```
% MATH 121, Spring 2023
% Author: Ojas Deshmukh, Santiago Segura, CSU Fresno State
% Demo code for L9, fixed point iteration

set(groot,'defaultAxesFontSize',18,'defaultAxesLineWidth',1.2,...
    'defaultLineLineWidth',1.0,'defaultPatchLineWidth',1.0,...
    'defaultTextFontSize',18);

MAX_ITER = 15;
ETOL = 1e-20;
XTOL = 1e-20;

% Demo
x0 = 2.5;
VERBOSE = true;


g1 = @(x) x + 2*(x.^2 - 6);
g2 = @(x) x + 0.1*(x.^2 - 6);
g3 = @(x) x + 100*(x.^2 - 6);

true_ans = sqrt(6);

[p1, err1] = my_fixedpoint(x0,g1,MAX_ITER,ETOL,XTOL,true);
[p2, err2] = my_fixedpoint(x0,g2,MAX_ITER,ETOL,XTOL,true);
[p3, err3] = my_fixedpoint(x0,g3,MAX_ITER,ETOL,XTOL,true);

figure;
semilogy(abs(true_ans-p1),'bo-'); hold on;
semilogy(abs(true_ans-p2),'rx-');
semilogy(abs(true_ans-p3),'kd-');
semilogy(abs(true_ans-p1)./(1-(p1-p2)./(p2-p3)),'b--x');
semilogy(abs(true_ans-p2)./(1-(p2-p3)./(p3-p1)),'r--x');
semilogy(abs(true_ans-p3)./(1-(p3-p1)./(p1-p2)),'k--x');
ylim([1e-20,1]);
legend({'g1','g2','g3','Aitken g1','Aitken g2','Aitken g3'},'Interpreter','latex');
xlabel('Iteration Count');
ylabel('$|\sqrt{6}-x_n|$','Interpreter','latex');

function [p, err_hist] = my_fixedpoint(x0,g,MAX_ITER,ETOL,XTOL,VERBOSE)
% This function implements the fixed-point iteration method to solve for
% the root of the function g(x).
% Inputs:
% - x0: initial guess
% - g: function handle of the iteration function
% - MAX_ITER: maximum number of iterations allowed
% - ETOL: error tolerance for stopping criterion
% - XTOL: tolerance for stopping criterion based on change in p
% - VERBOSE: if true, print iteration information
% Outputs:
% - p: the approximation for the root
% - err_hist: a vector containing the absolute error at each iteration

q = x0;
p0 = g(q);
err = 1;
count = 1;

p_hist = [q p0];
```

```
err_hist = [1 1];

% Implementation of fixed point iteration
p = p0;
while count < MAX_ITER && abs(err) > ETOL && abs(p-p0) > XTOL
    p = g(p0);
    lam = (p-p0)/(p0-q);
    err = lam/(1-lam)*(p-p0);
    count = count+1;
    q = p0;
    p0 = p;
    p_hist = [p_hist p];
    err_hist = [err_hist abs(err)];

    if VERBOSE
        fprintf('Iteration %d: p = %.16f, err = %.16e\n', count, p, abs(err));
        fprintf('%5d %6.5f %6.4e %6.4e\n',[count,r,err,f(r)]);
    end
end
end
```