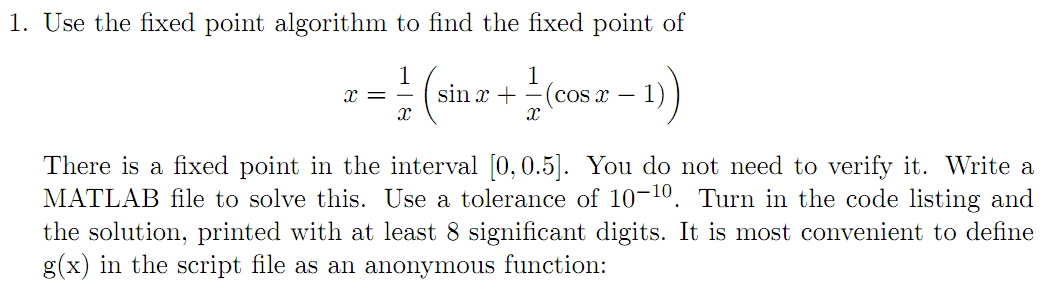
Ryan Brosnahan

Homework Set 2



%fixedPointRoot.m

function y = fipexPointRoot(g, p0,tol, maxI)

% g = function; p0 = guess; tol = precision; maxI = mapimum number of iterations

for k=1:maxI

p = g(p0);

abserr = abs(p-p0);

relerr = abserr/(abs(p)+eps);

if (abserr<tol) && (relerr<tol)

break

end

p0 = p;

end

if (k == maxI)

disp('failed to converge')

end

y = p;

>> g = @(x) (1/x)\*(sin(x)+(cos(x)-1)/x)

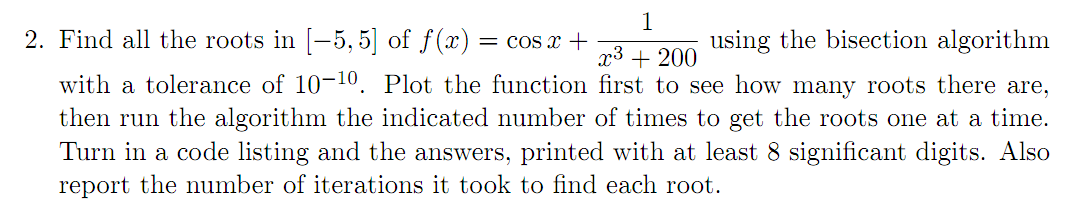
g =

@(x)(1/x)\*(sin(x)+(cos(x)-1)/x)

>> [y] = fixedPointRoot(g, 10, 10e-10, 3000)

y =

4.724436675979908e-01



%bisectionRoot.m

function [a, b, c] = bisectionRoot(f, a, b, tol)

%f = function; a = left guess; b = right guess; tol = precision

ya = f(a);

yb = f(b);

if (ya \* yb > 0)

disp('bad interval')

return;

end

maxI = 1 + round( log2((b-a)/tol ) );

for k=1:maxI

c = (a+b)/2;

yc = f(c);

if (yc == 0)

a = c; b = c;

elseif (yc \* yb > 0)

b = c; yb = yc;

else

a = c; ya = yc;

end

if (b-a < tol)

break

end

end

c = (a+b)/2;

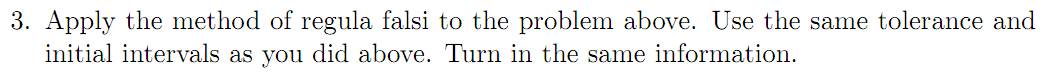
f =

@(x)cos(x)+1/(x^3+200)

>> [c] = bisectionRoot(f, 1, 2, 10e-10)

c =

1.575700417160988e+00



%RegulaFalsiRoot.m

function [c] = RegulaFalsiRoot(f, a, b, tol)

%f = function; a = left guess; b = right guess; tol = precision

ya = f(a);

yb = f(b);

if (ya \* yb > 0)

disp('bad interval')

return;

end

maxI = 50;

for k=1:maxI

c = a - ((b-a)\*(ya)/(yb-ya));

yc = f(c);

if (yc == 0)

a = c; b = c;

elseif (yc \* yb > 0)

b = c; yb = yc;

else

a = c; ya = yc;

end

if ((b-a < tol) || (abs(yc) < tol))

break

end

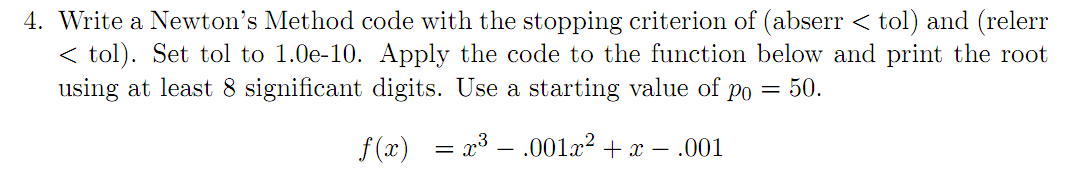
end

c = a - ((b-a)\*(ya)/(yb-ya));

>> RegulaFalsiRoot(f, 1, 2, 10e-10)

ans =

1.575700417931660e+00



%NewtonsRoot.m

function [x, k] = NewtonsRoot(f, fp, x0, tol, maxI)

%f = function; fp = fdx; x0 = guess; tol = precision; maxI = maximum

%iterations

for k=1:maxI

x = x0 - f(x0)/fp(x0);

abserr = abs(x-x0);

relerr = abserr / (abs(x) + eps);

if (abserr<tol) && (relerr<tol)

break

end

x0 = x;

end

f =

@(x)x^3-0.001\*x^2+x-0.001

>> fp = @(x) 3\*x^2 - 0.001\*2\* x + 1

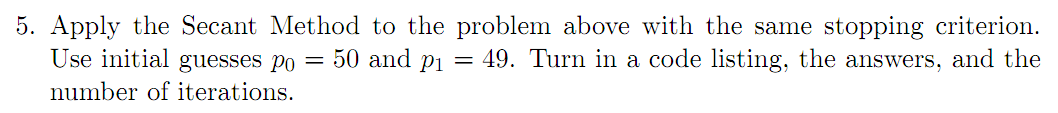
fp =

@(x)3\*x^2-0.001\*2\*x+1

>> NewtonsRoot(f, fp, 50, 1e-10, 100)

ans =

1.000000000000000e-03



%SecantRoott.m

function [x, k] = SecantRoot(f, x0, x1, tol, maxI)

%f = function; f = fdx; x0 = guess; tol = precision; maxI = maximum

%iterations

for k=1:maxI

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

abserr = abs(x-x0);

relerr = abserr / (abs(x) + eps);

if (abserr<tol) && (relerr<tol)

break

end

x1 = x0;

x0 = x;

end

>> SecantRoot(f, 50, 49, 1e-10, 100)

ans =

1.000000000000000e-03