```
%%%%% Matlab program (.m) file that gives solutions to "Problem Set Zero"

%%%%% You should play with this file to learn simple Matlab commands.
%%%%% Use "help" at the command window to figure out what various commands
%%%%% do. Eg, type "help clear" to figure out what the next line does.

clear all

%%%%% QUESTION 1: STEADY STATE

%%%%% Declare parameter values

g     = 0.03;
n     = 0.01;
s     = 0.20;
alpha = 0.33;
delta = 0.04;

parameters = [g;n;s;alpha;delta];

%%%%% Declare some symbols

syms kk gg nn ss aa dd

answer = solve('(1+gg)*(1+nn)*kk-ss*kk^aa-(1-dd)*kk=0',kk);

k_bars = double(subs(answer,{gg,nn,ss,aa,dd},parameters))

% Matlab treats "symbols" differently to "numbers"; the commands above
% declare some symbols then solve the difference equation for its
% steady-state values. I then substitute the vector of parameters into the
% answer to get "numbers" back. Use the Matlab syntax "help syms", "help
% solve" etc etc to investigate this more closely.

% Keep the non-trival (positive) steady-state for future reference

k_bar  = max(k_bars);

ky = k_bar^(1-alpha);

mpk = 1+alpha*k_bar^(alpha-1);

display(['Non-trivial steady state          =   ',num2str(k_bar)])
display(['Capital/output ratio              =   ',num2str(ky)])
display(['Gross marginal product of capital =   ',num2str(mpk)])

%%%%% QUESTION 2: TRANSITION TO STEADY STATE

error = 10;
iter  = 1;

k0    = k_bar*exp(-0.5);
```

```
% This sets the initial condition while the next command sets up an empty
% vector that we will use to store the sequence of capital stocks.

kt    = [];

% The following loop iterates on the difference equation until the error is
% small (less than 10e-6)

while error > 10e-6
    if iter == 1
        k = k0;
    else
        k = kt(iter-1);
    end

    k1 = (s/((1+g)*(1+n)))*k^(alpha) + ((1-delta)/((1+g)*(1+n)))*k;
% The previsious line computes the subsequent capital stock k1 given
% today's capital stock k

    error = abs(k1-k_bar);

    kt = [kt;k1];
    iter  = iter+1;

end

T = length(kt);

t = [0;cumsum(ones(T-1,1))];

figure(1)
plot(t,kt,t,k_bar*ones(T,1),'k--')
title('Figure 1: Time path of the capital stock')
xlabel('time, t')
ylabel('capital stock, k(t)')
legend('k(t)',0)

%%%%% Now for the phase diagram (this is a little tricky...)

kmax = 2*k_bar;
ks   = linspace(0,kmax,1000);

psis = (s/((1+g)*(1+n)))*ks.^(alpha) + ((1-delta)/((1+g)*(1+n)))*ks;

kt0 = kt(1:T-1);
kt1 = kt(2:T);

figure(2)
if k0 < k_bar
    plot(kt0,kt1,'b>-',ks,ks,'k--',ks,psis,'r-')
    title('Figure 2: Phase diagram')
```

```
else
    plot(kt0,kt1,'b<-',ks,ks,'k--',ks,psis,'r-')
    title('Figure 2: Phase diagram')
end
xlabel('k(t)')
ylabel('k(t+1)')
legend('k(t+1)','45-degree line','\psi(k(t))',0)

%%%%% PROBLEM 3: GROWTH RATE APPROXIMATIONS

%%%%% The next command produces a 1000-by-1 vector of evenly spaced "zs"
%%%%% between 0 and 1.

zs = linspace(0,1,1000)';

g_approximate     = log(1+zs);
g_actuals         = zs;

figure(3)
plot(100*zs,100*g_approximate,'k--',100*zs,100*g_actuals)
title('Figure 3: Accuracy of log-differences as aproximate growth rates')
xlabel('growth rate (%)')
legend('log-approximate growth rate','actual growth rate',0)
```

```
>> ps0_solutions

k_bars =

    3.9040
         0


ans =

Non-trivial steady state          =   3.904


ans =

Capital/output ratio              =   2.4907


ans =

Gross marginal product of capital =   1.1325

>>
```

Figure 1: Time path of the capital stock

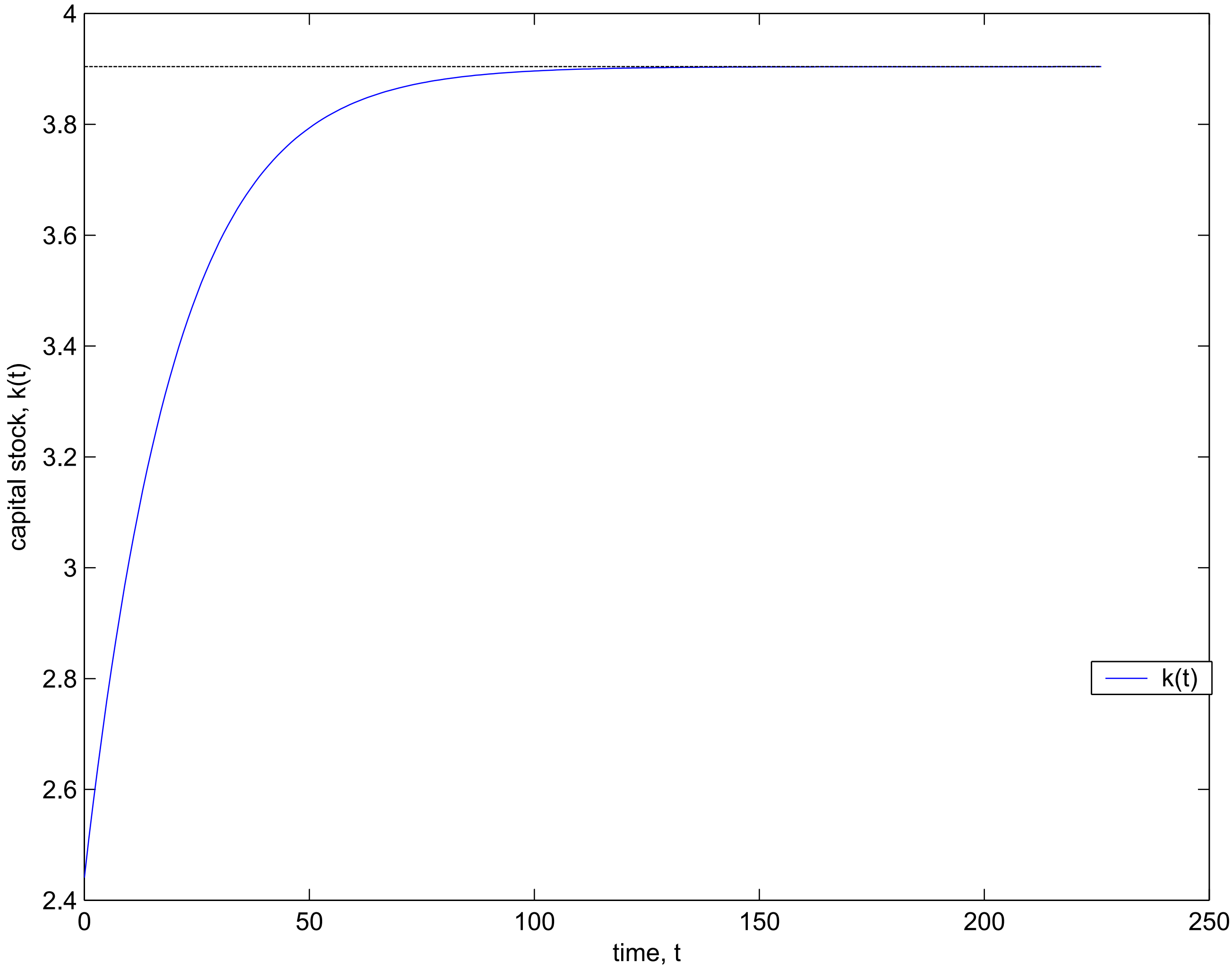

Figure 2: Phase diagram

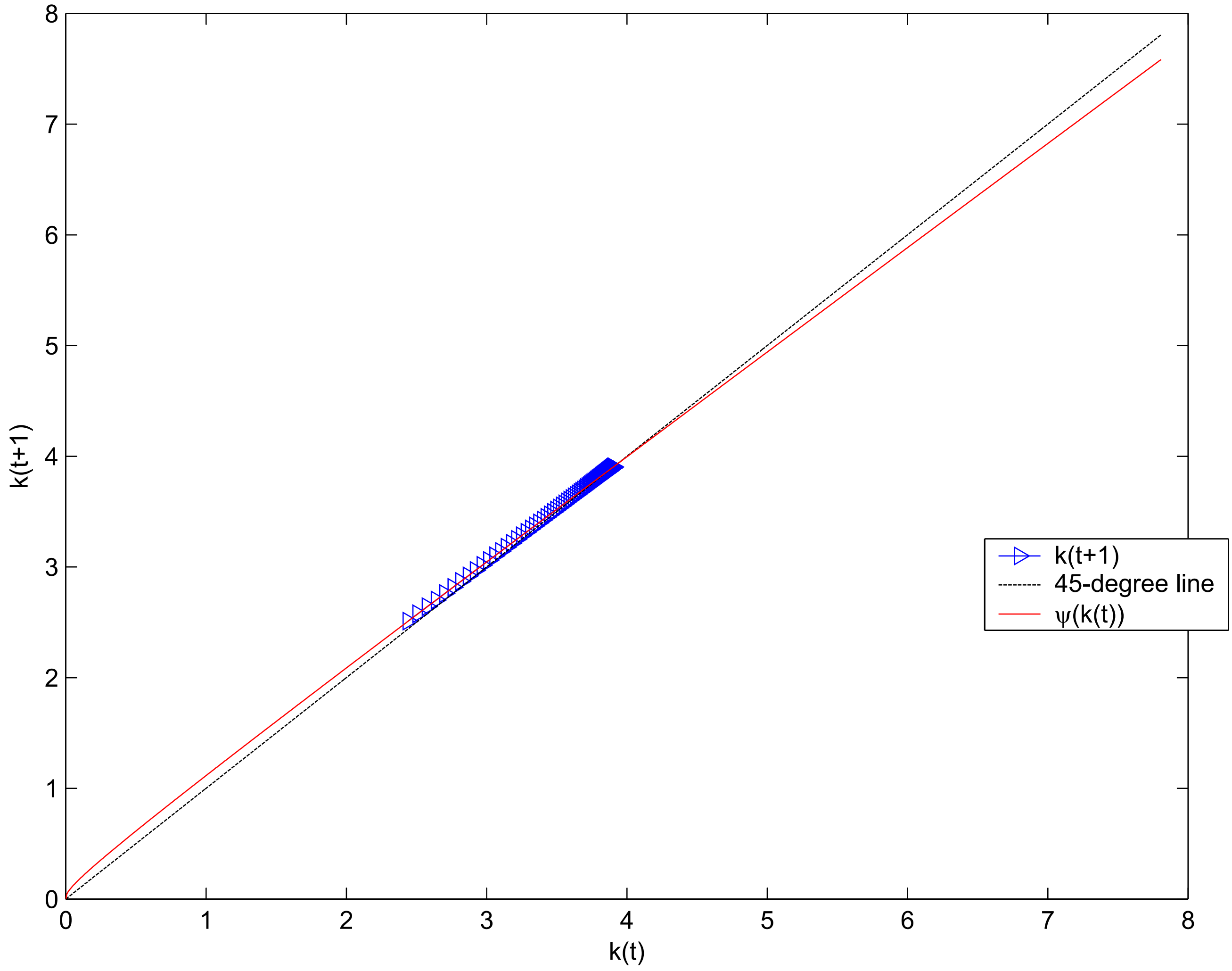

Figure 3: Accuracy of log-differences as aproximate growth rates