Example of a Matlab-exercise

Table of Contents

Lauri the Student 12345X

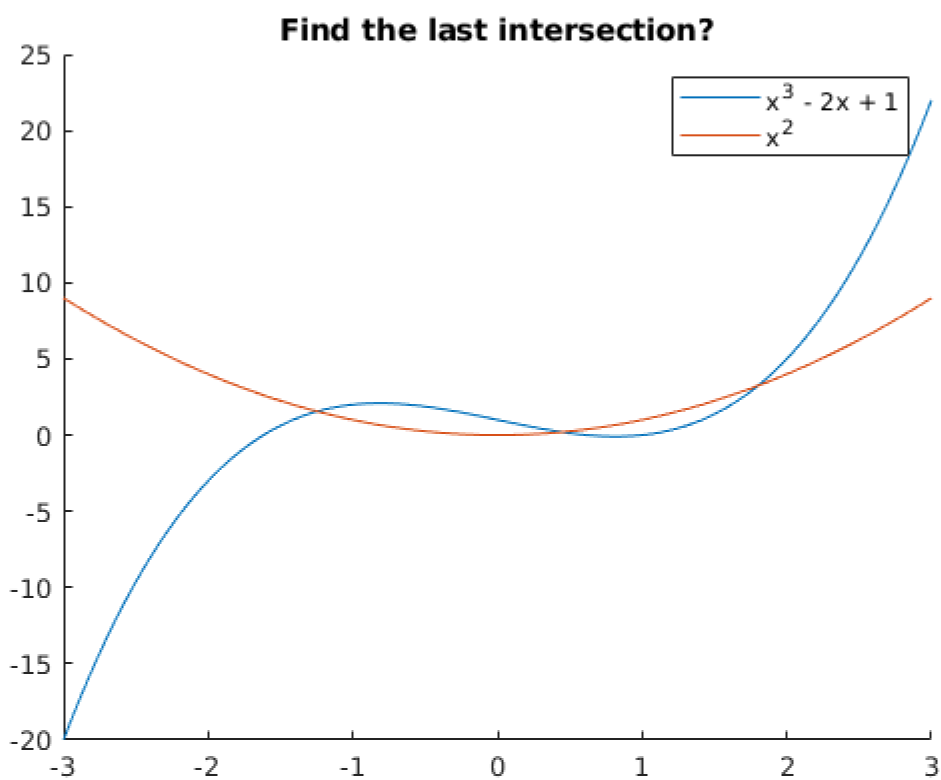
Problem: Using the bisection method, determine the point of intersection of the curves given by y = x^3 - 2x + 1 and y = x^2

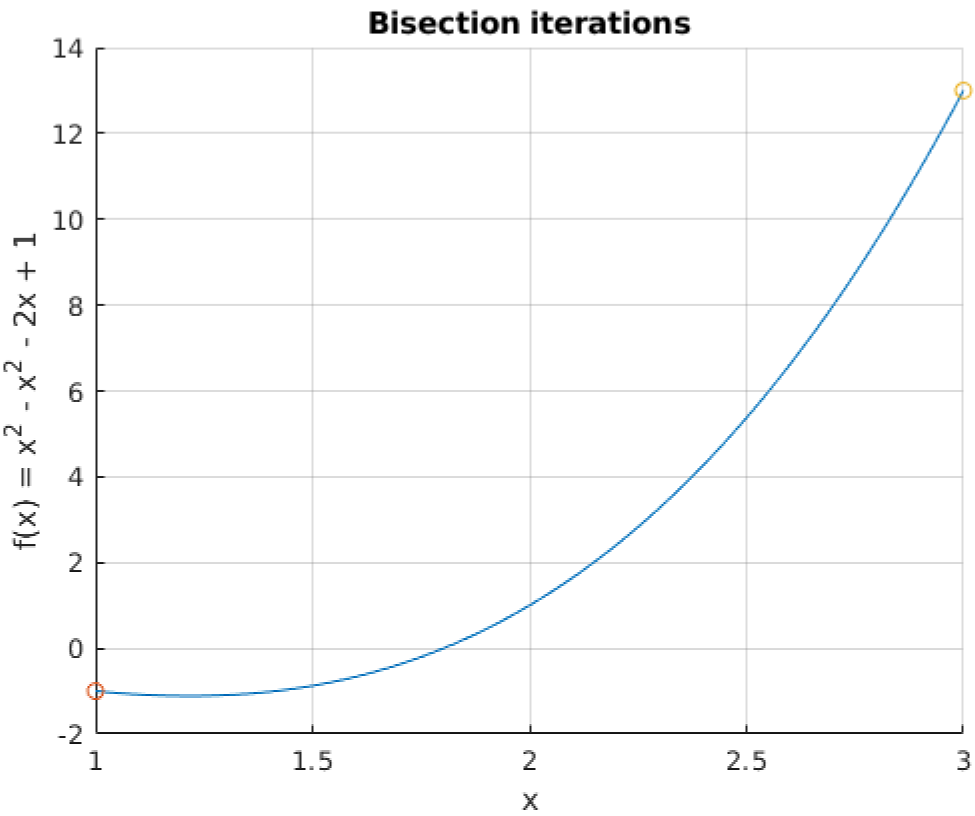
Initialize

Define given functions:

```
fun1 = @(x) x.^3 - 2*x + 1;
fun2 = @(x) x.^2;

% Plot the functions
xvec = linspace(-3, 3, 100);
figure; hold on;
plot(xvec, fun1(xvec));
plot(xvec, fun2(xvec));
legend('x^3 - 2x + 1', 'x^2');
title('Find the last intersection?');

% Actually, we are just solving the equation
% f(x) = x^3 - x^2 - 2x + 1 = 0 (which has several solutions)
% Find only the largest root
fun = @(x) x.^3 - x.^2 - 2*x + 1;

% From Mathematica/Wolfram Alpha:
xexact = 1.8019377358048382524722046390148901023318383 24;

% Guess the initial interval: [1, 3]
x0 = 1;
x1 = 3;

% Plot f in the interval
figure; hold on
xvec = linspace(x0, x1, 100);
plot1 = plot(xvec, fun(xvec));
ylabel('f(x) = x^2 - x^2 - 2x + 1');
xlabel x
title('Bisection iterations');
grid on
```

```
% Plot the interval
plot(x0, fun(x0), 'o');
plot(x1, fun(x1), 'o');

% Function value at endpoints
f0 = fun(x0);
f1 = fun(x1);

% Initial guess already correct ?
if f0 == 0
    fprintf('Solution found already! x = %g \n', x0);
elseif f1 == 0
    fprintf('Solution found already! x = %g \n', x1);
end

% The signs should differ:
if f0*f1 > 0
    error('Bad initial guess')
end
```



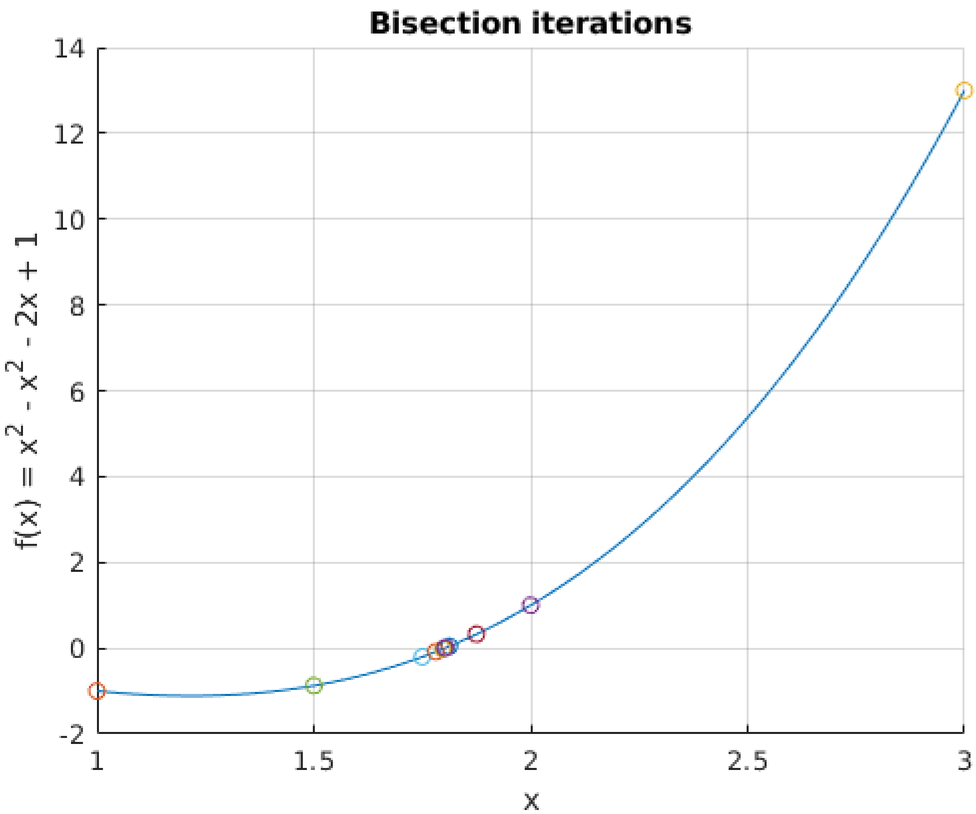


Bisection loop:

```
TOL = 1e-6; % Stop iteration when residual < TOL
MAXITER = 100; % Maximum n.o. iterations
clear errvec
disp('Start bisection loop')
for i = 1:MAXITER
    fprintf('Iter %g,interval [%g, %g], ', i, x0, x1);
    % Bisect and choose the correct interval
    xmid = 0.5 * (x0 + x1);
    fmid = fun(xmid);
    fprintf('residual = %g \n', fmid);

    plot(xmid, fmid, 'o');

    if abs(fmid) < TOL
        fprintf('Solution found within TOL=%g. x = %g \n', TOL, xmid);
        i = i-1;
        break;
    end

    if f0*fmid < 0 % Left interval
        % New interval endpoints:
        x0 = x0;
        x1 = xmid;
```

```
    elseif f1*fmid < 0 % Right interval
        x0 = xmid;
        x1 = x1;
    else
        error('Impossible?')
    end
    xeval = 0.5 * (x0 + x1);
    errvec(i) = abs(xeval - xexact);
end
fprintf('Total iterations: %g \n', i);
```

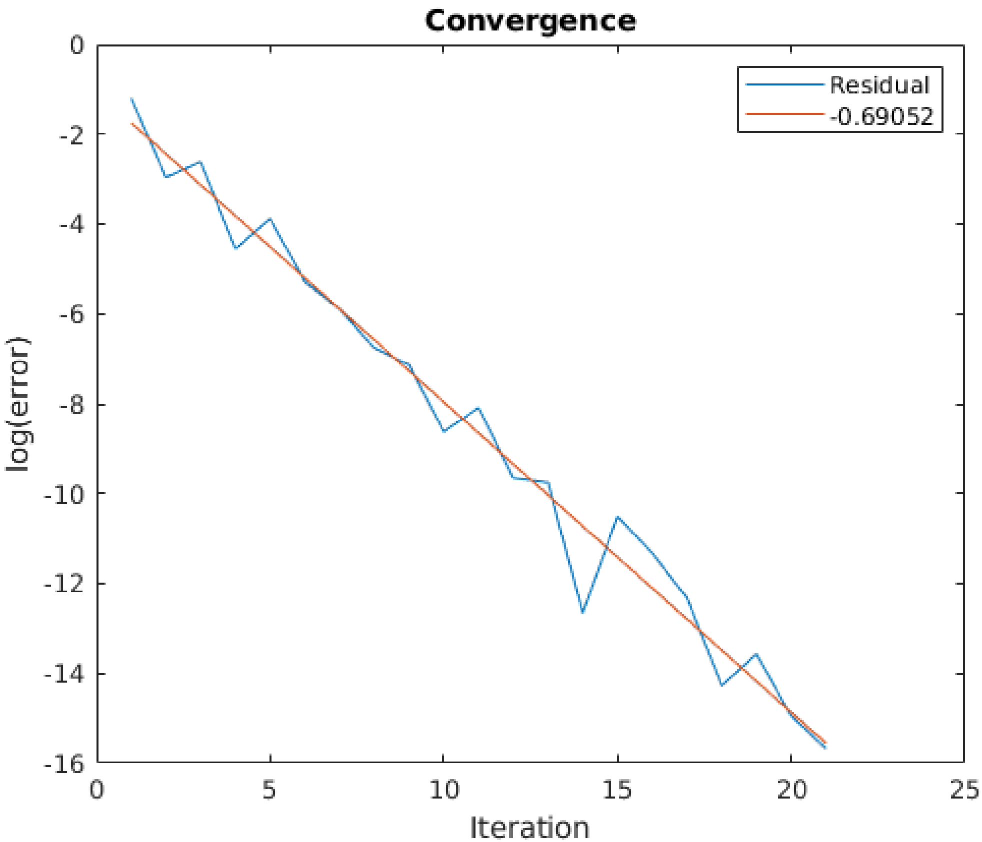
```
Start bisection loop
Iter 1,interval [1, 3], residual = 1
Iter 2,interval [1, 2], residual = -0.875
Iter 3,interval [1.5, 2], residual = -0.203125
Iter 4,interval [1.75, 2], residual = 0.326172
Iter 5,interval [1.75, 1.875], residual = 0.0441895
Iter 6,interval [1.75, 1.8125], residual = -0.0837097
Iter 7,interval [1.78125, 1.8125], residual = -0.0208321
Iter 8,interval [1.79688, 1.8125], residual = 0.0114093
Iter 9,interval [1.79688, 1.80469], residual = -0.00477856
Iter 10,interval [1.80078, 1.80469], residual = 0.00329854
Iter 11,interval [1.80078, 1.80273], residual = -0.000744211
Iter 12,interval [1.80176, 1.80273], residual = 0.00127612
Iter 13,interval [1.80176, 1.80225], residual = 0.000265689
Iter 14,interval [1.80176, 1.802], residual = -0.000239327
Iter 15,interval [1.80188, 1.802], residual = 1.31649e-05
Iter 16,interval [1.80188, 1.80194], residual = -0.000113085
Iter 17,interval [1.80191, 1.80194], residual = -4.99611e-05
Iter 18,interval [1.80193, 1.80194], residual = -1.83984e-05
Iter 19,interval [1.80193, 1.80194], residual = -2.61683e-06
Iter 20,interval [1.80194, 1.80194], residual = 5.274e-06
Iter 21,interval [1.80194, 1.80194], residual = 1.32858e-06
Iter 22,interval [1.80194, 1.80194], residual = -6.44125e-07
Solution found within TOL=1e-06. x = 1.80194
Total iterations: 21
```



Convergence analysis

```
figure;
plot(log(errvec))
title('Convergence');
xlabel Iteration
ylabel('log(error)')
fprintf('Fit 1st order polynomial (k, log(err_k)): \n');
errpoly = polyfit(1:i, log(errvec), 1)
convrate = errpoly(1);
hold on;
plot([1, i], [ polyval(errpoly, 1), polyval(errpoly, i) ] );
legend('Residual', [num2str(convrate)]);
```

```
Fit 1st order polynomial (k, log(err_k)):

errpoly =

   -0.6905   -1.0559
```



Discussion

The bisection method appears to work, and we conclude that the convergence is roughly linear. Blah Blah Blah.

Published with MATLAB® R2018a