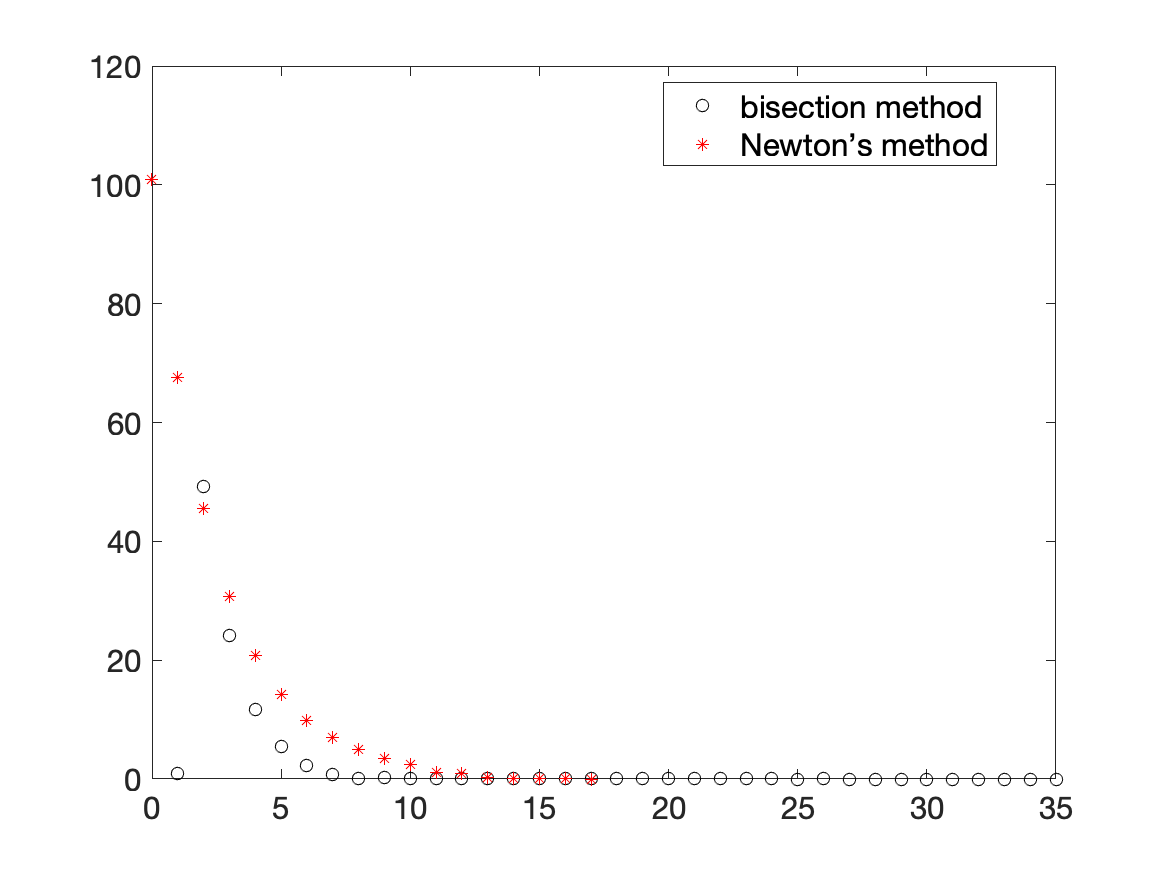
Siyue Zhu

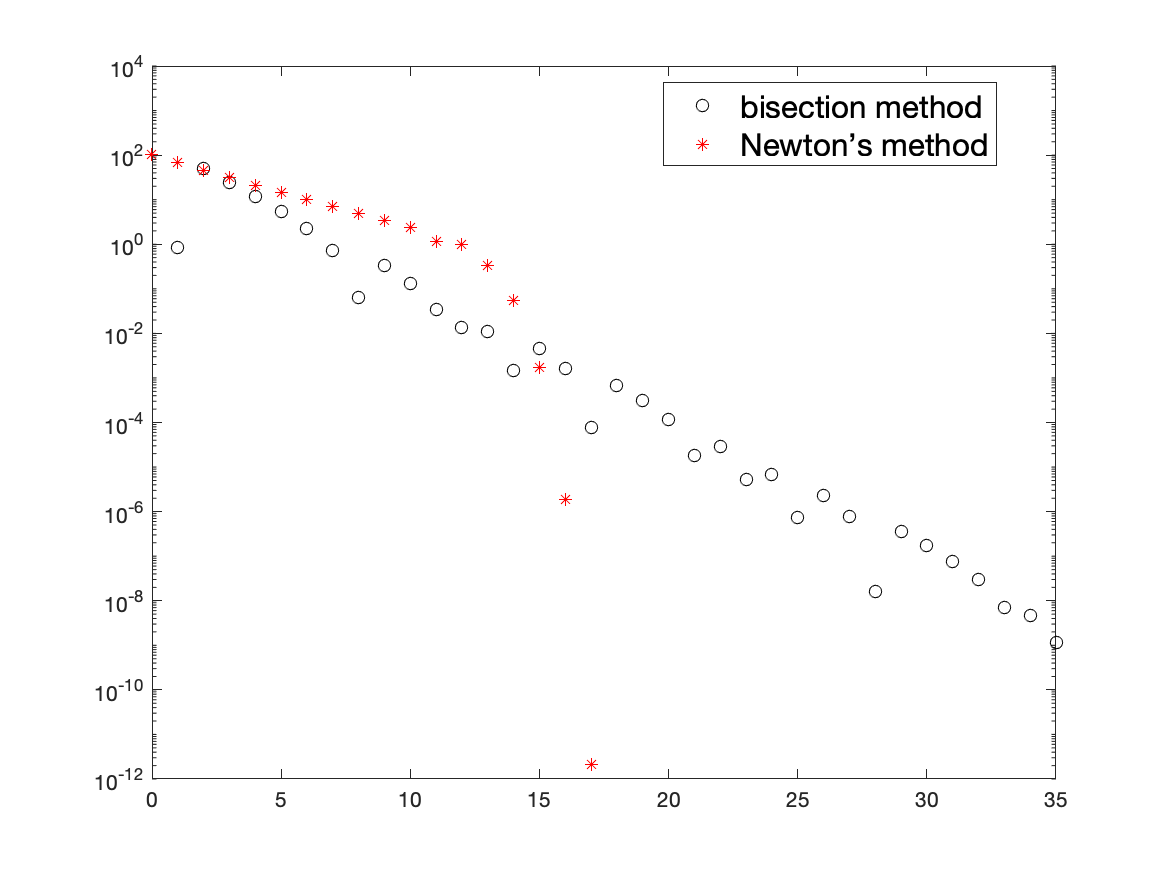
AMATH 301 Spring 2020

HW2

Problem 4

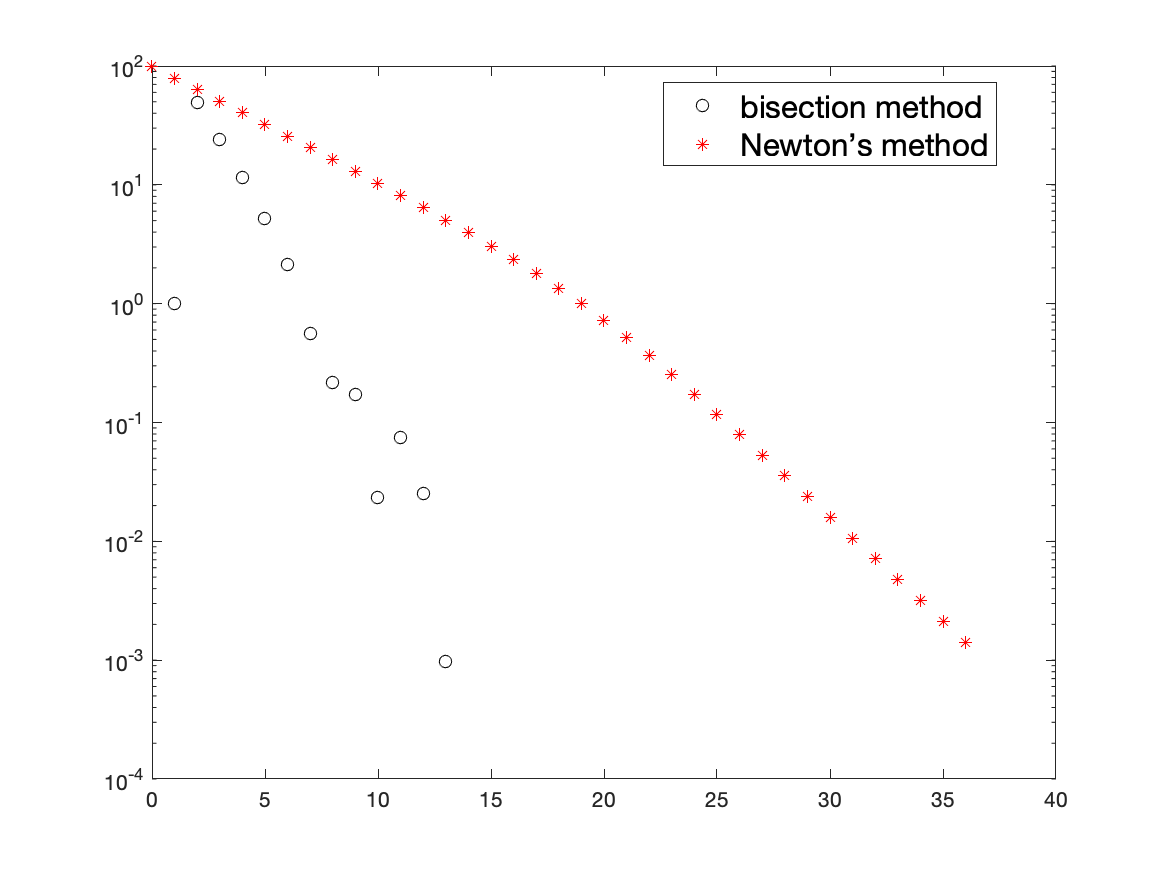


Problem 5



I think Newton’s method is better in this case since it needs less iteration in this case.

Problem 6



I think bisection method is better for this function because it needs less iteration in this case.

Problem 7

(b)

Code

% Problem 4

f = @ (x) x^3-x^2+2\*x+3;

f1 = @ (x) 3\*x^2-2\*x+2;

exact = fzero(f,0);

left = -100;

right = 100;

tolerance = 10 ^ (-8);

f\_mid = 1;

iterations = 0;

errors1 = [];

n1 = [];

while abs(f\_mid) > tolerance

mid = (left+right)/2;

f\_mid = f(mid);

if f\_mid\*f(left) < 0

right = mid;

elseif f\_mid\*f(right) < 0

left = mid;

else

disp('No root found')

break

end

errors1(iterations + 1) = abs(exact - mid);

n1(iterations + 1) = iterations + 1;

iterations = iterations + 1;

end

iterations = 0;

tolerance = 10 ^ (-8);

x = 100;

errors2 = [abs(exact - x)];

n2 = [0];

while iterations <100 && abs(f(x)) > tolerance

x = x - f(x)/f1(x);

errors2(iterations + 2) = abs(exact - x);

n2(iterations + 2) = iterations + 1;

iterations = iterations + 1;

end

plot(n1, errors1, 'ko'), hold on

plot(n2, errors2, 'r\*')

set(gca, 'fontsize', [15])

legend('bisection method', 'Newton’s method', 'location', 'best', 'fontsize', [15])

print('HW2\_fig1.png','-dpng')

% Problem 5

semilogy(n1, errors1, 'ko'), hold on

semilogy(n2, errors2, 'r\*')

legend('bisection method', 'Newton’s method', 'location', 'best', 'fontsize', [15])

print('HW2\_fig2.png','-dpng')

% Problem 6

f = @ (x) x^5-3\*x^4+5\*x^3-7\*x^2+6\*x-2;

f1 = @ (x) 5\*x^4-12\*x^3+15\*x^2-14\*x+6;

exact = fzero(f,0);

left = -100;

right = 100;

tolerance = 10 ^ (-8);

f\_mid = 1;

iterations = 0;

errors1 = [];

n1 = [];

while abs(f\_mid) > tolerance

mid = (left+right)/2;

f\_mid = f(mid);

if f\_mid\*f(left) < 0

right = mid;

elseif f\_mid\*f(right) < 0

left = mid;

else

disp('No root found')

break

end

errors1(iterations + 1) = abs(exact - mid);

n1(iterations + 1) = iterations + 1;

iterations = iterations + 1;

end

iterations = 0;

tolerance = 10 ^ (-8);

x = 100;

errors2 = [abs(exact - x)];

n2 = [0];

while iterations <100 && abs(f(x)) > tolerance

x = x - f(x)/f1(x);

errors2(iterations + 2) = abs(exact - x);

n2(iterations + 2) = iterations + 1;

iterations = iterations + 1;

end

semilogy(n1, errors1, 'ko'), hold on

semilogy(n2, errors2, 'r\*')

legend('bisection method', 'Newton’s method', 'location', 'best', 'fontsize', [15])

print('HW2\_fig3.png','-dpng')