THE RAMSEY MODEL

- The planner problem:

$$\max_{\{c_s, k_{s+1}\}_{s=t}^{\infty}} U_t = \sum_{s=t}^{\infty} \tilde{\beta}^{s-t} u(c_s)$$

$$\text{s.t. } \gamma k_{t+1} = f(k_t) + (1-\delta)k_t - c_t$$

$$\tilde{\beta} \equiv \beta\gamma^{1-\mu}$$

- FOCs:

$$u'(c_t) = \hat{\lambda}_t$$

$$\tilde{\beta}\hat{\lambda}_{t+1}\left[f'\left(\hat{k}_{t+1}\right) + 1 - \delta\right] = \hat{\lambda}_t$$

$$\gamma\hat{k}_{t+1} = f\left(\hat{k}_t\right) + (1-\delta)\hat{k}_t - c_t$$

$$\lim_{j\to\infty} \beta^j \hat{\lambda}_j \hat{k}_{j+1} = 0$$

- The Euler equation:

$$\tilde{\beta} u'(c_{t+1})\left[f'\left(\hat{k}_{t+1}\right) + 1 - \delta\right] = u'(c_t)$$

- Assumptions on functional forms and parameter values:

$$u(c) = c^{1-\mu}/(1-\mu) \qquad f(k) = \phi k^{\alpha}$$

$$\beta = 0.96, \mu = 2, \alpha = 0.4, \gamma = 1.016, \delta = 0.1$$

- The value of the scale parameter φ is chosen so to make the steady-state capital level equal to unity.

- The policy function has to satisfy the following functional equation:

$$c(k)\left(\frac{\tilde{\beta}}{\gamma}\right)^{\frac{1}{\mu}}\left[\alpha\phi(k')^{\alpha-1}+1-\delta\right]^{\frac{1}{\mu}}=c(k')$$

where:

$$k'=\frac{\phi k^{\alpha}+(1-\delta)k-c(k)}{\gamma}$$

- We approximate the policy function *c(k)* over an interval $D \equiv [\underline{k}, \overline{k}] \in R_+$ with a linear combination of Chebyshev polynomials:

$$\widehat{c}(k;\boldsymbol{\theta}) = \sum_{i=0}^{d} \theta_i \psi_i(k)$$

where:

$$\psi_i(k) \equiv T_i\left(2\frac{k-\underline{k}}{\overline{k}-\underline{k}} - 1\right)$$

- In our exercise, we choose:

$$\underline{k} = 0.1,\ \overline{k} = 1.9,\ d = 15$$

- We find n zeros of Chebyshev polynomials in *[-1,1]*, reverse the normalization and transform them into the corresponding values in $[\underline{k}, \overline{k}]$.
- Then, we numerically solve the Euler equation:

$$\widehat{c}(k_i)\left(\frac{\tilde{\beta}}{\gamma}\right)^{\frac{1}{\mu}}\left[\alpha\phi\left[\frac{\phi k_i^{\alpha}+(1-\delta)k_i-\widehat{c}(k_i)}{\gamma}\right]^{\alpha-1}+1-\delta\right]^{\frac{1}{\mu}}=\widehat{c}\left[\frac{\phi k_i^{\alpha}+(1-\delta)k_i-\widehat{c}(k_i)}{\gamma}\right]$$

at these points k_i for the n parameters in θ.

```
global mu sk be dk phi g

mu=2;
sk=0.4;
be=0.96;
dk=0.1;
g=1.016;
phi=(g^mu/be-1+dk)/sk;

kss=1;
yss=phi;
css=phi+1-dk-g;

ka=0.1;
kb=1.9;
```



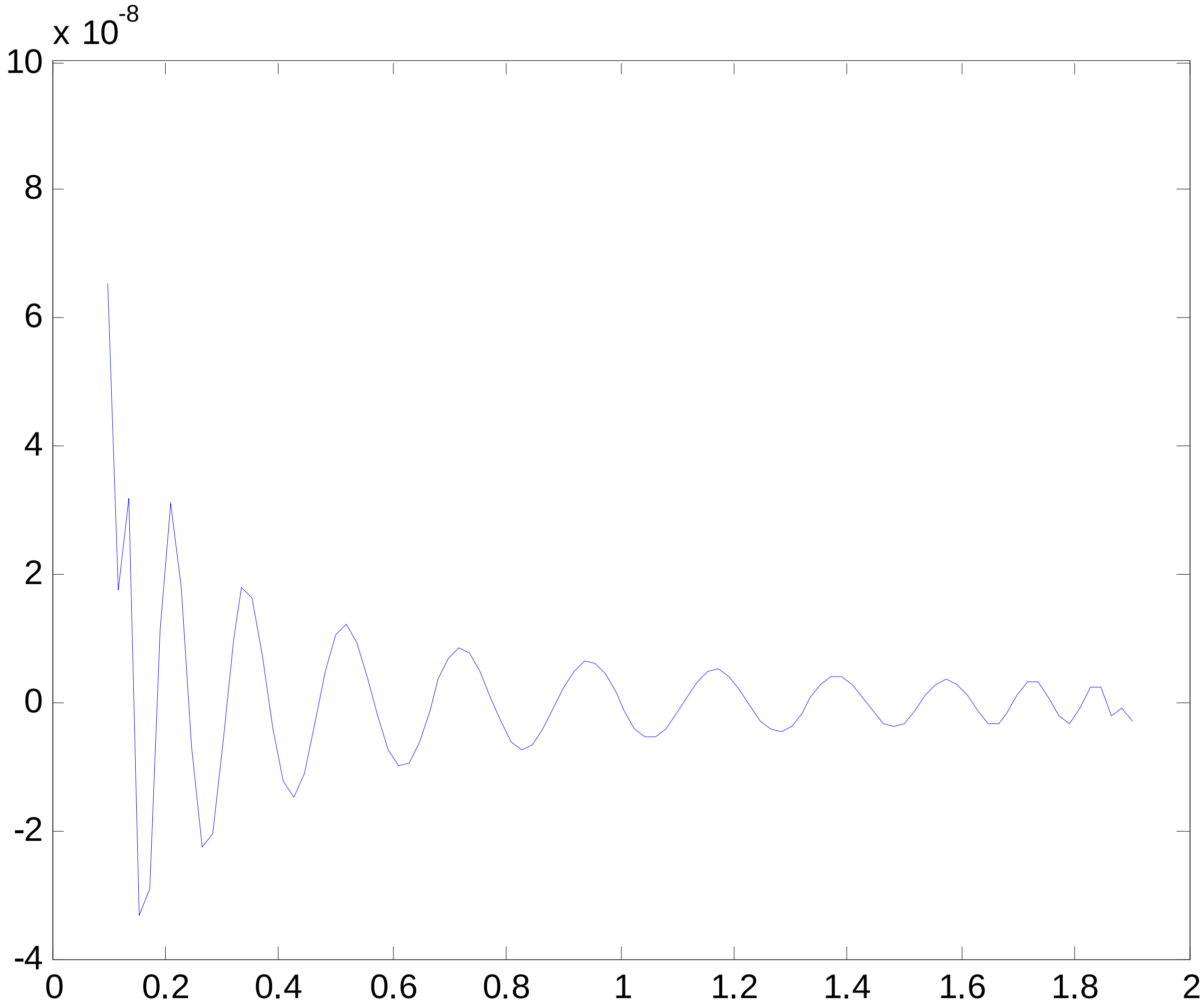
```
clear
param

ka=0.1;
kb=1.9;
d=24;

opt=[sqrt(eps) sqrt(eps) 500 2];
cf0=[0.1;0.15];

tic
cf=ColCheby('residcol',cf0,ka,kb,d,opt);
toc

[k,er]=CheckCheby1('residcol',cf,ka,kb,100);
plot(k,er)

save coef cf ka kb d
```

```
function res=ResidCol(cf,k,T,ka,kb)

global mu sk be dk phi g

c=T*cf;
y=phi.*k.^sk;
k1=((1-dk)*k+y-c)/g;
c1=ChebyPol(2*(k1-ka)/(kb-ka)-1,cf);
res=((be/g^mu)*(sk*phi.*k1.^(sk-1)+1-dk)).^(1/mu).*c-c1;
```

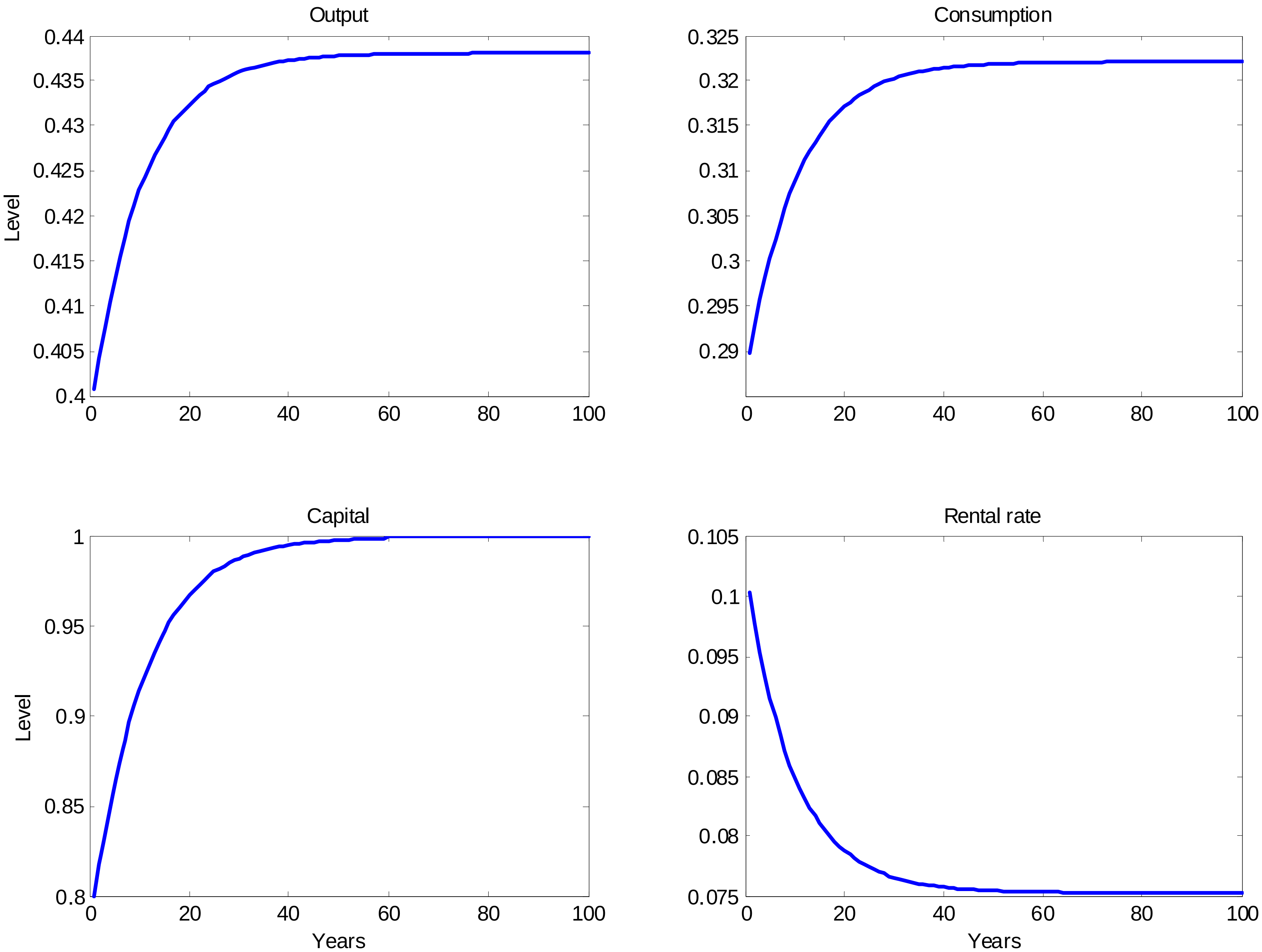
```
function [cf,er,cod]=ColCheby1(fname,cf0,a,b,d,opt,varargin)

for n=size(cf0,1):d+1
    [z,k]=chebynodes(a,b,n);
    T=cheby(z,n-1);
    [cf,cod,er]=trustsolve(fname,cf0,opt,k,T,a,b,varargin{:});
    if cod>0
        disp(['Coefs: ' num2str(n)...
             '. The solver reports: ' num2str(cod)]);
    end
    cf=real(cf);
    cf0=newguess(cf,1);
end
```

```
function [x,er]=CheckCheby1(fname,cf,a,b,n,varargin)

d=size(cf,1)-1;
[z,x]=UniformNodes(a,b,n);
T=cheby(z,d);
er=residcol(cf,x,T,a,b,varargin{:});
disp(['Avg. Abs. Er.: ' num2str(mean(abs(er)))]);
disp(['Avg. Med. Er.: ' num2str(median(abs(er)))]);
disp(['Std. Er.:      ' num2str(std(er))]);
disp(['Max. Abs. Er.: ' num2str(max(abs(er)))]);
```

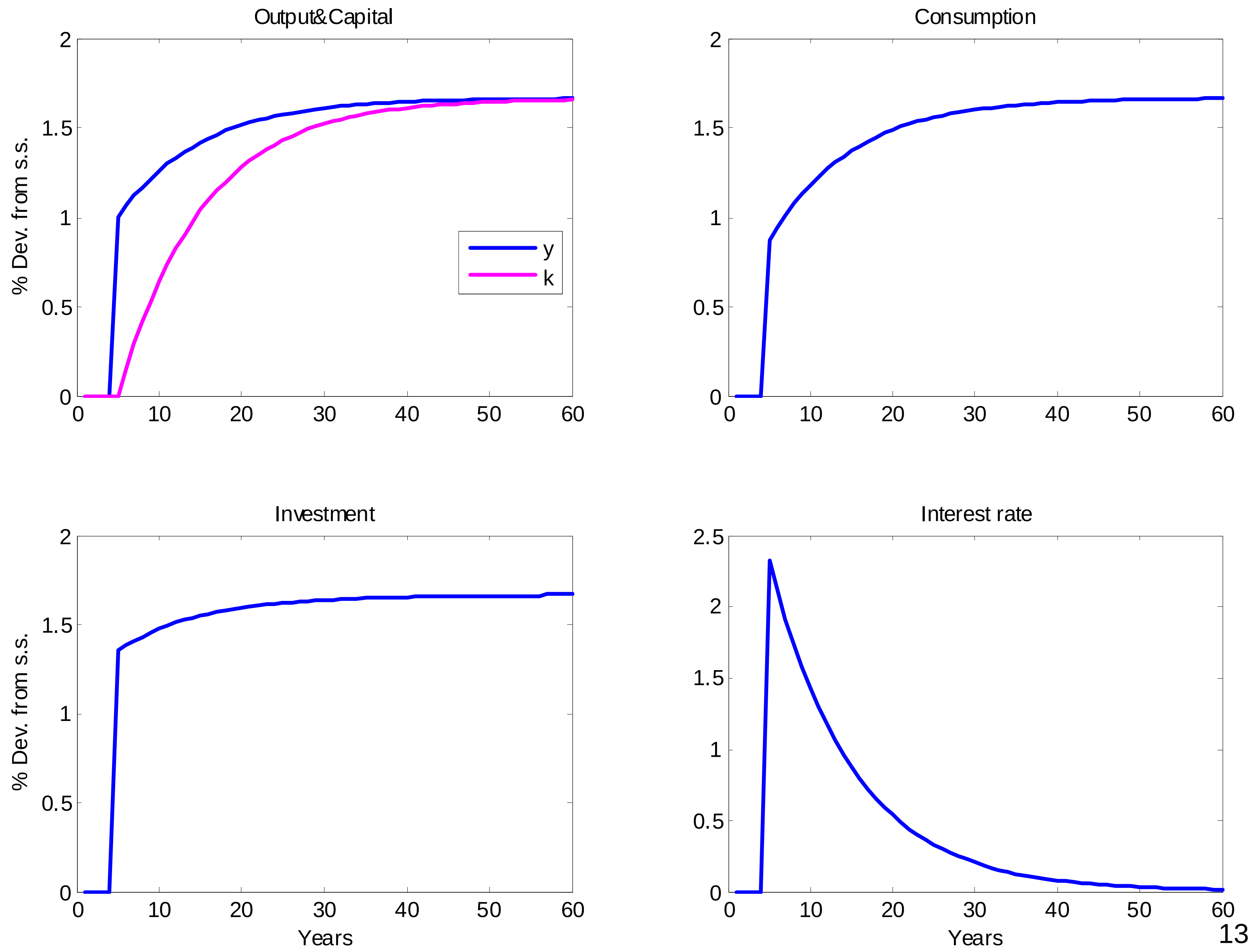
```
>> coef
Elapsed time is 0.054660 seconds.
Avg. Abs. Er.: 6.8862e-009
Avg. Med. Er.: 4.0566e-009
Std. Er.:      1.1371e-008
Max. Abs. Er.: 6.5605e-008
```




```
clear
param
load coef

h=100;

yp=1; cp=2; ip=3; kp=4; rp=5;

v=zeros(5,h+1);
v(kp,1)=0.8;

for j=1:h
    v(cp,j)=ChebyPol(2*(v(kp,j)-ka)/(kb-ka)-1,cf);
    v(yp,j)=phi*v(kp,j)^sk;
    v(ip,j)=v(yp,j)-v(cp,j);
    v(rp,j)=sk*phi*v(kp,j)^(sk-1)-dk;
    v(kp,j+1)=((1-dk)*v(kp,j)+v(ip,j))/g;
end

v=v(:,1:h);

subplot(2,2,1), plot(v(yp,:)); title('Output'); ylabel('Level')
subplot(2,2,2), plot(v(cp,:)); title('Consumption');
subplot(2,2,3), plot(v(kp,:)); title('Capital'); xlabel('Years'); ylabel('Level')
subplot(2,2,4), plot(v(rp,:)); title('Rental rate'); xlabel('Years')
```



Output
Consumption
Capital
Rental rate
Level
Level
Years
Years
0.44
0.435
0.43
0.425
0.42
0.415
0.41
0.405
0.4
0.325
0.32
0.315
0.31
0.305
0.3
0.295
0.29
1
0.95
0.9
0.85
0.8
0.105
0.1
0.095
0.09
0.085
0.08
0.075
0
20
40
60
80
100


```
clear
param
load coef

h=60; h1=5;
yp=1; cp=2; ip=3; kp=4; rp=5;

v=zeros(5,h+1);
ss1=[yss;css;yss-css;kss;sk*yss/kss-dk];
v(:,1:h1)=ss1(:,ones(1,h1));

phi=phi*1.01;

ks2=((g^mu/be-1+dk)/(phi*sk))^(1/(sk-1));
ka2=ks2*0.1;
kb2=ks2*1.9;

cf2=colcheby1('residcol',cf,ka2,kb2,length(cf)-1,[]);

for j=h1:h
    v(cp,j)=ChebyPol(2*(v(kp,j)-ka2)/(kb2-ka2)-1,cf2);
    v(yp,j)=phi*v(kp,j)^sk;
    v(ip,j)=v(yp,j)-v(cp,j);
    v(rp,j)=sk*phi*v(kp,j)^(sk-1)-dk;
    v(kp,j+1)=((1-dk)*v(kp,j)+v(yp,j)-v(cp,j))/g;
end;

v=(v(:,1:h)-ss1(:,ones(1,h)))./ss1(:,ones(1,h))*100;

subplot(2,2,1), plot(v([yp kp],:)');
legend('y','k',0); title('Output&Capital'); ylabel('% Dev. from s.s.')
subplot(2,2,2), plot(v(cp,:)'); title('Consumption');
subplot(2,2,3), plot(v(ip,:)'); title('Investment');
xlabel('Years'); ylabel('% Dev. from s.s.')
subplot(2,2,4), plot(v(rp,:)'); title('Interest rate'); xlabel('Years');
```



Output&Capital
Consumption
Investment
Interest rate
% Dev. from s.s.
% Dev. from s.s.
y
k
Years
Years
0
0.5
1
1.5
2
2.5
0
10
20
30
40
50
60

- An alternative solution method: **time iteration** on the Euler equation (simple and effective, maybe not particularly elegant)

Algorithm *Choose a suitable grid of points over an interval* $[\underline{k}, \overline{k}] \in R_+$, *say* $\mathbf{k} = \{k_i\}_{i=1}^n$, *and an initial guess for the optimal consumption levels at the nodes* k_j, *say* $\mathbf{c}_0 = \{c_{0i}\}_{i=1}^n$. *Then, for* $j \geq 0$:

1. *Given* $\mathbf{c}_j$, *compute:*

$$k'_{j,i} = \frac{\phi k_i^\alpha + (1-\delta)k_i - c_{j,i}}{\gamma}$$

2. *Given* $\mathbf{k}$, $\mathbf{c}_j$, *and* $\mathbf{k}'_j = \{k'_{j,i}\}_{i=1}^n$, *obtain* $\mathbf{c}'_j$ *via cubic interpolation (or extrapolation, if needed).*

3. *Given* $\mathbf{c}'_j$, *compute* $\hat{\mathbf{c}}_j$ *as:*

$$\hat{c}_{j,i} = c'_{j,i}\left\{\frac{\tilde{\beta}}{\gamma}\left[\alpha\phi(k'_{j,i})^{\alpha-1} + 1 - \delta\right]\right\}^{-\frac{1}{\mu}}.$$

4. *Update the current guess:*

$$\mathbf{c}_{j+1} = \upsilon\hat{\mathbf{c}}_j + (1-\upsilon)\mathbf{c}_j$$

where $\upsilon \in (0,1)$, *and iterate on (1)-(4) until convergence.*

```
param

[z,k]=ChebyNodes(ka,kb,d);
c_spline=0.1+0.15*k;

tic
dif=1;
while dif>1e-9
    c1=EulerEq(c_spline,k);
    dif=max(abs(c1-c_spline));
    c_spline=c1;
end
toc
```

```
function c2=EulerEq(c,k)

global mu sk be dk phi g

y=phi.*k.^sk;
k1=((1-dk)*k+y-c)/g;
c1=interp1(k,c,k1,'cubic');
c2=((be/g^mu)*c1.^(-mu).*(sk*phi.*k1.^(sk-1)+1-dk)).^(-1/mu);
```



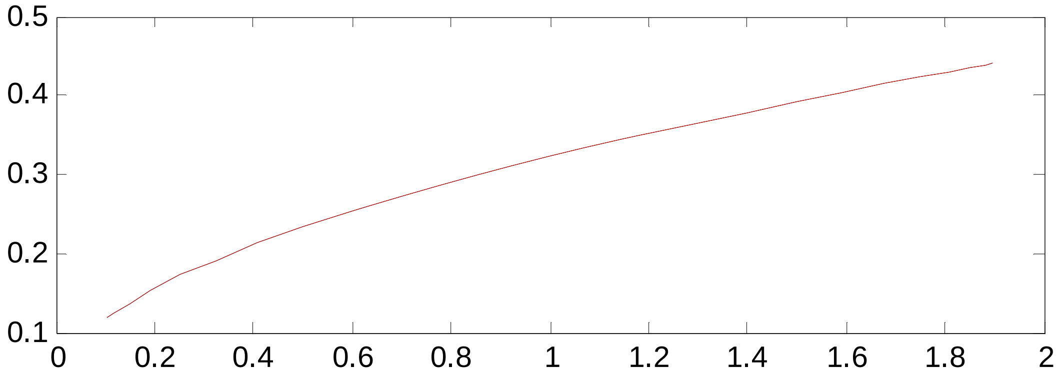
```
tic
dif=1;
while dif>1e-9
    [c1,pc]=EulerEq1(c_interp,k,d);
    dif=max(abs(c1-c_interp));
    c_interp=c1;
end
toc
```

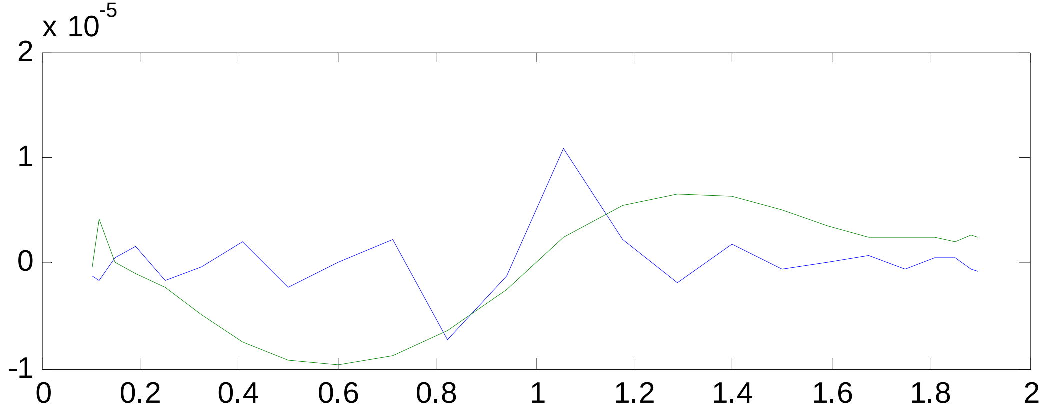
```
function [c2,pc]=EulerEq1(c,k,d)

global mu sk be dk phi g

y=phi.*k.^sk;
k1=((1-dk)*k+y-c)/g;
pc=polyfit(k,c,d);
c1=polyval(pc,k1);
c2=((be/g^mu)*c1.^(-mu).*(sk*phi.*k1.^(sk-1)+1-dk)).^(-1/mu);
```

```
[z1,k1]=uniformNodes(ka,kb,100);
c1_cheby=ChebyPol(z1,cf);
c1_spline=interp1(k,c_spline,k1,'cubic');
c1_interp=polyval(pc,k1);
er_spline=EulerEqRes(c1_spline,k1);
er_interp=EulerEqRes1(c1_interp,k1,pc);
er_cheby=EulerEqRes2(c1_cheby,cf,k1,ka,kb);

disp(['Avg. Abs. Er.: ' num2str(mean(abs([er_cheby er_spline er_interp])))]);
disp(['Avg. Med. Er.: ' num2str(median(abs([er_cheby er_spline er_interp])))]);
disp(['Std. Er.:      ' num2str(std([er_cheby er_spline er_interp]))]);
disp(['Max. Abs. Er.: ' num2str(max(abs([er_cheby er_spline er_interp])))]);

subplot(2,1,1), plot(k,[c_cheby,c_interp,c_spline]);
subplot(2,1,2), plot(k,[c_cheby-c_interp,c_cheby-c_spline]);
```

```
function res=EulerEqRes(c,k)

global mu sk be dk phi g

y=phi.*k.^sk;
k1=((1-dk)*k+y-c)/g;
c1=interp1(k,c,k1,'cubic');
res=((be/g^mu)*(sk*phi.*k1.^(sk-1)+1-dk)).^(1/mu).*c-c1;
```

```
>> iter
Elapsed time is 0.045682 seconds.
Elapsed time is 0.039925 seconds.
Avg. Abs. Er.: 6.8862e-009 2.0317e-006  1.827e-006
Avg. Med. Er.: 4.0566e-009 1.2206e-006 1.5559e-006
Std. Er.:      1.1371e-008 3.3935e-006 2.3656e-006
Max. Abs. Er.: 6.5605e-008 2.1499e-005 8.4023e-006
```





THE BROCK-MIRMAN MODEL

Assume that TFP, denoted $a_t \in R_{++}$, follows a stationary Markov process. In particular, assume that the natural logarithm of a_t follows an AR(1) process of the form $\ln(a_{t+1}) = \rho \ln(a_t) + \varepsilon_t$, where $0 < \rho < 1$ is the persistence parameter and $\varepsilon_t \sim N(0, \sigma^2)$ the *iid* innovation.

The planner solves the following stochastic optimal control problem, taking the initial condition $\{k_t, a_t\}$ and the stochastic process for a_t as given:

$$\max_{\{c_s, k_{s+1} \mid a_t\}_{s=t}^{\infty}} U_t = E_t\left(\sum_{s=t}^{\infty} \tilde{\beta}^{s-t} \frac{c_s^{1-\mu}}{1-\mu} \right)$$

$$s.t. \;\; \gamma k_{t+1} = a_t k_t^{\alpha} + (1-\delta) k_t - c_t$$

where $\tilde{\beta} = \beta \gamma^{1-\mu} \in (0,1)$, $\alpha \in (0,1)$, and $\gamma > 1$.

As usual, we build a Lagrangian and derive it with respect to c_t, k_{t+1}, and λ_t, to obtain the following first order conditions:

$$c_t^{-\mu} = \lambda_t$$

$$E_t[\alpha\lambda_{t+1}a_{t+1}k_{t+1}^{\alpha-1} + \lambda_{t+1}(1-\delta)] = \varphi\lambda_t$$

$$\gamma k_{t+1} = a_t k_t^{\alpha} + (1-\delta)k_t - c_t$$

where $\varphi \equiv \gamma/\tilde{\beta}$.

We exploit the model's recursive structure to approximately solve the following functional equation for the policy function $c(k,a)$:

$$E\left[c\left(k', \mathrm{e}^{\ln(a')}\right)^{-\mu}\left[\alpha\mathrm{e}^{\ln(a')}(k')^{\alpha-1} + 1 - \delta\right] \mid k, \mathrm{e}^{\ln(a)}\right] = \varphi c(k, \mathrm{e}^{\ln(a)})^{-\mu}$$

where:

$$k' = \frac{\mathrm{e}^{\ln(a)}k^{\alpha} + (1-\delta)k - c(k,\mathrm{e}^{\ln(a)})}{\gamma}$$

$$\ln(a') = \rho\ln(a) + \varepsilon$$

$$\varepsilon \sim N(0,\sigma^2)$$

The policy function is approximated over $D \equiv [\underline{k}, \overline{k}] \times [\underline{a}, \overline{a}]$ with a linear combination of multidimensional basis functions taken from a *2*-fold tensor product of Chebyshev polynomials:

$$\widehat{c}(k, a; \boldsymbol{\theta}) = \sum_{i=0}^{d} \sum_{j=0}^{d} \theta_{ij} \psi_{ij}(k, a)$$

where:

$$\psi_{ij}(k, a) \equiv T_i\left(2\frac{k - \underline{k}}{\overline{k} - \underline{k}} - 1\right) T_j\left(2\frac{a - \underline{a}}{\overline{a} - \underline{a}} - 1\right)$$

Given that $\ln(a') = \rho \ln(a) + \sigma\sqrt{2}\, z$, where $z \sim N(0, 1)$, the Euler equation (ref: b38) becomes:

$$\int_{-\infty}^{\infty} c(k', \mathrm{e}^{\rho \ln(a) + \sigma z}; \boldsymbol{\theta})^{-\mu} \left[\alpha \mathrm{e}^{\rho \ln(a) + \sigma z} (k')^{\alpha - 1} + 1 - \delta\right] \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} dz =$$

$$\varphi c(k, \mathrm{e}^{\ln(a)}; \boldsymbol{\theta})^{-\mu}$$

where:

$$k' = \frac{\mathrm{e}^{\ln(a)} k^{\alpha} + (1 - \delta)k - c(k, \mathrm{e}^{\ln(a)}; \boldsymbol{\theta})}{\gamma}$$

The integral in (ref: euler) can be numerically approximated using the Gauss-Hermite quadrature formula:

$$\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} c(k', \mathrm{e}^{\rho\ln(a)+\sigma z}; \theta)^{-\mu}\left[\alpha \mathrm{e}^{\rho\ln(a)+\sigma z}(k')^{\alpha-1} + 1 - \delta\right] e^{-\frac{z^2}{2}} dz =$$

$$\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\infty} c(k', \mathrm{e}^{\rho\ln(a)+\sigma z}; \theta)^{-\mu}\left[\alpha \mathrm{e}^{\rho\ln(a)+\sigma z}(k')^{\alpha-1} + 1 - \delta\right] e^{-z^2} dz \approx$$

$$\frac{1}{\sqrt{\pi}}\sum_{j=1}^{n} a_j c(k', \mathrm{e}^{\rho\ln(a)+\sigma z_j}; \theta)^{-\mu}\left[\alpha \mathrm{e}^{\rho\ln(a)+\sigma z_j}(k')^{\alpha-1} + 1 - \delta\right]$$

where the z_j's and the a_j's are respectively the Gauss-Hermite quadrature nodes and weights.

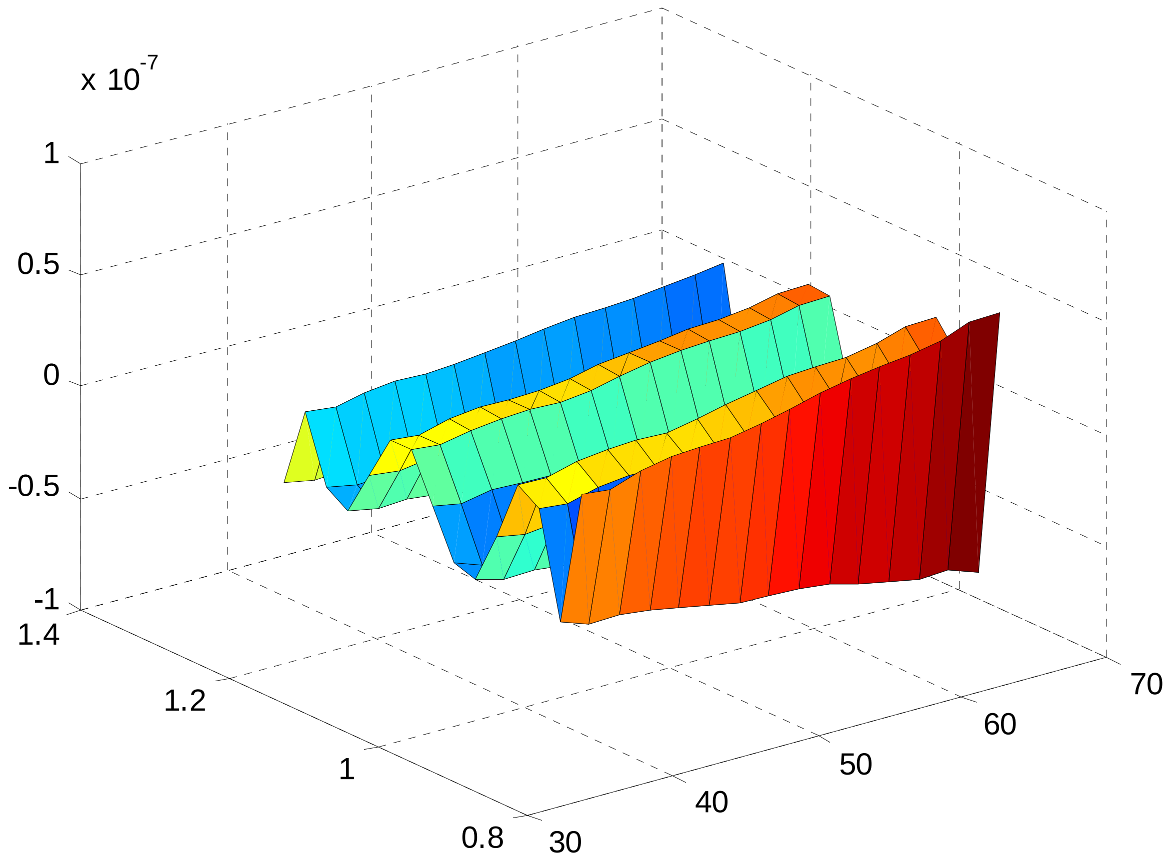
```
global sk be mu dk g rho sigma nd

mu=2;
sk=0.4;
be=0.99;
dk=0.025;
rho=0.95;
g=1.004;
sigma=0.007;

kss=((g/be-1+dk)/sk)^(1/(sk-1));
rky=sk/(g/be-1+dk);
yss=kss/rky;
sc=1-(g-1+dk)*rky;
css=sc*yss;
iss=yss-css;

ka=kss*0.7;
kb=kss*1.3;
oma=0.8;
omb=1.2;
```

```
clear
param

d=6;

cf0=[4.6;0.35;0.58;0.05];

tic
cf=colcheby2('residcol',cf0,ka,kb,oma,omb,d,[]);
toc

[k,om,er]=checkcheby2('residcol',cf,ka,kb,oma,omb,15);
surf(k,om,er)

save coef cf
```



```
function res=ResidCol(cf,sv,T,ka,kb,oma,omb)

global sk be mu dk g rho sigma

c=T*cf;
k=sv(:,1);
om=sv(:,2);
k1=((1-dk)*k+om.*k.^sk-c)/g;
m=length(k1);
[n,w]=nodes(11,'h');
ex=zeros(m,1);
for j=1:m
    k1z=k1(j)*ones(11,1);
    om1z=exp(rho*log(om(j))+sigma*sqrt(2)*n);
    c1=chebypol([2*(k1z-ka)/(kb-ka)-1 2*(om1z-oma)/(omb-oma)-1],cf);
    ex(j)=(1/sqrt(pi))*w'*((sk*om1z.*k1z.^(sk-1)+1-dk)./(c1.^mu));
end
res=c-((be/g)*ex).^(-1/mu);
```

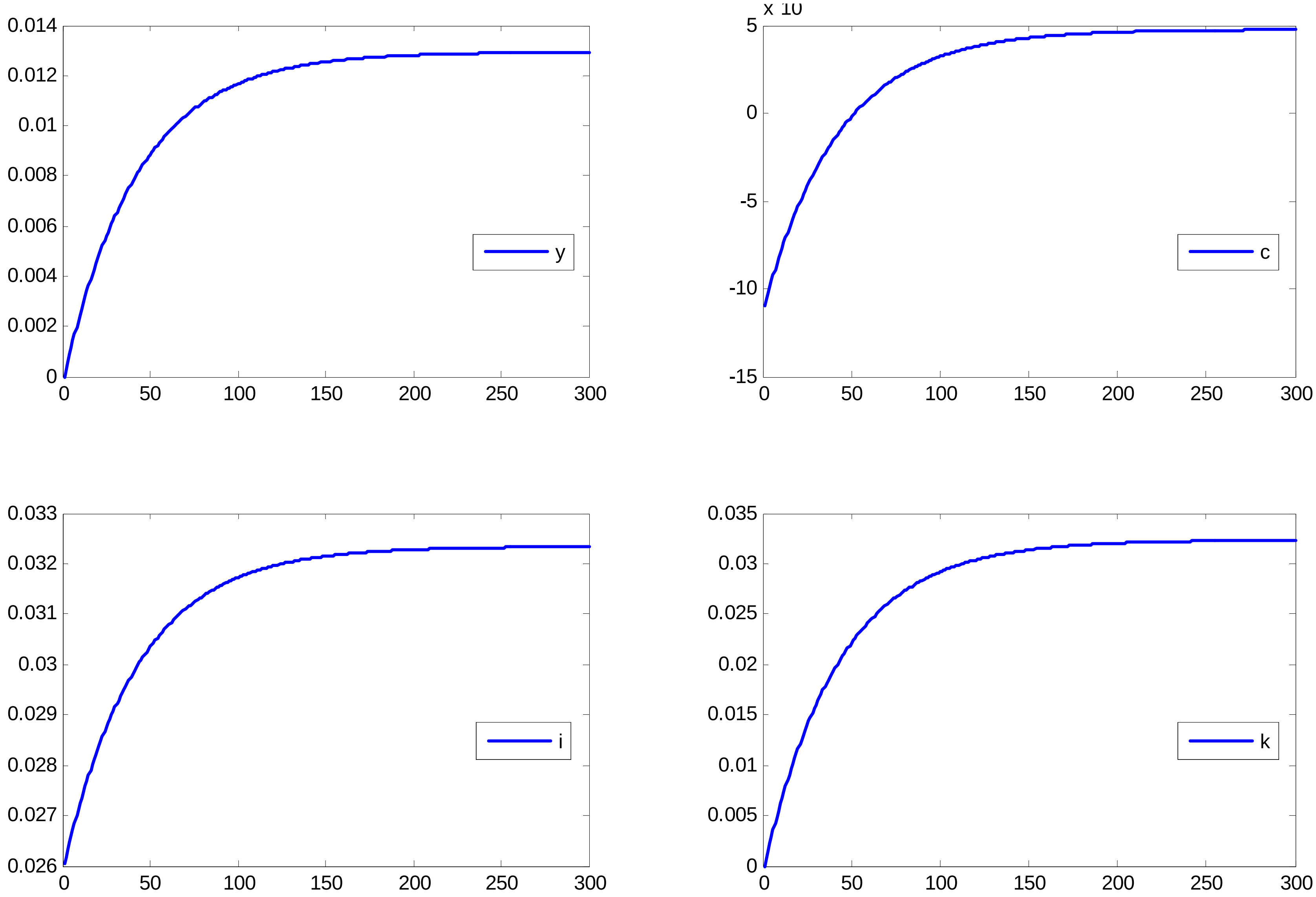
```
Elapsed time is 1.732000 seconds.
Avg. Abs. Er.: 2.6184e-008
Avg. Med. Er.: 2.7443e-008
Std. Er.: 3.0476e-008
Max. Abs. Er.: 6.7357e-008
```



```
clear
param
load coef

h=300;

yp=1; cp=2; ip=3; kp=4; omp=5;
v=zeros(5,h+1);

v(kp,1)=kss;
v(omp,1)=1;

for j=1:h
    zv=[2*(v(kp,j)-ka)/(kb-ka)-1 2*(v(omp,j)-oma)/(omb-oma)-1];
    v(cp,j)=chebypol(zv,cf);
    v(yp,j)=v(omp,j)*v(kp,j)^sk;
    v(ip,j)=v(yp,j)-v(cp,j);
    v(kp,j+1)=((1-dk)*v(kp,j)+v(ip,j))/g;
    v(omp,j+1)=exp(rho*log(v(omp,j)));
end;

v=v(:,1:h);
save simul_out v h
impulse
```

```
load simul_out

t=1:h;

subplot(2,2,1), plot(t,(v(yp,:)/yss-1)*100); legend('y',0);
subplot(2,2,2), plot(t,(v(cp,:)/css-1)*100); legend('c',0);
subplot(2,2,3), plot(t,(v(ip,:)/iss-1)*100); legend('i',0);
subplot(2,2,4), plot(t,(v(kp,:)/kss-1)*100); legend('k',0);
```

```
load simul_out

t=1:h;

subplot(2,2,1), plot(t,v(yp,:)); legend('y',0);
subplot(2,2,2), plot(t,v(cp,:)); legend('c',0);
subplot(2,2,3), plot(t,v(ip,:)); legend('i',0);
subplot(2,2,4), plot(t,v(kp,:)); legend('k',0);

pause

subplot(2,2,1), plot(t,(v(yp,:)/yss-1)*100); legend('y',0);
subplot(2,2,2), plot(t,(v(cp,:)/css-1)*100); legend('c',0);
subplot(2,2,3), plot(t,(v(ip,:)/iss-1)*100); legend('i',0);
subplot(2,2,4), plot(t,(v(kp,:)/kss-1)*100); legend('k',0);
```


0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0 50 100 150 200 250 300
y
x 10
5
0
-5
-10
-15
0 50 100 150 200 250 300
c
0.033
0.032
0.031
0.03
0.029
0.028
0.027
0.026
0 50 100 150 200 250 300
i
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0 50 100 150 200 250 300
k

TIME-CONSISTENT OPTIMAL FISCAL POLICY

- The household's Bellman equation:

$$
\begin{aligned}
v(k,K) &= \max_{c} u[c, G(K)] + \beta v(k', K') \\
\text{s.t.} \quad k' &= k + [1-\tau(K)][w(K) + [r(K)-\delta]k] - c \\
K' &= \mathcal{H}(K)
\end{aligned}
$$

- FOC:

$$u_c = \beta v'_k$$

- Envelope condition:

$$v_k = \beta v'_k[1 + (1-\tau)(r-\delta)]$$

- Euler equation:

$$u_c = \beta u'_c[1 + (1-\tau')(r'-\delta)]$$

- The government Bellman equation:

$$\begin{aligned} V(K) &= \max_{\tau} u[C(K), G(K,\tau)] + \beta V(K') \\ \text{s.t.} \quad K' &= (1-\delta)K + f(K) - C(K) - G(K,\tau) \\ G(K,\tau) &= \tau[f(K) - \delta K] \end{aligned}$$

- FOC:

$$u_G = \beta V'_K$$

- Envelope condition:

$$V_K = u_C C_K + u_G G_K + \beta V'_K (f_K + 1 - \delta - C_K - G_K)$$

- Generalized Euler equation:

$$u_G = \beta[u'_C C'_K + u'_G (f'_K + 1 - \delta - C'_K)]$$

- In equilibrium:

$$
\begin{aligned}
u_C &= \beta u'_C[1 + (1 - \tau')(f'_K - \delta)] \\
u_G &= \beta[u'_C C'_K + u'_G(f'_K + 1 - \delta - C'_K)] \\
K' &= (1 - \delta)K + f - C - G \\
G &= \tau(f - \delta K)
\end{aligned}
$$

- Solution:

$$\tau(K)\ ,\ C(K)$$

- Assumptions on functional forms:

$$u(C,G) = \frac{C^{1-\sigma}G^{\gamma(1-\sigma)} - 1}{1-\sigma} \qquad f(K) = AK^{\alpha}$$

- We approximate the policy functions *C(K)* and *τ(K)* over a rectangle $D \equiv [\underline{k}, \overline{k}] \in R_+$ with a linear combination of Chebyshev polynomials:

$$\widehat{C}(K;\boldsymbol{\theta}_C) = \sum_{i=0}^{d} \theta_{C,i}\psi_i(K)$$

$$\widehat{\tau}(K;\boldsymbol{\theta}_\tau) = \sum_{i=0}^{d} \theta_{\tau,i}\psi_i(K)$$

where:

$$\psi_i(K) \equiv T_i\left(2\frac{K-\underline{K}}{\overline{K}-\underline{K}} - 1\right)$$

- In our exercise, we choose:

$$\underline{k} = 2.38,\ \overline{k} = 6.38,\ d = 10$$

- Note that:

$$\widehat{C_K}(K;\boldsymbol{\theta}_C) = \sum_{i=0}^{d} \theta_{C,i}\psi_i'(K) =$$

$$\frac{2}{\overline{K}-\underline{K}}\sum_{i=0}^{d} \theta_{C,i}T_i'\left(2\frac{K-\underline{K}}{\overline{K}-\underline{K}}-1\right)$$

and that:

$$T_n'(x) \equiv \frac{n\sin[n\arccos(x)]}{\sin[\arccos(x)]}, \quad n = 0,1,\ldots$$

```
function Tp=DifCheby(x,d,r)

[m,k]=size(x);
if (k~=1)&((r<1)|(r>k))
    error('Error: r<1 or r>k!!')
end
d=round(d);
x=acos(x);
D=0:d;
D=D(ones(m,1),:);
if k==1
    S=sin(x);
    S=S(:,ones(1,d+1));
    Tp=(D.*sin(D.*x(:,ones(1,d+1)))))./S;
else
    C=zeros(m,d+1,k);
    for j=1:k
        if j==r
            S=sin(x(:,j));
            S=S(:,ones(1,d+1));
            C(:,:,j)=(D.*sin(D.*x(:,j*ones(1,d+1)))))./S;
        else
            C(:,:,j)=cos(D.*x(:,j*ones(1,d+1)));
        end
    end
    Tp=C(:,:,k);
    q=(1:d+1)';
    for i=k-1:-1:1
        z=repmat(Tp,1,d+1);
        q1=q(:,ones(1,size(Tp,2)))';
        Tp=C(:,q1(:),i).*z;
    end
end
```

```
function p=DifChebyPol(x,c,a,b,r)

n=size(c,1)^(1/size(x,2))-1;
p=(2/(b-a))*real(difcheby(x,n,r)*c);
```

```
global A sigma sk dk be zeta

A=1;
sigma=1;
sk=0.36;
dk=0.09;
be=0.96;
zeta=0.2;

ka=2.38;
kb=6.38;
```

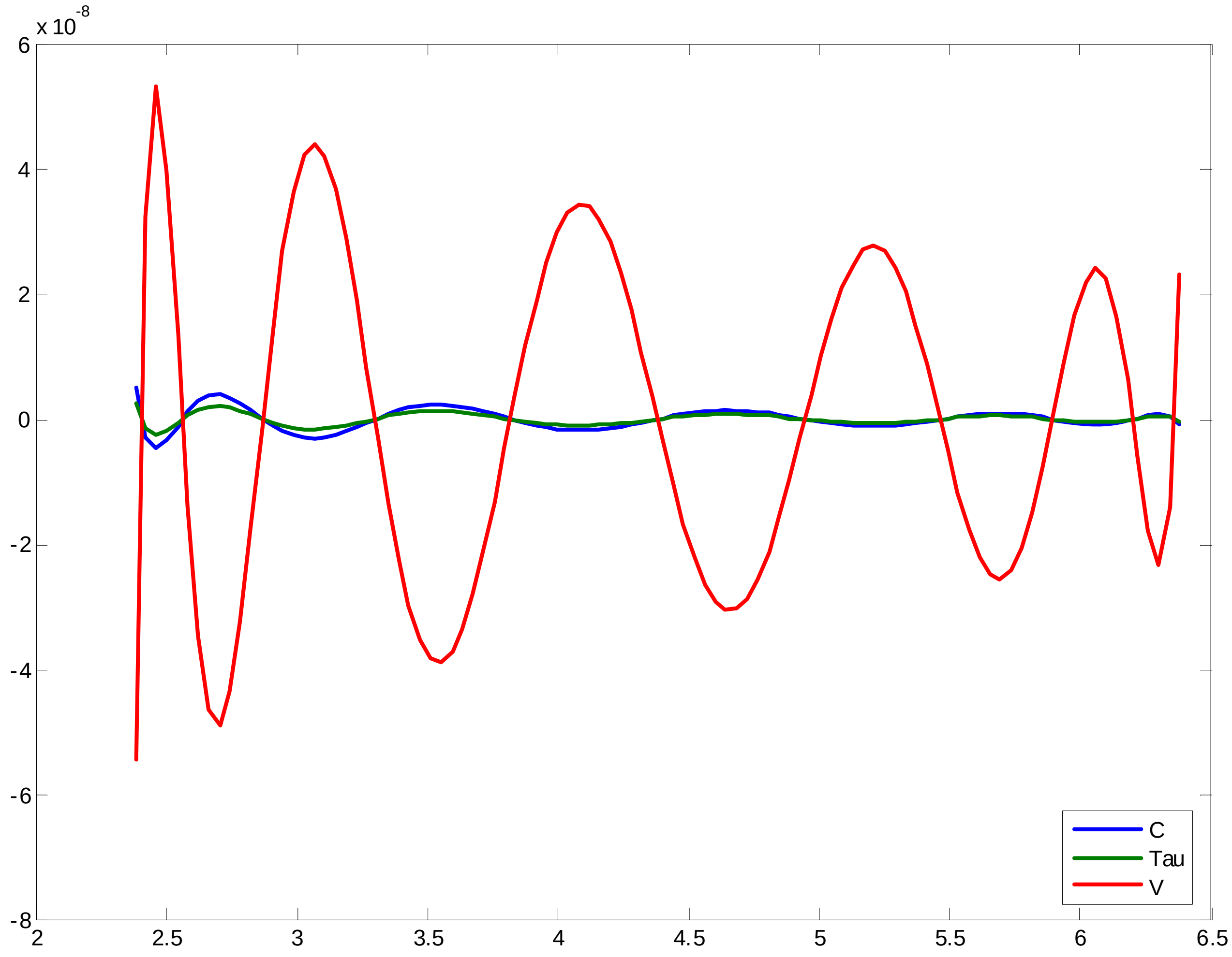
```
clear
param

d=10;

cf0=[0.95 0.14 -6;0.16 0.02 2];

tic
cf=ColCheby1('residcol',cf0,ka,kb,d,[]);
toc
[k,er]=CheckCheby1('residcol',cf,ka,kb,100);

plot(k,er);

save coef cf d
```

```
function res=ResidCol(cf,k,T,ka,kb)

global be dk

cv=T*cf;
c=cv(:,1);
t=cv(:,2);
V=cv(:,3);
[y,r,g]=Vars(k,t);
[u,uc,ug]=MUtility(c,g);
k1=(1-dk)*k+y-c-g;
q1=2*(k1-ka)/(kb-ka)-1;
cv1=chebypol(q1,cf);
c1=cv1(:,1);
t1=cv1(:,2);
V1=cv1(:,3);
c1p=difchebypol(q1,cf(:,1),ka,kb,1);
[y1,r1,g1]=Vars(k1,t1);
[u1,uc1,ug1]=MUtility(c1,g1);

res=[be*uc1.*(1+(1-t1).*(r1-dk))-uc,...
     be*(uc1.*c1p+ug1.*(r1+1-dk-c1p))-ug,...
     u+be*V1-V];
```

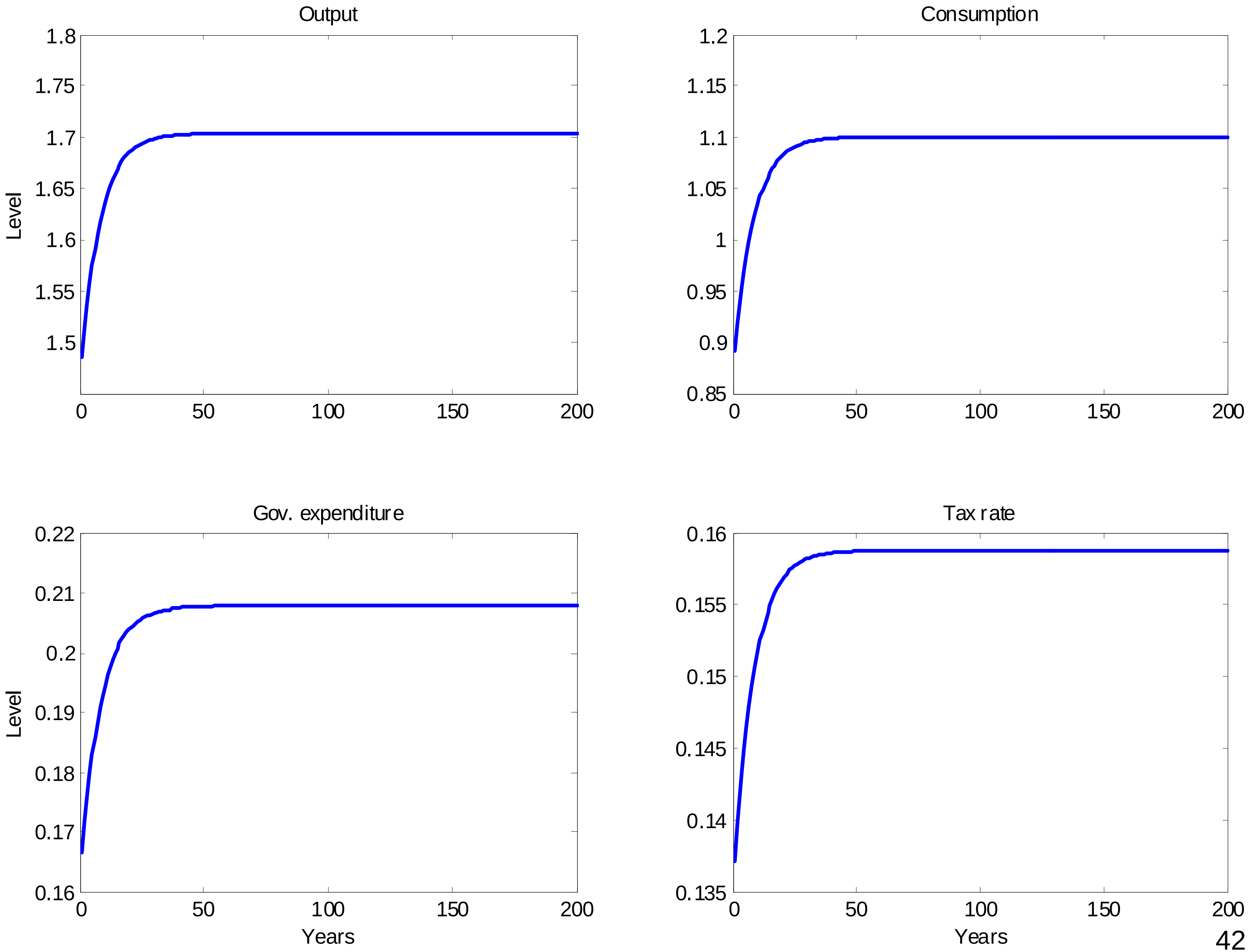
```
function [y,r,g]=Vars(k,t)

global A dk sk

y=A*k.^sk;
r=sk*y./k;
g=t.*(y-dk*k);
```

```
function [u,uc,ug]=MUtility(c,g)

global zeta sigma

uc=c.^(-sigma).*g.^(zeta*(1-sigma));
ug=zeta*c.^(1-sigma).*g.^(zeta*(1-sigma)-1);
if sigma==1
    u=log(c)+zeta*log(g);
else
    u=((c.*g.^zeta).^(1-sigma)-1)/(1-sigma);
end
```

```
>> coef
Elapsed time is 0.203000 seconds.
Avg. Abs. Er.: 1.2305e-009  6.475e-010 2.2323e-008
Avg. Med. Er.: 9.1506e-010  4.939e-010 2.2293e-008
Std. Er.:      1.6162e-009 8.4171e-010 2.5637e-008
Max. Abs. Er.: 5.0417e-009 2.5669e-009 5.4526e-008
```



```
clear
param
load coef

h=200;

yp=1; cp=2; ip=3; kp=4; rp=5; gp=6; tp=7; Vp=8;

v=zeros(8,h+1);
v(kp,1)=3;

for j=1:h
    q=2*(v(kp,j)-ka)/(kb-ka)-1;
    v(cp,j)=chebypol(q,cf(:,1));
    v(tp,j)=chebypol(q,cf(:,2));
    v(Vp,j)=chebypol(q,cf(:,3));
    [v(yp,j),v(rp,j),v(gp,j)]=Vars(v(kp,j),v(tp,j));
    v(ip,j)=v(yp,j)-v(cp,j)-v(gp,j);
    v(kp,j+1)=(1-dk)*v(kp,j)+v(ip,j);
end;

v=v(:,1:h);

subplot(2,2,1), plot(v(yp,:)); title('Output'); ylabel('Level')
subplot(2,2,2), plot(v(cp,:)); title('Consumption')
subplot(2,2,3), plot(v(gp,:));
title('Gov. expenditure'); xlabel('Years'); ylabel('Level')
subplot(2,2,4), plot(v(tp,:)); title('Tax rate'); xlabel('Years')
```



Output
Consumption
Gov. expenditure
Tax rate
Level
Level
Years
Years
1.8
1.75
1.7
1.65
1.6
1.55
1.5
1.2
1.15
1.1
1.05
1
0.95
0.9
0.85
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.16
0.155
0.15
0.145
0.14
0.135
0
50
100
150
200