

Manual de Usuario

Por: Jorge Juan Araujo

Prueba: Es el archivo que contiene un ejemplo para cada uno de los métodos, y es donde se ejecutan estos

Capítulo 1:

Busquedas Incrementales (C1_busquedas):

Este programa encuentra un intervalo donde $f(x)$ tiene cambio de signo usando el método de búsquedas incrementales:

Entradas:

f, función continua

x0, punto inicial

h, paso

Nmax, número máximo de iteraciones

Salidas:

a, extremo izquierdo del intervalo

b, extremo derecho del intervalo

iter, número de iteraciones

errabs, error absoluto

errrel, error relativo

tabla, Tabla con cada iteración

Bisección (C2_biseccion):

Este programa halla la solución a la ecuación $f(x)=0$ en el intervalo $[a,b]$ usando el método de la bisección.

Entradas:

f, función continua

a, extremo izquierdo del intervalo inicial

b, extremo final del intervalo final

tol, tolerancia

Nmax, número máximo de iteraciones

Salidas:

x, solución
iter, número de iteraciones
errabs, error absoluto
errrel, error relativo
tabla, Tabla con cada iteración

Regla Falsa (C3_reglafalsa):

Este programa halla la solución a la ecuación $f(x)=0$ en el intervalo $[a,b]$ usando el método de la regla falsa

Entradas:

f, función continua
a, extremo derecho del intervalo inicial
b, extremo final del intervalo final
tol, tolerancia
Nmax, número máximo de iteraciones

Salidas

x, solución
iter, número de iteraciones
errabs, error absoluto
errrel, error relativo
tabla, Tabla con cada iteración

Punto Fijo (C4_puntofijo):

Este programa halla la solución a la ecuación $f(x)=0$ resolviendo el problema análogo $x=g(x)$ usando el método de punto fijo.

Entradas:

f, función continua
g, función continua
x0, aproximación inicial
tol, tolerancia
Nmax, número máximo de iteraciones

Salidas:

x, solución
iter, número de iteraciones

errabs, error absoluto
errrel, error relativo
tabla, Tabla con cada iteración

Newton (C5_newton):

Este programa halla la solución a la ecuación $f(x)=0$ usando el método de Newton

Entradas:

f, función continua
 f' , función continua
 x_0 , aproximación inicial
tol, tolerancia
Nmax, número máximo de iteraciones

Salidas:

x, solución
iter, número de iteraciones
errabs, error absoluto
errrel, error relativo
tabla, Tabla con cada iteración

Secante (C6_secante):

Este programa halla la solución a la ecuación $f(x)=0$ usando el método de la secante

Entradas:

f, función continua
 x_0 , aproximación inicial
 x_1 , aproximación inicial
tol, tolerancia
Nmax, número máximo de iteraciones

Salidas:

x, solución
iter, número de iteraciones
errabs, error absoluto
errrel, error relativo
tabla, Tabla con cada iteración

Raíces Múltiples (C7_raicesmlt):

Este programa halla la solución a la ecuación $f(x)=0$ usando el método de raíces múltiples

Entradas:

f, función continua

f', función continua primera derivada

f'', función continua segunda derivada

x0, aproximación inicial

tol, tolerancia

Nmax, número máximo de iteraciones

Salidas:

x, solución

iter, número de iteraciones

errabs, error absoluto

errrel, error relativo

tabla, Tabla con cada iteración

Capítulo 2:

Eliminación Gaussiana Simple (C8_gausspl):

Este programa halla la solución al sistema $Ax=b$ usando el método de eliminación gaussiana simple.

Entradas:

A, matrix invertible, linealmente independiente, cuadrada

b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b

Salidas:

x, solución

Eliminación Gaussiana Parcial (C9_gaussParcial):

Este programa halla la solución al sistema $Ax=b$ usando el método de eliminación gaussiana simple parcial.

Entradas:

A, matrix invertible, linealmente independiente, cuadrada

b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b

Salidas:

x, solución

Eliminación Gaussiana Total (C10_gaussTotal): XXX

Este programa halla la solución al sistema $Ax=b$ usando el método de eliminación gaussiana simple total.

Entradas:

A, matrix invertible, linealmente independiente, cuadrada

b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b

Salidas:

x, solución

Factorización LU (C11_lusimpl):

Este programa halla la solución al sistema $Ax=b$ y la factorización LU de A usando el método de factorización LU con eliminación gaussiana simple.

Entradas:

A, matrix invertible, linealmente independiente, cuadrada

b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b

Salidas:

x, solución

L, matriz L de la factorización

U, matriz U de la factorización

Crout (C13_Crout):

Este programa halla la solución al sistema $Ax=b$ y la factorización LU de A usando el método de Crout

Entradas:

A, matrix invertible, linealmente independiente, cuadrada

b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b

Salidas:

x, solución

L, matriz L de la factorización
U, matriz U de la factorización

Doolittle (C14_Doolittle):

Este programa halla la solución al sistema $Ax=b$ y la factorización LU de A usando el método de Doolittle

Entradas:

A, matrix invertible, linealmente independiente, cuadrada
b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b

Salidas:

x, solución
L, matriz L de la factorización
U, matriz U de la factorización

Cholesky (C15_Cholesky):

Este programa halla la solución al sistema $Ax=b$ y la factorización LU de A usando el método de Cholesky

Entradas:

A, matrix invertible, linealmente independiente, cuadrada
b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b

Salidas:

x, solución
L, matriz L de la factorización
U, matriz U de la factorización

Jacobi (C16_jacobi):

Este programa halla la solución al sistema $Ax=b$ usando el método de Jacobi

Entradas:

A, matrix invertible, linealmente independiente, cuadrada
b, vector constante. El numero de filas de A tiene que ser igual al numero de elementos de b
x0, aproximación inicial
tol, tolerancia
Nmax, número máximo de iteraciones

Salidas:

x, solución

iter, número de iteraciones

errabs, error absoluto

errrel, error relativo

tabla, Tabla con cada iteración (solo funciona si se usa con 4 ecuaciones, incógnitas)

Gauss-Seidel (C17_gseidel):

Este programa halla la solución al sistema $Ax=b$ usando el método de Gauss-Seidel

Entradas:

A, matrix invertible, linealmente independiente, cuadrada

b, vector constante. El número de filas de A tiene que ser igual al número de elementos de b

x0, aproximación inicial

tol, tolerancia

Nmax, número máximo de iteraciones

Salidas:

x, solución

iter, número de iteraciones

errabs, error absoluto

errrel, error relativo

tabla, Tabla con cada iteración (solo funciona si se usa con 4 ecuaciones, incógnitas)

Capítulo 3:

Vandermonde (C19_vandermonde):

Este programa halla el polinomio interpolante de los datos dados usando el método de Vandermonde

Entradas:

X, abscisas

Y, ordenadas

Salidas:

función, La función buscada

Grafica de la función y los puntos dados al principio

Diferencias Divididas (Newton) (C20_difdivididas):

Este programa halla el polinomio interpolante de los datos dados usando el método de diferencias divididas de Newton

Entradas:

X, abscisas

Y, ordenadas

Salidas:

función, La función buscada

Grafica de la función y los puntos dados al principio

Splines Lineales (C22_trazlin):

Este programa halla el polinomio interpolante de los datos dados usando el método de splines lineales

Entradas:

X, abscisas

Y, ordenadas

Salidas:

función, La función buscada

Grafica de la función y los puntos dados al principio

Splines Cuadráticos (C23_trazcuad): XXX

Este programa halla el polinomio interpolante de los datos dados usando el método de splines cuadráticos

Entradas:

X, abscisas

Y, ordenadas

Salidas:

función, La función buscada

Grafica de la función y los puntos dados al principio

Lagrange (C21_lagrange):

Este programa halla el polinomio interpolante de los datos dados usando el método de Lagrange

Entradas:

X, abscisas

Y, ordenadas

Salidas:

función, La función buscada

Grafica de la función y los puntos dados al principio