

---

```

%This program will solve for the transition path of capital
%from the initial level (k0) to the steady state level (kss)
%for the standard growth model. There is no endogenous labor supply decision.
%The production function is Cobb-Douglas, with a capital share of alpha.
%The utility function is  $(c^{(1-\sigma)})/(1-\sigma)$ .
%The program assumes that the initial level of capital is below the steady
%state value. Specifically, it is assumed that  $k_0 = \lambda k_{ss}$ .
%T is the number of periods for which the path is computed.

```

```

%Parameter values

```

```

delta = .08;
alpha = 1/3;
beta = .96;
sigma = 1.01;
lambda = .5;
T = 200;
A = 1;

```

```

%Steady state capital stock and initial condition

```

```

kss = ((1/beta-(1-delta))/A/alpha)^(1/(alpha-1));
srate = (delta*kss)/A*kss^alpha; %savings rate

```

## Solve transition path for the neoclassical model

```

%Solution for path ksol(t)

```

```

k0 = lambda*kss;
ksol(1)=k0;

```

```

for t=2:T

```

```

    %compute path for various choices of k(t), given ksol(t-1)
    %each guess for k(t) will generate a series that has first element
    %ksol(t-1), second element the guess and then subsequent elements given by
    %iterating on the first order conditions. We call this sequence kguess.

```

```

    kguess(1)=ksol(t-1);

```

```

    %Step 1: Set boundaries for interval that ksol(t) must lie in.

```

```

    kmin=ksol(t-1);kmax=kss;

```

```

    %Step 2: Pick a point in the interval as a hypothetical choice for k(t)

```

```

    while abs(kmax-kmin)>.00000015*kss

```

```

        kn=.5*(kmin+kmax);

```

```

        kguess(2)=kn;

```

```

        %Step 3: Determine the rest of the path given the choice for k(t)

```

```

        %stop is simply an indicator to tell us if we need to continue to the

```

```

    next

```

```

        %element of the kguess series. i is the index of the kguess series,

```

```

    with i=1 corresponding

```

```

        %to k(t-1) and i=2 corresponding to k(t).

```

```

        stop=0;

```

```

        i=2;

```

```

        while stop < 1

```

---

```

        i=i+1;
        %implied value for kguess(i)
        kguess(i)=A*kguess(i-1)^alpha+(1-delta)*kguess(i-1)-...
            (beta*(A*alpha*kguess(i-1)^(alpha-1)+(1-delta)))^(1/sigma)*...
            (A*kguess(i-2)^alpha+(1-delta)*kguess(i-2)-kguess(i-1));
        %check to see if kguess is going to zero (i.e., did we enter region I)
        if kguess(i)<=kguess(i-1),
kmin=kn;stop=1;else,kguess(i)=kguess(i);end
        %check to see if kguess is going beyond kss (i.e., did we enter region
        %III)
        if kguess(i)>kss, kmax=kn;stop=1;else,kguess(i)=kguess(i);end
    end
end
    % We have now determined (within the limits of our approximation) the
next value of k(t)
    ksol(t)=kguess(2);
end

```

## Solve the transition path for the Solow model

```

k_solow(1) = k0;
for t = 2:T
    k_solow(t) = srate * A * k_solow(t-1)^alpha + (1-delta) * k_solow(t-1);
end

```

## Plot comparison

```

clear model
model{1} = ['Model = ', 'Solow'];
model{2} = ['Model = ', 'Neo'];

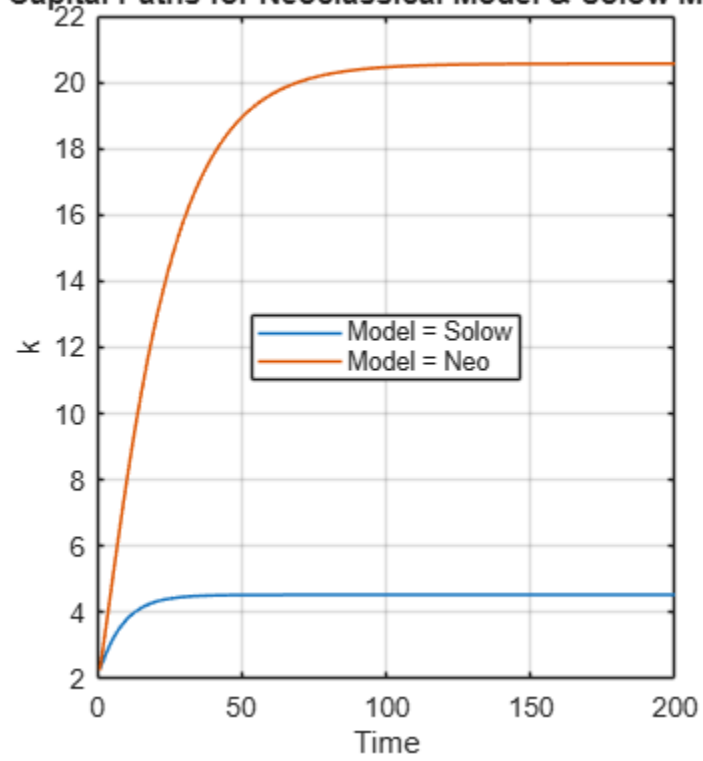
figure;
plot(1:T, ksol); % plot capital paths
hold on;
plot(1:T, k_solow);
hold off;

title('Capital Paths for Neoclassical Model & Solow Model');
xlabel('Time');
ylabel('k');
legend(model, 'Location', 'best');
grid on;

```

---

**Capital Paths for Neoclassical Model & Solow Model**



*Published with MATLAB® R2025b*