y sencillo de implementar. La convergencia es lenta porque se basa simplemente en reducir el intervalo a la mitad. Dado que encierra la raíz, siempre es convergente. Si una de las suposiciones iniciales está cerca de la raíz, llegar a la raíz requerirá más iteraciones. La tasa de error se puede controlar aumentando o disminuyendo el número de iteraciones. [Bisección] [Ejercicio] [Analítico] [Algoritmo] [función] .. 1. Ejercicio Referencia: Burden 2.1 ejemplo 1 p38 La ecuación mostrada tiene una raíz en [1,2], ya que f(1)=5 y f(2)=14 y existe cambio de signo. Muestre los resultados parciales del algoritmo de la bisección con una tolerancia de 0.0001 f(x) = x^3 + 4x^2 -10 = 0 [Bisección] [Ejercicio] [Analítico] [Algoritmo] [función] .. 2. Desarrollo Analítico El desarrollo del ejercicio tradicionalmente realizado con lápiz, papel y calculadora, muestra el orden y detalle de las operaciones que se pueden traducir a un algoritmo en Python. El objetivo además de desarrollar la comprensión del método, permite en una evaluación observar si el estudiante conoce el método y usa apropiadamente los valores en cada iteración. iteración 1 a = 1, b=2 c = vfrac{a+b}{2} = vfrac{1+2}{2} = 1.5 f(1) = (1)^3 + 4(1)^2 -10 = -5 f(1.5) = (1.5)^3 + 4(1.5)^2 -10 = 2.37 f(2) = (2)^3 + 4(2)^2 -10 =14 cambio de signo a la izquierda a = 1, b= c = 1.5 tramo = |1.5-1| =0.5 iteración 2 a = 1, b=1.5 c = vfrac{1+1.5}{2} = 1.25 f(1) = -5 f(1.25) = (1.25)^3 + 4(1.25)^2 -10 = -1.794 f(1.5) = 2.37 cambio de signo a la derecha a = c = 1.25, b=1.5 tramo = |1.5-1.25| = 0.25 iteración 3 continuar como tarea. La tabla resume los valores de las iteraciones tabla para Bisección i a c b f(a) f(c) f(b) tramo 1 1 1.5 2 -5 2.37 14 0.5 2 1 1.25 1.5 -5 -1.79 2.37 0.25 3 1.25 ... 1.5 La misma tabla se puede realizar con un algoritmo para tener los resultados más rápido y observar el comportamiento del método. Observe los resultados de f(c), principalmente en la iteración i=9 con tramo=0.00097 que representa el error de estimación del valor vs tolerancia. i [a' c' b'] f(a)' f(c)' f(b)'] tramo 0 [1, 1.5, 2] [-5, 2.375 14, 10.5 1 [1, 1.25, 1.5] [-5, -1.7969 2.375] 0.25 2 [1.25, 1.375, 1.5] [-1.7969 0.1621 2.375] 0.125 3 [1.25, 1.3125, 1.375] [-1.7969 -0.8484 0.1621] 0.0625 4 [1.3125, 1.34375, 1.375] [-0.8484 -0.351 0.1621] 0.03125 5 [1.34375, 1.359375, 1.375] [-0.351 -0.0964 0.1621] 0.015625 6 [1.359375, 1.3671875, 1.375] [-0.0964 0.0324 0.1621] 0.0078125 7 [1.359375, 1.36328125, 1.3671875] [-0.0964 -0.0321 0.0324] 0.00390625 8 [1.36328125, 1.365234375, 1.3671875] [-3.2150e-02 7.2025e-05 3.2356e-02] 0.001953125 9 [1.36328125, 1.3642578125, 1.365234375] [-3.2150e-02 -1.6047e-02 7.2025e-05] 0.0009765625 raíz en: 1.3642578125 >>> Se realiza la gráfica los puntos [c,f(c)] de la tabla para observar el resultado, resaltando que los puntos al final se aglomeran alrededor de la solución o raíz de la ecuación. [Bisección] [Ejercicio] [Analítico] [Algoritmo] [función] .. 3. Algoritmo en Python El video presenta el desarrollo básico conceptual del algoritmo en Python para una comprensión del proceso paso a paso. Instrucciones en Python del Algoritmo básico del video # Algoritmo de Bisección # [a,b] se escogen de la gráfica de la función # error = tolera import numpy as np import matplotlib.pyplot as plt # INGRESO fx = lambda x: x**3 + 4*x**2 - 10 a = 1 b = 2 tolera = 0.001 # PROCEDIMIENTO tramo = b-a while not(tramo) <= a = c b = b tramo = b-a # SALIDA print(' raíz en: ', c) print('error en tramo: ', tramo) [Bisección] [Ejercicio] [Analítico] [Algoritmo] [función] .. 4. Algoritmo en Python como función El algoritmo presentado en el video se puede mejorar, por ejemplo simplificando los dos condicionales en uno. if (cambia=tolera and itera import scipy.optimize as opt >>> fx = lambda x: x**3 + 4*x**2 - 10 >>> opt.bisect(fx, 1,2,xtol=0.001) 1.3642578125 que es el resultado de la raíz para la última iteración del ejercicio. Lo que muestra que el algoritmo realizado tiene un valor más aproximado. Sin embargo por didáctica y mejor comprensión de los métodos y su implementación en algoritmos que es parte del objetivo de aprendizaje, se continuará desarrollando la forma básica y detallada con Python. Referencia: [Bisección] [Ejercicio] [Analítico] [Algoritmo] [función] El documento contiene aproximadamente 20 ejercicios resueltos utilizando el método de la bisección, una técnica numérica fundamental para la resolución de ecuaciones no lineales. Cada ejercicio ha sido desarrollado paso a paso, mostrando el procedimiento detallado para encontrar la raíz de una función dentro de un intervalo dado, aplicando los criterios de convergencia y precisión requeridos. Además, se incluyen ejemplos con distintos niveles de complejidad, lo que permite reforzar la comprensión del método y su aplicación en diversos contextos matemáticos e ingenieriles. Se analiza la importancia de la selección adecuada del intervalo inicial, el número de iteraciones necesarias y el margen de error aceptable para garantizar soluciones óptimas.

- purdue global masters degree
- veru
- retail store resume template
- https://ppulhperspektywa.pl/files/edytor/file/40163747494.pdf
- https://yaslibakicisi.net/userfiles/file/561e90e6-c881-4b81-8113-76fd0503e0d5.pdf
- https://krishnabhumi.in/userfiles/file/d6951a4e-a0d2-467d-bfa8-941d3eeb087d.pdf
- which example illustrates suicingbscription pr
- hugukuma
- parkour unblocked games
- https://consilius.org/users/UserFiles/File/81836931546.pdf
- interfaces and see examples of visual coding scripts in action.