```
function hw1_4
set(0,'defaulttextfontname','times','defaulttextfontsize',14);
set(0,'defaultaxesfontname','times','defaultaxesfontsize',14);

x = -20:0.1:20;
f = f_func(x);

% plot the function
figure(1)
plot(x,f,'ko-');
grid on; xlabel('x'); ylabel('f(x)');

epsilon = 1e-10;
x0 = -12:0.05:12;
itermax = 1000;

roots = zeros(length(x0),1);
flags = zeros(length(x0),1);
for i = 1:length(x0)
    [roots(i), flags(i)] = find_root(x0(i),epsilon,itermax);
end

roots_uniq = unique(roots);

% plot the root
figure(1)
hold on;
plot(roots,f_func(roots),'r*');
title('Fig1: function f(x)');
hold off;

% plot the initial guess vs. root
figure(2)
plot(x0,roots,'*-'); xlabel('x0'); ylabel('root / f(x)');
hold on;
plot(x0,f_func(x0),'k-'); % plot function
plot(x0,repmat(roots_uniq',length(x0),1),'r-'); % mark the roots
legend('roots','f(x)');
axis equal; axis tight;
hold off;
title('Fig2: initial guess vs. root found');

% plot the convergence
figure(3)
plot(x0,flags,'*'); xlabel('x0');
title('Fig3: the iteration converges?');
end

% find the root with Newton's method
function [root,flag] = find_root(x0,epsilon,itermax)
root = x0;
flag = false;   % whether you find a root
```

```
for i = 1:itermax
    f_curr = f_func(x0);
    f_grad_curr = f_grad_func(x0);

    if abs(f_grad_curr) < epsilon   % denominator is too small, stop
       break;
    end

    x1 = x0 - f_curr/f_grad_curr;

    if abs( x1-x0 ) < epsilon
        flag = true;     % find the root
        root = x0;
        break;
    end

    x0 = x1;
end
end

% value of the function at x
function val = f_func(x)
val = x.^2/10 + 6 * sin(x) + 2;
end

% value of the gradient at x
function val = f_grad_func(x)
val = x./5 + 6 * cos(x);
end
```

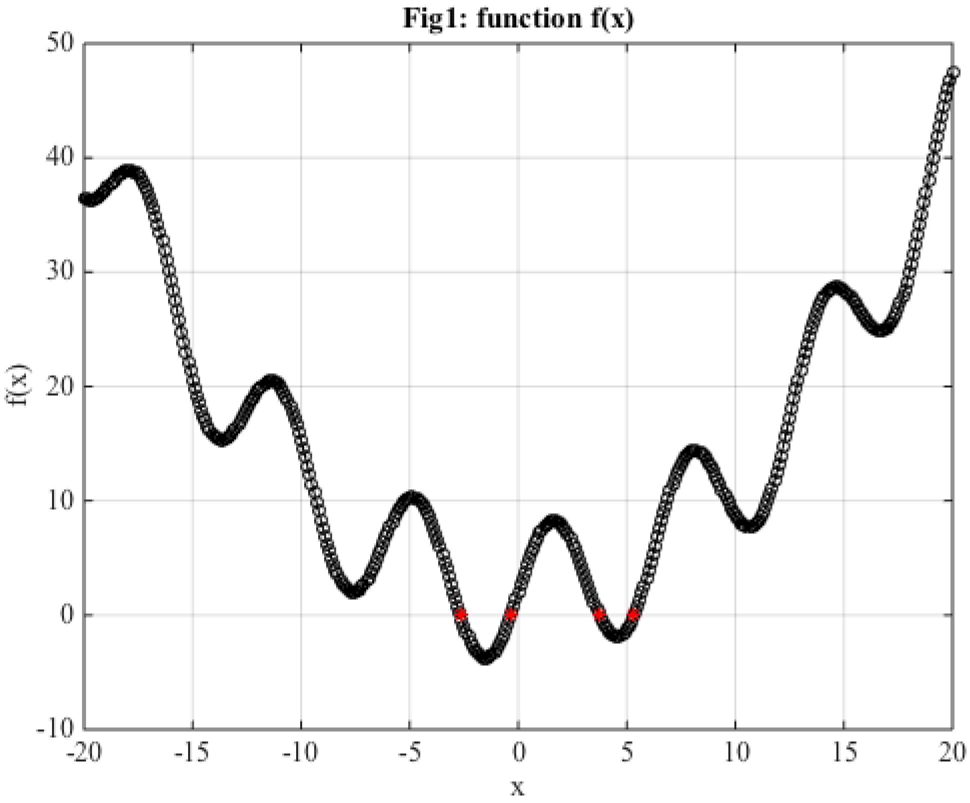


Fig1: function f(x)
f(x)
x
50
40
30
20
10
0
-10
-20
-15
-10
-5
0
5
10
15
20

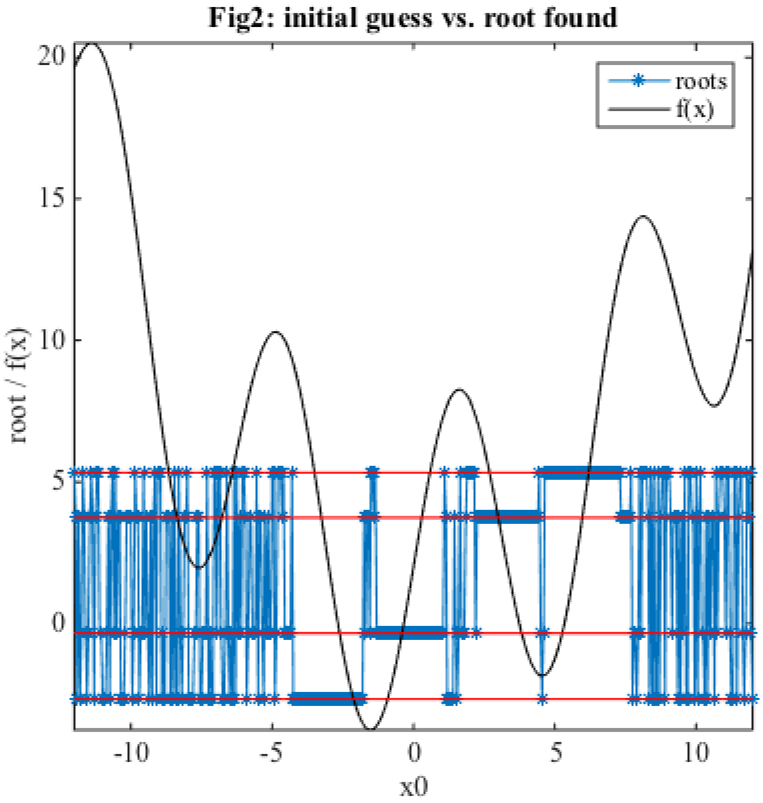
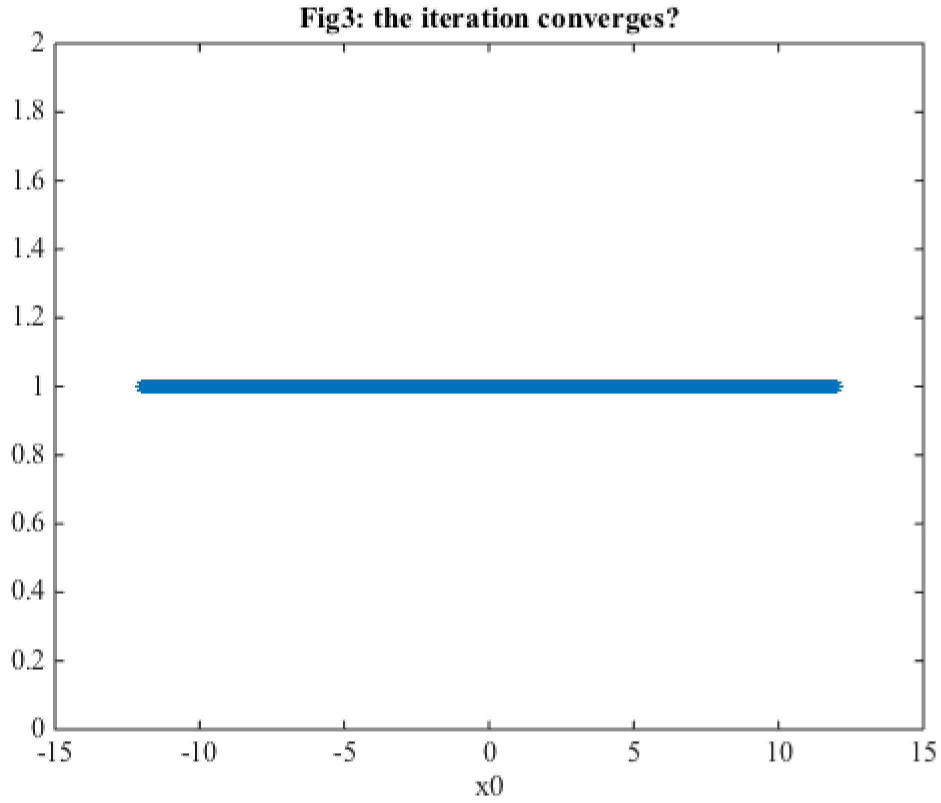


Fig2: initial guess vs. root found
roots
f(x)
root / f(x)
x0
20
15
10
5
0
-10
-5
0
5
10



Published with MATLAB® R2014b