
```

%
% BISECTION ALGORITHM 2.1
%
% To find a solution to  $f(x) = 0$  given the continuous function
%  $f$  on the interval  $[A,B]$ , where  $f(A)$  and  $f(B)$  have
% opposite signs:
%
% This code solves Example 1 on Page 50 of the book.
%
% INPUT: endpoints a,b; tolerance TOL;
%         maximum number of iterations NI.
%
% OUTPUT: approximate solution p or
%         a message that the algorithm fails.
%
% -----clean up -----
clear all; close all; clc
% -----

% Change function f for a new problem
f = @(x) (x+4)*x*x-10; % use nested formulation

% -----initialize the problem-----
% endpoints
a = 1;
b = 2;
% tolerance
TOL = 1e-4;
% maximum number of iterations
NI = 50;
% -----

% -----output on screen -----
fprintf('\n Bisection Method')
fprintf('\n\n I           P           F(P)\n')

% STEP 1
i = 1;
fa = f(a);
fb = f(b);
% sainty check
% if fa and fb have the same sign, cannot apply bisection method. EXIT
if fa*fb >=0
    fprintf('\n f(a) and f(b) have the same sign!')
    fprintf('\n Bisection method is not guaranteed to converge.')
    fprintf('\n Please choose a, b such that f(a) and f(b) have
different sign.\n')
    return
end
converge = false; % convergence flag

```

```

% STEP 2: iteration
while i<=NI
    % STEP 3
    % compute p at the i's step
    c = (b-a)/2;
    p = a+c;
    fp = f(p);

    % print out all intermediate approximations
    fprintf('%3i    %.8f    %.3e\n', i, p, fp)

    % STEP 4
    % check if meets the stopping criteria
    if (abs(fp)<eps || c < TOL) % eps is Matlab-machine zero
        converge = true;
        break;
    else
        % STEP 5
        i = i+1;
        % STEP 6
        if fa*fp > 0
            a = p;
            fa = fp;
        else
            b = p;
            fb = fp;
        end
    end
end

if converge
    fprintf('\n\nApproximate solution P = %.8f\n',p)
    fprintf('With F(P) = %.3e\n',fp)
    fprintf('Number of iterations = %3i\n',i)
    fprintf('Tolerance = %.3e (b-a)/2 = %.3e\n',TOL, c)
else
    fprintf('\n\nInteration number = %3i\n',NI)
    fprintf(' gave approximation %.8f\n',p)
    fprintf('F(P) = %.3e not within tolerance %.3e\n',fp, TOL)
end

```

Bisection Method

<i>I</i>	<i>P</i>	<i>F(P)</i>
1	1.50000000	2.375e+00
2	1.25000000	-1.797e+00
3	1.37500000	1.621e-01
4	1.31250000	-8.484e-01
5	1.34375000	-3.510e-01
6	1.35937500	-9.641e-02
7	1.36718750	3.236e-02
8	1.36328125	-3.215e-02
9	1.36523438	7.202e-05

10	1.36425781	-1.605e-02
11	1.36474609	-7.989e-03
12	1.36499023	-3.959e-03
13	1.36511230	-1.944e-03
14	1.36517334	-9.358e-04

Approximate solution $P = 1.36517334$
With $F(P) = -9.358e-04$
Number of iterations = 14
Tolerance = $1.000e-04$ $(b-a)/2 = 6.104e-05$

Published with MATLAB® R2019a