function raiz=bissec(def,a,b,tol)

f=inline(def);

c=1;

if f(a)\*f(b)<0

x=a;

while abs(f(x))>tol

x=(a+b)/2;

if f(a)\*f(x)<0

b=x;

else a=x;

end

c=c+1;

end

raiz=x;

c

else raiz='não tem troca de sinal. Portanto não tem raiz no intervalo'

end

end

function y=newton(x0,tol)

c=1;

function s=f(x)

s=log(x+1)-x^2+3;

end

function s=df(x)

s=1/(x+1)-2\*x;

end

while abs(f(x0))>tol

a=x0-f(x0)/df(x0);

x0=a;

c=c+1;

end

c

a

end

function y=pfixo(x0,tol)

c=1;

function g=f(x);

g=0.5\*log(5/x);

end

while abs(f(x0)-x0)>tol

a=f(x0);

x0=a;

c=c+1;

end

c

a

end

function y=secante(x0,x1,tol)

c=1;

function s=f(x)

s=x^2-log(x+1);

end

while abs(f(x0))>tol

a=x0-f(x0)\*(x1-x0)/(f(x1)-f(x0));

x0=x1;

x1=a;

c=c+1;

end

c

a

end

function r = diagdominante(A,n)

for i = 1:n

j = 1:n;

j(i) = [];

B = abs(A(i,j));

Check(i) = abs(A(i,i)) - sum(B); % checagem em cada linha

if Check(i) < 0

fprintf('A matriz não é diagonal dominante na linha %2i\n\n',i)

else

fprintf('A matriz é diagonal dominante na linha %2i\n\n',i)

end

end

end

function [X,delta,Z] = jacobi(A,b,X0,eps,max)

n=length(b);

Xant = X0; % inicializa Xant

X=X0; % inicializa X

Z = X0'; % inicializa Z

for k=1:max, % iterar até max vezes

for j = 1:n, % para cada equação

% X(j)= (b(j)-a(j,1)\*Xant(1)-...-a(j,j-1)\*Xant(j-1)-

%a(j,j+1)\*Xant(j+1)-...-a(n,n)\*Xant(n))/ ajj

Sum = b(j) - A(j,[1:j-1,j+1:n])\*Xant([1:j-1,j+1:n]);

X(j) = Sum/A(j,j);

end

Z = [Z;X']; % armazena a história

delta = norm(abs(X-Xant),1);

if (delta<eps) break, end

Xant = X;

end

Z

end

function [X,delta,Z] = gseidel(A,b,X0,eps,max)

n = length(b);

Xant = X0; % inicializa Xant

X=X0; % inicializa X

Z = X0'; % inicializa Z

for k=1:max, % iterar até max vezes

for j = 1:n, % para cada equação

% X(j)= (b(j)-a(j,1)\*X(1)-...-a(j,j-1)\*X(j-1)- a(j,j+1)\*Xant(j+1)-...-

%a(n,n)\*Xant(n))/ ajj

if j==1

Sum = b(1) - A(1,2:n)\*Xant(2:n);

elseif j==n

Sum = b(n) - A(n,1:n-1)\*X(1:n-1);

else

Sum = b(j)-A(j,1:j-1)\*X(1:j-1)-A(j,j+1:n)\*Xant(j+1:n);

end

X(j) = Sum/A(j,j);

end

Z = [Z;X']; % armazena a história

delta = norm(abs(X-Xant),1);

if (delta<eps) break, end

Xant = X;

end

Z

end

function s=vandermonde(x,y)

A=vander(x);

yt=y';

s=A\yt

end

function C=lagrange(x,y)

n1=length(x);

n=n1-1;

L=zeros(n1,n1);

for k=1:n+1;

v=1;

for j=1:n+1;

if k~=j

v=conv(v,poly(x(j)))/(x(k)-x(j));

end

end

L(k,:)=v;

end

C=y\*L

function a=minqreta(x,y)

n1=length(x);

m1=0;m2=0;b1=0;b2=0;

for i=1:n1;

m1=m1+x(i)^2;

m2=m2+x(i);

b1=b1+x(i)\*y(i);

b2=b2+y(i);

end

M=[n1,m2;m2,m1];

b=[b2;b1];

a=M\b

end

function a=minqparabola(x,y)

n1=length(x);

m1=0;m2=0;m3=0;m4=0;b1=0;b2=0;b3=0;

for i=1:n1;

m2=m2+x(i)^2;

m1=m1+x(i);

m3=m3+x(i)^3;

m4=m4+x(i)^4;

b1=b1+y(i);

b2=b2+x(i)\*y(i);

b3=b3+y(i)\*x(i)^2;

end

M=[n1,m1,m2;m1,m2,m3;m2,m3,m4];

b=[b1;b2;b3];

a=M\b

end

function l=riemannesq(def,a,b,n)

f=inline(def);

x=linspace(a,b,n);

y=f(x);

h=(b-a)/(n-1);

l=sum(y(1:(n-1)))\*h;

end

function r=riemanndir(def,a,b,n)

f=inline(def)

x=linspace(a,b,n);

y=f(x);

h=(b-a)/(n-1);

r=sum(y(2:(n)))\*h;

end

function I=trapezio(def,a,b,n)

f=inline(def);

I=0

h=(b-a)/(n-1)

for k=2:n-1

x=a+h\*(k-1);

I=I+f(x);

end

I=(2\*I+f(a)+f(b))\*h/2

end

function I=simpson(def,a,b,n)

f=inline(def);

I=f(a)+f(b);

h=(b-a)/(n-1);

w=4;

for k=2:n-1

x=a+h\*(k-1);

I=I+f(x)\*w;

w=6-w ;

end

I=I\*h/3

end

function [x,y]=taylor2(def,def2,a,b,y0,N)

f=inline(def);

f2=inline(def2);

h=(b-a)/N;

x=zeros(1,N+1);

y=zeros(1,N+1);

x(1)=a;

y(1)=y0;

for i=1:N;

y(i+1)=y(i)+h\*feval(f,x(i),y(i))+(h^2/2)\*feval(f2,x(i),y(i));

x(i+1)=x(i)+h;

end

y

plot(x,y);

end

function [x,y]=euler(def,a,b,y0,N)

f=inline(def);

h=(b-a)/N;

x=zeros(1,N+1);

y=zeros(1,N+1);

x(1)=a;

y(1)=y0;

for i=1:N;

y(i+1)=y(i)+h\*feval(f,x(i),y(i));

x(i+1)=x(i)+h;

end

y

plot(x,y);

end

function [x,y]=rk2(def,a,b,y0,N)

f=inline(def);

h=(b-a)/N;

x=zeros(1,N+1);

y=zeros(1,N+1);

x(1)=a;

y(1)=y0;

for i=1:N;

k1(i)=h\*feval(f,x(i),y(i));

k2(i)=h\*feval(f,x(i)+h,y(i)+k1(i));

y(i+1)=y(i)+(1/2)\*(k1(i)+k2(i));

x(i+1)=x(i)+h;

end

y

plot(x,y);

end

function [x,y]=rk4(def,a,b,y0,N)

f=inline(def);

h=(b-a)/N;

x=zeros(1,N+1);

y=zeros(1,N+1);

x(1)=a;

y(1)=y0;

for i=1:N;

k1(i)=h\*feval(f,x(i),y(i));

k2(i)=h\*feval(f,x(i)+0.5\*h,y(i)+0.5\*k1(i));

k3(i)=h\*feval(f,x(i)+0.5\*h,y(i)+0.5\*k2(i));

k4(i)=h\*feval(f,x(i)+h,y(i)+k3(i));

y(i+1)=y(i)+(1/6)\*(k1(i)+2\*k2(i)+2\*k3(i)+k4(i));

x(i+1)=x(i)+h;

end

y

plot(x,y);

end

function [x,y]=adams(def,a,b,y0,N)

f=inline(def);

h=(b-a)/N;

x=zeros(1,N+1);

y=zeros(1,N+1);

x(1)=a;

y(1)=y0;

for i=1:3;

y(i+1)=y(i)+h\*feval(f,x(i),y(i));

x(i+1)=x(i)+h;

end

for i=4:N;

y(i+1)=y(i)+(h/24)\*(55\*feval(f,x(i),y(i))-59\*feval(f,x(i-1),y(i-1))+37\*feval(f,x(i-2),y(i-2))-9\*feval(f,x(i-3),y(i-3)));

x(i+1)=x(i)+h;

end

y

plot(x,y);

end

function [x,y]=euler\_sis(def,def2,a,b,y0,w0,N)

f=inline(def);

g=inline(def2);

h=(b-a)/N;

x=zeros(1,N+1);

y=zeros(1,N+1);

x(1)=a;

y(1)=y0;

w(1)=w0;

for i=1:N;

y(i+1)=y(i)+h\*feval(f,x(i),y(i),w(i));

w(i+1)=w(i)+h\*feval(g,x(i),y(i),w(i));

x(i+1)=x(i)+h;

end

y

w

plot(x,y,x,w);

end