Exercises for Practical DSGE Modelling

Alina Barnett
Martin Ellison

Bank of England, December 2005

1. Interactive MATLAB

Launch MATLAB by double clicking on the appropriate icon. In this first exercise we will use MATLAB in interactive mode, which means it behaves similarly to a pocket calculator. After opening, the screen is divided into three parts. In the top right-hand corner is the command window. It is in here that interactive commands can be typed in and executed immediately. The top left-hand window shows the current values of all the variables in the system – it is a glorified version of the memory button on a calculator.

1. Enter the command **A = [1, 2; 3, 4]** to define the matrix $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$. Print the matrix by typing **A** in the command window. You should also be able to see A appear as a current variable in the system. If you click on it you can see its values.

2. Define a second matrix **B = [0.5, 0.6; 1, 1.5]**. What are the values of **C = A*B**, **C = B*A**, **C = A.*B** and **C = A.^B**?

3. Find the inverse of the matrix $D = \begin{pmatrix} 2 & 2 \\ 1 & 2 \end{pmatrix}$. To look up the function for inverting matrices type **help matfun**. What is the inverse of $D = \begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix}$?

4. Use the colon operator to define a row vector **X** with elements starting from 0 and proceeding to 100 in steps of 1. The syntax of the command is (start)**:**(increment)**:**(finish). Check in the memory section that **X** has been defined correctly.

5. Define a second vector **Y** that contains the squares of the values in the elements of **X**. Plot the graph of **X** against **Y** using the command **plot(x,y);**

2. Drawing from a normal distribution

In this exercise we use MATLAB's ability to draw random numbers form a normal distribution. The numbers drawn are not completely random (for that we would need to sit for hours rolling a die or tossing a coin) but to all intents and purposes they can be thought of as random. In fact, there are only 2^1492 random numbers in the computer, but for most purposes this is more than sufficient.

1. Define a vector **Z** of 100 random numbers by the command **Z = randn(100,1)**. The numbers should have mean zero and unit variance. What is the mean and variance of the simulated series? To find the functions for mean and variance use **help datafun**.

2. We are now ready to start programming in MATLAB. Open a new M-file from the File menu and include just the command **Z = randn(100,1)**. Now select debug and run from the run menu. The end result is just the same as if you had used MATLAB interactively.

3. The next task is to calculate the mean of the simulated series recursively, i.e. first calculate the mean of the first element of **Z**, then the mean of the first two elements of **Z**, the first three, and continuing up until the last calculation of the mean of all 100 simulated numbers. In MATLAB, the easiest way to do this is to begin by defining the random numbers **Z**. Next, use a **for i=1:100 … end** loop to calculate the means recursively. Each time the loop is executed the mean has to be calculated for **Z(1:i)** and stored in memory. Complete the exercise by plotting the recursive estimates of the mean.

4. Repeat the exercise above but for recursive estimation of the variance of the simulated series. Plot the recursive estimates on the same graph as the mean using the **plot(x,y,'r')**;**hold on;** plot option. Which estimate converges quickest, the mean or the variance?

5. How many numbers do you need to simulate before the simulated mean is within 0.01 of the theoretical mean of zero? To program this exercise it is easier to use a **while … end** loop, in which the condition for continuing the loop is that the absolute value of the estimated mean is greater than 0.01. Compare you results with the convergence of the variance estimates. (In this part you may find it useful to start your program with the command **randn('state',0)**. This forces the computer to use the same random numbers each time you run the program so guarantees that the results stay the same).

3. Simulating an AR(1) process

In some models it is assumed that the interest rate followed an AR(1) process of the form $\hat{i}_t = \rho \hat{i}_{t-1} + v_t$. This exercise is designed to simulate the behaviour of interest rates under such an assumption.

1. Define the shocks v_t to the interest rate in the same way as before. To have shocks with variance 0.1, we need to write **vt = 0.1^0.5*randn(100,1)**.

2. Simulate the series for $\hat{i}_t$ using a **for … end** loop with different values of the persistence parameter ρ. Use a value of zero for $\hat{i}_{-1}$.

3. What are the theoretical mean and variance of $\hat{i}_t$? How do the simulated mean and variance compare with the theoretical values?

4. Simulating a Markov-switching process

Another simple way in which interest rates might be set is to assume they follow a Markov-switching process. In this scenario, interest rates switch between a discrete set of values, with switches occurring randomly. The simplest case is one with only two states and symmetric probabilities of switching between the states. In this exercise we will simulate this process and examine its behaviour.

1. Suppose that the central bank switches interest rates between the levels of -1% and +1%. The probability of switching from either -1% to +1% is 5%, i.e. 0.05. Simulate the behaviour of interest rates for 500 periods. The easiest way to do this is again with a **for … end** loop from 1 to 500. To assess whether there has been a switch each period, make a random draw from a uniform distribution using **rand(1,1)**. If this number is greater than 1-0.05 = 0.95 then the process should switch. This will ensure that switches occur on average 5% of the time. A neat trick to use here is that when there is a switch, the interest rate is equal to the previous period's interest rate multiplied by -1. If there is no switch then the previous interest rate prevails.

2. Plot the simulated series.

3. The theoretical mean and variance for this Markov-chain are 0 and 1 respectively. How do these compare with the mean and variance obtained from simulations?

5. Log-linearisation of baseline model

This exercise is about log-linearisation of the baseline DSGE model presented in the classes. To make the exercise more concrete, we introduce calibrated values for the parameters of the model. These numbers, taken from Ellison and Scott (2000) are chosen so that key-features of the model are matched to the UK economy.

Parameter	Calibrated value	Explanation
β	0.99	Discount rate
σ	1	Intertemporal elasticity of substitution in consumption
χ	1.55	Multiplier on hours worked in disutility of work in utility function
η	0	1+ η is exponent on hours worked in disutility of work in utility function
θ	2.064	Elasticity of demand for firm *i*'s product
ω	0.5	Percentage of firms unable to change their price each period
α	3	(1/α) is elasticity of wages w.r.t output gap
δ	1.5	Coefficient on inflation in simple rule for interest rate

1. Calculate the steady-state value of output, $\bar{Y}$, in the economy. Can you explain why you get quite a nice round answer? Using MATLAB, investigate how $\bar{Y}$ varies with the calibrated parameters θ, χ, η and σ. What is the economic intuition for each of these relationships?

2. Prepare a MATLAB program with the state-space form matrices A_0, A_1, B_0 as a function of the deep calibrated parameters. We will use these matrices in the next set of exercises when we solve the baseline DSGE model.

6. Log-linearisation of an RBC model

The baseline DSGE model we have discussed so far emphasises the role that nominal shocks play in the determination of output. Another class of models known as Real Business Cycle (RBC) models, focuses on technology shocks as the dominant source of output fluctuations. In this exercise, we derive the log-linearised form of a simple RBC model. The exercise involves considerable work with pencil and paper rather than MATLAB. In a simple RBC model, the consumer maximises the discounted value of current and expected future utility from consumption, subject to a budget constraint and the (exogenous) law of motion for technology. The full problem is given below.

$$
\begin{aligned}
&\max_{\{C_t\}} E\sum_{t=0}^{\infty}\beta^t \frac{C_t^{1-\sigma}}{1-\sigma} \\
&s.t. \\
&C_t + K_t = A_t K_{t-1}^{\alpha} + (1-\delta)K_{t-1} \\
&\ln A_{t+1} = \rho \ln A_t + \varepsilon_t
\end{aligned}
$$

The utility function is of the standard CRRA form. The left hand side of the budget constraint shows expenditure, which is divided between consumption C_t and capital K_t carried forward to the next period. The right hand side of the budget constraint is correspondingly income, which derives from the productivity and depreciated value of existing capital, $A_t K_{t-1}^{\alpha}$ and $(1-\delta)K_{t-1}$. δ is the discount rate. Technology is assumed to follow and AR(1) process with persistence parameter ρ and i.i.d. shocks ε_t.

1. Show that the Euler equation for consumption can be written in the following form.

   $$C_t^{-\sigma} = E_t\beta\left[C_{t+1}^{-\sigma}(\alpha A_{t+1}K_t^{\alpha-1} + 1 - \delta)\right]$$

   Interpret the Euler equation.

2. Find the steady state values $\overline{A}, \overline{C}, \overline{K}$ of technology, consumption and capital in the model by taking steady-state versions of the Euler equation, budget constraint and low of motion for technology. How do $\overline{A}, \overline{C}, \overline{K}$ vary with δ, β, α. Why?

3. Log-linearise the first order conditions, i.e. Euler equation, budget constraint and law of motion for technology, to obtain a 3-dimensional system in $\overline{A}_t, \overline{C}_t, \overline{K}_t$ and the shock ε_t. You will need the general formula to log-linearise the budget constraint.

4. Put the model in state-space form

7. The CRRA utility function when σ = 1

It is well known that the constant relative risk aversion utility function tends in the limit to the logarithmic function when σ = 1. In other words,

$$\lim_{\sigma \to 1} \frac{C_t^{1-\sigma}}{1-\sigma} = \ln C_t$$

Prove that this is correct. To do this, you will need to apply l'Hôpital's rule, which is repeated below.

If $\lim_{\sigma \to 1} f(\sigma) = 0$ and $\lim_{\sigma \to 1} g(\sigma) = 0$ then $\lim_{\sigma \to 1} \frac{f(\sigma)}{g(\sigma)} = \lim_{\sigma \to 1} \frac{f'(\sigma)}{g'(\sigma)}$

There are a couple of tricks to the proof!

8. B-K conditions in baseline model

This exercise is designed to check whether the Blanchard-Kahn conditions hold in the baseline model with AR(1) policy shocks. The state-space form of the model is given by

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \sigma^{-1} \\ 0 & 0 & \beta \end{pmatrix} \begin{pmatrix} v_{t+1} \\ E_t \hat{x}_{t+1} \\ E_t \hat{\pi}_{t+1} \end{pmatrix} = \begin{pmatrix} \rho & 0 & 0 \\ \sigma^{-1} & 1 & \sigma^{-1}\delta \\ 0 & -\kappa & 1 \end{pmatrix} \begin{pmatrix} v_t \\ \hat{x}_t \\ \hat{\pi}_t \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \varepsilon_{t+1}$$

We use the same calibration as in exercise 6 and ρ = 0.5. The following M-file sets up the standard state space form $A_0 E_t X_{t+1} = A_1 X_t + B_0 \varepsilon_{t+1}$ and calculates the alternative state space form $E_t X_{t+1} = AX_t + B\varepsilon_{t+1}$. The code is available for download from

http://www2.warwick.ac.uk/fac/soc/economics/staff/faculty/ellison/boe/q8.m

```
% Calibrated parameter values

clear;
beta=0.99;
sigma=1;
chi=1.55;
eta=0;
theta=2.064;
omega=0.5;
alpha=3;
delta=1.5;
rho=0.5;

% Calculate kappa

kappa=(1-omega)*(1-beta*omega)/(alpha*omega);

% Define state space matrices

A0=zeros(3,3);
A0(1,1)=1;
A0(2,2)=1;
A0(2,3)=sigma^-1;
A0(3,3)=beta;

A1=zeros(3,3);
A1(1,1)=rho;
A1(2,1)=sigma^-1;
A1(2,2)=1;
A1(2,3)=sigma^-1*delta;
A1(3,2)=-kappa;
A1(3,3)=1;
```

```
B0=zeros(3,1);
B0(1,1)=1;

% Calculate alternative state space matrices

A=inv(A0)*A1;
B=inv(A0)*B0;
```

1. Use the command **[p,lambda] = eig(A)** to perform the Jordan decomposition of the A matrix. The p* matrix can be extracted using the command **pstar = inv(p)**. Are the Blanchard-Kahn conditions satisfied for this calibration of the model? If the eigenvalues are complex use **abs(lambda)** to obtain their magnitude.

2. Partition the lambda, pstar and R=Pstar*B matrices. Before doing this, ensure that the stable eigenvectors (with absolute value less than one) are in the top-left corner of the lambda matrix and the unstable eigenvectors (with absolute value greater than one) are in the bottom-right partition of the lambda matrix. If the eigenvalues are in the wrong places, use the following code to sort them into ascending order.

   ```
   % Sort eigenvalues and eigenvectors in ascending order

   val=diag(lambda);
   t=sortrows([val p’],1);
   lambda=diag(t(:,1));
   p=t(:,2:4)’;
   pstar=inv(p);
   ```

   What does each row of this code do?

3. Derive the law of motion for forward-looking variables as a function of backward-looking variables in the model. Your final law of motion may include terms with extremely small imaginary components, e.g. $-1.2+1.3\times10^{-16}i$. If this happens do not worry, it is due to the computer being unable to distinguish between very small numbers and zero. Use the command **real(x)** to ignore the imaginary part.

4. Derive the law of motion for future backward-looking variables as a function of current backward-looking variables.

9. B-K conditions and δ

This exercise investigates the relationship between the Blanchard-Kahn conditions and the value for δ, the coefficient on inflation in the simple monetary policy rule. We use the same state-space form as in the previous question, so begin with the following M-file. The code can be downloaded from

http://www2.warwick.ac.uk/fac/soc/economics/staff/faculty/ellison/boe/q9.m

```
% Calibrated parameter values

clear;
beta=0.99;
sigma=1;
chi=1.55;
eta=0;
theta=2.064;
omega=0.5;
alpha=3;
delta=1.5;
rho=0.5;

% Calculate kappa

kappa=(1-omega)*(1-beta*omega)/(alpha*omega);

% Define state space matrices

A0=zeros(3,3);
A0(1,1)=1;
A0(2,2)=1;
A0(2,3)=sigma^-1;
A0(3,3)=beta;

A1=zeros(3,3);
A1(1,1)=rho;
A1(2,1)=sigma^-1;
A1(2,2)=1;
A1(2,3)=sigma^-1*delta;
A1(3,2)=-kappa;
A1(3,3)=1;

B0=zeros(3,1);
B0(1,1)=1;

% Calculate alternative state space matrices

A=inv(A0)*A1;
B=inv(A0)*B0;
```

```
% Jordan decomposition of A

[p,lambda] = eig(A);
pstar = inv(p);

% Sort eigenvalues and eigenvectors in ascending order

val=diag(lambda);
t=sortrows([val p'],1);
lambda=diag(t(:,1));
p=t(:,2:4)';
pstar=inv(p);
```

1. What restrictions are required on the parameter δ to ensure that the Blanchard-Kahn conditions are met? To answer this question, write a **for … end** loop that checks whether the Blanchard-Kahn conditions are satisfied for different values of δ from 0 to 2 in steps of 0.1. If the eigenvalues are complex use **abs(lambda)** to obtain their magnitude. For each value of δ check that there are 1 stable and 2 unstable roots. What is the economic intuition for your result?

2. For values of δ which satisfy the Blanchard-Kahn conditions, investigate how the solved-out equilibrium laws of motion depend on δ. Use the command **real(x)** to ignore any extremely small imaginary parts.

10. B-K conditions in RBC model

In exercise 6 you were asked to derive the state-space form of a real business cycle model. You should have obtained an algebraic equation of the form

$$\begin{pmatrix} 1-(1-\delta)\beta & (\alpha-1)(1-(1-\delta)\beta) & -\sigma \\ 0 & \bar{k} & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{A}_{t+1} \\ \hat{k}_t \\ E_t \hat{c}_{t+1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & -\sigma \\ \bar{y} & \alpha\bar{y}+(1-\delta)\bar{k} & -\bar{c} \\ \rho & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{A}_t \\ \hat{k}_{t-1} \\ \hat{c}_t \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \varepsilon_t$$

The following M-file sets up the state-space form and solves for the Jordan decomposition for a simple calibration of the model. It is available for download at

http://www2.warwick.ac.uk/fac/soc/economics/staff/faculty/ellison/boe/q10.m

```
% Calibrated parameter values

clear;
beta=0.9;
alpha=0.75;
sigma=1;
delta=0.3;
rho=0.95;

% Calculate steady-state values

kbar=((1-(1-delta)*beta)/(alpha*beta))^(1/(alpha-1));
cbar=kbar^alpha-delta*kbar;
ybar=kbar^alpha;

% Define state space matrices

A0=zeros(3,3);
A0(1,1)=1-(1-delta)*beta;
A0(1,2)=(1-(1-delta)*beta)*(alpha-1);
A0(1,3)=-sigma;
A0(2,2)=kbar;
A0(3,1)=1;

A1=zeros(3,3);
A1(1,3)=-sigma;
A1(2,1)=ybar;
A1(2,2)=alpha*ybar+(1-delta)*kbar;
A1(2,3)=-cbar;
A1(3,1)=rho;

B0=zeros(3,1);
B0(3,1)=1;
```

```
% Calculate alternative state space matrices

A=inv(A0)*A1;
B=inv(A0)*B0;

% Jordan decomposition of A

[p,lambda] = eig(A);
pstar = inv(p);

% Sort eigenvalues and eigenvectors in ascending order

val=diag(lambda);
t=sortrows([val p’],1);
lambda=diag(t(:,1));
p=t(:,2:4)’;
pstar=inv(p);
```

1. Are the Blanchard-Kahn conditions satisfied for this calibration of the model? If the eigenvalues are complex use **abs(lambda)** to obtain their magnitude.

2. Can you find alternative calibrations for which the Blanchard-Kahn conditions do not hold? Use the command **real(x)** to ignore any extremely small imaginary parts.

11. Stylised facts

This exercise asks you to calculate some stylised facts for the baseline model with persistent interest rate shocks. As a reminder, the state space form and recursive solution of the model are defined by

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \sigma^{-1} \\ 0 & 0 & \beta \end{pmatrix} \begin{pmatrix} v_{t+1} \\ E_t \hat{x}_{t+1} \\ E_t \hat{\pi}_{t+1} \end{pmatrix} = \begin{pmatrix} \rho & 0 & 0 \\ \sigma^{-1} & 1 & \sigma^{-1}\delta \\ 0 & -\kappa & 1 \end{pmatrix} \begin{pmatrix} v_t \\ \hat{x}_t \\ \hat{\pi}_t \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \varepsilon_{t+1}$$

$$y_t = -P_{22}^{*-1} P_{21}^{*} w_t$$
$$w_{t+1} = (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} \Lambda_1 (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*}) w_t + (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} R_1 \varepsilon_{t+1}$$

The MATLAB M-file below calculates the state space form, performs the Jordan decomposition, and prints out the matrices of the recursive solution to the model. It can be downloaded from

http://www2.warwick.ac.uk/fac/soc/economics/staff/faculty/ellison/boe/q11.m

```
% Calibrated parameter values

clear;
beta=0.99;
sigma=1;
chi=1.55;
eta=0;
theta=2.064;
omega=0.5;
alpha=3;
delta=1.5;
rho=0.5;

% Calculate kappa

kappa=(1-omega)*(1-beta*omega)/(alpha*omega);

% Define state space matrices

A0=zeros(3,3);
A0(1,1)=1;
A0(2,2)=1;
A0(2,3)=sigma^-1;
A0(3,3)=beta;
```

```
A1=zeros(3,3);
A1(1,1)=rho;
A1(2,1)=sigma^-1;
A1(2,2)=1;
A1(2,3)=sigma^-1*delta;
A1(3,2)=-kappa;
A1(3,3)=1;

B0=zeros(3,1);
B0(1,1)=1;

% Calculate alternative state space matrices

A=inv(A0)*A1;
B=inv(A0)*B0;

% Jordan decomposition of A

[p,lambda]=eig(A);
pstar=inv(p);

% Sort eigenvalues and eigenvectors in ascending order

val=diag(lambda);
t=sortrows([val p’],1);
lambda=diag(t(:,1));
p=t(:,2:4)’;
pstar=inv(p);

% Partition matrices

LAMBDA1=lambda(1,1);
LAMBDA2=lambda(2:3,2:3);

P11=pstar(1,1);
P12=pstar(1,2:3);
P21=pstar(2:3,1);
P22=pstar(2:3,2:3);

R=pstar*B;

% Print out matrices of recursive solution of model

real(inv(P11-P12*inv(P22)*P21)*LAMBDA1*(P11-P12*inv(P22)*P21));
real(inv(P11-P12*inv(P22)*P21)*R(1));
real(-inv(P22)*P21);
```

1. Using the above code as a starting point, simulate the model for *n* periods, where *n* is a large number such as 1000. To do this, begin by drawing random numbers (of unit variance) from a normal distribution for the interest rate innovations $\{\varepsilon_t\}$ using the command **et = randn(n,1)**. Next, simulate the backward-looking variables $\{w_t\}$ recursively using a **for … end** loop. Finally, calculate the forward-looking variables $\{y_t\}$ directly from the backward-looking variables $\{w_t\}$. In this model, w_t is the persistent shock v_t to the interest rate and y_t is a 2 × 1 vector of the output gap and inflation.

2. Calculate the interest rate each period by applying the simple interest rate rule $\hat{i}_t = \delta\hat{\pi}_t + v_t$.

3. Calculate the standard deviations of the interest rate, output gap and inflation using the command **std(x)**. To ameliorate the effect of starting values, only use simulated values from **100:(n-1)**. Do your calculated standard deviations agree with those given in the lecture? If not, why not?

4. Calculate the correlations between the interest rate, output gap and inflation using the command **corrcoef(X)**. This command calculates the correlation between the columns of X so define X as a 900 × 3 vector, in which the columns are the simulated interest rate, output gap and inflation respectively.

5. Calculate the autocorrelation of the output gap. You can use the same command **corrcoef(X)** as before. This time, fill the first column of X with the simulated output gap observations from **100:(n-1)** and the second column with simulated output gap observations from **99:(n-2)**. By doing this, you are calculating the correlation coefficient between the output gap at time *t* (the first column) and the output at time *t-1* (the second column). Calculating further autocorrelations is a simple generalization of this procedure.

6. Calculate the cross correlation between the output gap and interest rates at lag 1 and lead 1. The simplest approach to doing this is to follow the procedure in part (5), although this time fill the second column of X with interest rate observations from **99:(n-2)** for the correlation between the output gap at time *t* and interest rates at time *t-1*, and interest rate observations for **101:n** for the correlation between the output gap at time *t* and interest rates at time *t+1*.

12. Impulse response functions

This exercise is about impulse response functions, in our case the response of interest rates, the output gap and inflation to innovations ε_{t+1} in the interest rate shock v_t. The steps necessary to calculate the impulse responses are very similar to parts (1) and (2) of the previous exercise performed when simulating the model. In fact, if you want it is possible to recycle much of the code from the previous exercise when answering this question. The state space form and recursive solution of the model remain the same:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \sigma^{-1} \\ 0 & 0 & \beta \end{pmatrix}\begin{pmatrix} v_{t+1} \\ E_t\hat{x}_{t+1} \\ E_t\hat{\pi}_{t+1} \end{pmatrix} = \begin{pmatrix} \rho & 0 & 0 \\ \sigma^{-1} & 1 & \sigma^{-1}\delta \\ 0 & -\kappa & 1 \end{pmatrix}\begin{pmatrix} v_t \\ \hat{x}_t \\ \hat{\pi}_t \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\varepsilon_{t+1}$$

$$y_t = -P_{22}^{*-1}P_{21}^{*}w_t$$
$$w_{t+1} = (P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})^{-1}\Lambda_1(P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})w_t + (P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})^{-1}R_1\varepsilon_{t+1}$$

As in the previous exercise, the starting point is a MATLAB M-file to calculate the state space form, perform the Jordan decomposition, and print out the matrices of the recursive solution to the model. The code below is identical to that in the previous exercise and can be downloaded from

http://www2.warwick.ac.uk/fac/soc/economics/staff/faculty/ellison/boe/q12.m

```
% Calibrated parameter values

clear;
beta=0.99;
sigma=1;
chi=1.55;
eta=0;
theta=2.064;
omega=0.5;
alpha=3;
delta=1.5;
rho=0.5;

% Calculate kappa

kappa=(1-omega)*(1-beta*omega)/(alpha*omega);
```

```
% Define state space matrices

A0=zeros(3,3);
A0(1,1)=1;
A0(2,2)=1;
A0(2,3)=sigma^-1;
A0(3,3)=beta;

A1=zeros(3,3);
A1(1,1)=rho;
A1(2,1)=sigma^-1;
A1(2,2)=1;
A1(2,3)=sigma^-1*delta;
A1(3,2)=-kappa;
A1(3,3)=1;

B0=zeros(3,1);
B0(1,1)=1;

% Calculate alternative state space matrices

A=inv(A0)*A1;
B=inv(A0)*B0;

% Jordan decomposition of A

[p,lambda]=eig(A);
pstar=inv(p);

% Sort eigenvalues and eigenvectors in ascending order

val=diag(lambda);
t=sortrows([val p’],1);
lambda=diag(t(:,1));
p=t(:,2:4)’;
pstar=inv(p);

% Partition matrices

LAMBDA1=lambda(1,1);
LAMBDA2=lambda(2:3,2:3);

P11=pstar(1,1);
P12=pstar(1,2:3);
P21=pstar(2:3,1);
P22=pstar(2:3,2:3);

R=pstar*B;
```

```
% Print out matrices of recursive solution of model

real(inv(P11-P12*inv(P22)*P21)*LAMBDA1*(P11-P12*inv(P22)*P21));
real(inv(P11-P12*inv(P22)*P21)*R(1));
real(-inv(P22)*P21);
```

1. Calculate the response up to horizon h of the output gap and inflation to a unit ε_t shock. To do this efficiently, define the shocks $\{\varepsilon_t\}$ to be zero in all periods except the first by applying the pair of commands **et = zeros(h,1)** and **et(1) = 1**. The response of the backward-looking variables $\{w_t\}$ can be calculated recursively using a **for … end** loop. The forward-looking variables $\{y_t\}$ are calculated as before as a function of the backward-looking variables. Graph the impulse response functions of the output gap and inflation using the **plot(x,y)** command. Are the impulse response functions the same as those in the lecture? They should be!

2. Calculate the impulse response function for the interest rate by applying the simple interest rate rule $\hat{i}_t = \delta\hat{\pi}_t + v_t$. Add this to your graph.

3. What is the impulse response function for the real interest rate? Use the *ex post* real interest rate as defined by $\hat{i}_t - \hat{\pi}_t$.

4. Investigate the consequences for the impulse response function of changing the degree of persistence ρ in the interest rate shock. What happens if ρ = 0? Do you appreciate now why we added this persistent shock to make the model more interesting?

13. FEVD

This question is about calculating the forecast error variance decomposition in the simplest possible model. Unfortunately, the model is not very simple since we need to introduce an additional shock. FEVD only makes sense if we have more than one shock - in the one-shock model we have looked at so far, the interest rate shock is responsible for 100% of the variance in each variable at any horizon (because it is the only shock!).

To allow for a second shock in the system, we introduce persistent cost-push shocks in the Phillips curve. The model is therefore given by the following structural equations:

$$\hat{x}_t = E_t \hat{x}_{t+1} - \sigma^{-1}(\hat{i}_t - E_t \hat{\pi}_{t+1})$$
$$\hat{\pi}_t = \beta E_t \hat{\pi}_{t+1} + \kappa \hat{x}_t + u_t$$
$$\hat{i}_t = \delta \hat{\pi}_t + v_t$$

$$\kappa = \frac{(1-\omega)(1-\beta\omega)}{\alpha\omega}$$

The two shocks v_t and u_t are AR(1) processes with shocks ε^v_{t+1} and ε^u_{t+1}. The shocks themselves have variances σ^2_v and σ^2_u.

$$v_{t+1} = \rho_v v_t + \varepsilon^v_{t+1}$$
$$u_{t+1} = \rho_u u_t + \varepsilon^u_{t+1}$$

The state-space form of the model is as follows:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \sigma^{-1} \\ 0 & 0 & 0 & \beta \end{pmatrix} \begin{pmatrix} v_{t+1} \\ u_{t+1} \\ E_t \hat{x}_{t+1} \\ E_t \hat{\pi}_{t+1} \end{pmatrix} = \begin{pmatrix} \rho_v & 0 & 0 & 0 \\ 0 & \rho_u & 0 & 0 \\ \sigma^{-1} & 0 & 1 & \sigma^{-1}\delta \\ 0 & -1 & -\kappa & 1 \end{pmatrix} \begin{pmatrix} v_t \\ u_t \\ \hat{x}_t \\ \hat{\pi}_t \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \varepsilon^r_{t+1} \\ \varepsilon^u_{t+1} \end{pmatrix}$$

We calibrate the model with four additional parameters. The calibration of the other parameters is the same as before.

Parameter	Calibrated value	Explanation
ρ_v	0.5	Persistence of interest rate shocks
ρ_u	0.8	Persistence of cost-push shocks
σ_v	1	Standard deviation of interest rate shocks
σ_u	0.5	Standard deviation of cost-push shocks

The following M-file sets up the state-space form, solves for the Jordan decomposition of the model, and calculates the impulse response functions. The impulse response functions are stored in two variables, irf_v and irf_u. irf_v is a 4 × 25 matrix which shows the response of $(v_t \quad u_t \quad \hat{x}_t \quad \hat{\pi}_t)'$ to a unit ε_t^v shock over the horizons 1 to 25. Each column of irf_v corresponds to a different horizon. irf_u is a similar 4 × 25 matrix for the response to a unit ε_t^u shock. The M-file is available for download at

http://www2.warwick.ac.uk/fac/soc/economics/staff/faculty/ellison/boe/q13.m

```
% Calibrated parameter values

clear;
beta=0.99;
sigma=1;
chi=1.55;
eta=0;
theta=2.064;
omega=0.5;
alpha=3;
delta=1.5;
rhov=0.5;
rhou=0.8;
sigmav=1;
sigmau=0.5;

% Calculate kappa

kappa=(1-omega)*(1-beta*omega)/(alpha*omega);
```

```
% Define state space matrices

A0=zeros(4,4);
A0(1,1)=1;
A0(2,2)=1;
A0(3,3)=1;
A0(3,4)=sigma^-1;
A0(4,4)=beta;

A1=zeros(4,4);
A1(1,1)=rhov;
A1(2,2)=rhou;
A1(3,1)=sigma^-1;
A1(3,3)=1;
A1(3,4)=sigma^-1*delta;
A1(4,2)=-1;
A1(4,3)=-kappa;
A1(4,4)=1;

B0=zeros(4,2);
B0(1,1)=1;
B0(2,2)=1;

% Calculate alternative state space matrices

A=inv(A0)*A1;
B=inv(A0)*B0;

% Jordan decomposition of A

[p,lambda]=eig(A);
pstar=inv(p);

% Sort eigenvalues and eigenvectors in ascending order

val=diag(lambda);
t=sortrows([val p’],1);
lambda=diag(t(:,1));
p=t(:,2:5)’;
pstar=inv(p);
```

```
% Partition matrices

LAMBDA1=lambda(1:2,1:2);
LAMBDA2=lambda(3:4,3:4);

P11=pstar(1:2,1:2);
P12=pstar(1:2,3:4);
P21=pstar(3:4,1:2);
P22=pstar(3:4,3:4);

R=pstar*B;

% Calculate impulse response functions

h=20;
irf_v=zeros(4,h);
irf_u=zeros(4,h);
irf_v(1:2,1)=inv(P11-P12*inv(P22)*P21)*R(1:2,:)*[1;0];
irf_u(1:2,1)=inv(P11-P12*inv(P22)*P21)*R(1:2,:)*[0;1]);

i=1;
for i=1:(h-1);
      irf_v(1:2,i+1)=inv(P11-P12*inv(P22)*P21)*LAMBDA1*(P11-
                     P12*inv(P22)*P21)*irf_v(1:2,i);
      irf_u(1:2,i+1)=inv(P11-P12*inv(P22)*P21)*LAMBDA1*(P11-
                     P12*inv(P22)*P21)*irf_u(1:2,i);
end;

irf_u(3:4,:)=-inv(P22)*P21*irf_u(1:2,:);
irf_v(3:4,:)=-inv(P22)*P21*irf_v(1:2,:);

irf_v=real(irf_v);
irf_u=real(irf_u);
```

1. Verify your understanding of the M-file by plotting the response of the output gap and inflation to the two different types of shocks. Note that the shock in each case is normalised to 1, not to one standard deviation. It is probably more informative to use the **subplot** option to generate a different graph for each set of impulse response functions.

2. Calculate the impulse response function of the interest rate to each shock using the formula $\hat{i}_t = \delta\hat{\pi}_t + v_t$. Add these to the graphs.

3. Calculate the forecast error variance decomposition of output at horizons up to h = 10. As the FEVD is a simple transform of the impulse response functions, it is relatively straightforward to use a **for … end** loop to apply the relevant formula:

$$\frac{\sum_{i=1}^{h}(\Psi_i^1)^2\sigma_{v^1}^2}{\sum_{i=1}^{h}(\Psi_i^1)^2\sigma_{v^1}^2+\sum_{i=1}^{h}(\Psi_i^2)^2\sigma_{v^2}^2}$$

   For each value of h from 1 to 10 in the **for … end** loop, sum the squares of the relevant elements of irf_v and irf_u. The FEVD should be calculated from these two sums, weighted by the corresponding variances of the shocks. Graph your results.

4. Calculate the FEVD for inflation and interest rates and compare them to the results obtained for output.

5. Investigate how the results change as the relative persistence and/or the relative variance of the two shocks changes.

14. Multi-shock DSGE model

In this exercise you are required to analyse a version of the baseline DSGE model in which there are three shocks: interest rate shocks, cost-push shocks and aggregate demand (IS) shocks. In the analysis you should be able to recycle a lot of the code used before, most notably from exercises 11 to 13. The basic structure of the model is the same as before, consisting of a dynamic IS curve, a Phillips curve, and a simple rule for monetary policy.

$$\hat{x}_t = E_t\hat{x}_{t+1} - \sigma^{-1}(\hat{i}_t - E_t\hat{\pi}_{t+1}) + g_t$$
$$\hat{\pi}_t = \beta E_t\hat{\pi}_{t+1} + \kappa\hat{x}_t + u_t$$
$$\hat{i}_t = \delta\hat{\pi}_t + v_t$$

$$\kappa = \frac{(1-\omega)(1-\beta\omega)}{\alpha\omega}$$

The three shocks are assumed to be independent AR(1) processes given by

$$\begin{pmatrix} v_{t+1} \\ u_{t+1} \\ g_{t+1} \end{pmatrix} = \begin{pmatrix} \rho_v & 0 & 0 \\ 0 & \rho_u & 0 \\ 0 & 0 & \rho_g \end{pmatrix}\begin{pmatrix} v_t \\ u_t \\ g_t \end{pmatrix} + \begin{pmatrix} \upsilon_v & 0 & 0 \\ 0 & \upsilon_u & 0 \\ 0 & 0 & \upsilon_g \end{pmatrix}\begin{pmatrix} \varepsilon^v_{t+1} \\ \varepsilon^u_{t+1} \\ \varepsilon^g_{t+1} \end{pmatrix}$$

The model is calibrated as before, with the only additional parameters to calibrate being those relating to the shock processes. For completeness, the full calibration is reported below:

Parameter	Calibrated value	Explanation
β	0.99	Discount rate
σ	1	Intertemporal elasticity of substitution in consumption
ω	0.5	Percentage of firms unable to change their price each period
α	3	(1/α) is elasticity of wages w.r.t output gap
δ	1.5	Coefficient on inflation in simple rule for interest rate
ρ_v	0.5	Persistence of interest rate shocks
ρ_u	0.8	Persistence of cost-push shocks
ρ_g	0.3	Persistence of AD shocks
ν_v	1	Standard deviation of interest rate shocks
ν_u	0.5	Standard deviation of cost-push shocks
ν_u	1	Standard deviation of AD shocks

1. Substitute out for the interest rate in the IS curve and write down the state-space form of the model $A_0 E_t X_{t+1} = A_1 X_t + B_0 \varepsilon_{t+1}$. You should find that X_t is a 5×1 vector $\begin{pmatrix} v_t & u_t & g_t & \hat{x}_t & \hat{\pi}_t \end{pmatrix}'$ and ε_t is a 3×1 vector $\begin{pmatrix} \varepsilon_{t+1}^v & \varepsilon_{t+1}^u & \varepsilon_{t+1}^g \end{pmatrix}'$. The parameter matrices A_0, A_1 and B_0 are of dimension 5×5, 5×5 and 5×3 respectively.

2. Invert the state space form into the alternative specification $E_t X_{t+1} = A X_t + B \varepsilon_{t+1}$ and perform the Jordan decomposition of the 5×5 matrix A. Check that the Blanchard-Kahn conditions are satisfied. As there are three backward-looking variables $\begin{pmatrix} v_t & u_t & g_t \end{pmatrix}$ and two forward-looking variables $\begin{pmatrix} \hat{x}_t & \hat{\pi}_t \end{pmatrix}$ in the model, you should have three stable eigenvalues of magnitude less than 1 and two unstable eigenvalues of magnitude greater than 1.

3. Partition the X_t vector into backward-looking variables $\tilde{w}_t = (v_t \quad u_t \quad g_t)'$ and forward looking variables $\tilde{y}_t = (\hat{x}_t \quad \hat{\pi}_t)'$. Partition also the matrices Λ and P from the Jordan decomposition. Calculate the solution for the equilibrium behaviour of the economy using the equations

$$y_t = -P_{22}^{*-1} P_{21}^{*} w_t$$
$$w_{t+1} = (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} \Lambda_1 (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*}) w_t + (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} R_1 \varepsilon_{t+1}$$

4. Calculate via simulation a series of stylised facts which summarise the behaviour of this model economy. Pay particular attention to the correlations and cross-correlations between variables, which are no longer ±1 because there are many shocks hitting the economy. Do the signs of the correlations match with your expectations based on the behaviour of the UK economy? If not, why do you think the model and data are different? How (heuristically) would you change the model to better match UK data?

5. Derive the responses of interest rates, the output gap and inflation to the three different shocks. Explain the intuition for the response of each variable to each type of shock.

6. Which of the three shocks contributes most to fluctuations in interest rates, the output gap and inflation? The key to answering this is to perform a forecast error variance decomposition for the three variables. As a reminder, the relevant formula is

$$\frac{\sum_{i=1}^{h} (\Psi_i^1)^2 \sigma_{v^1}^2}{\sum_{i=1}^{h} (\Psi_i^1)^2 \sigma_{v^1}^2 + \sum_{i=1}^{h} (\Psi_i^2)^2 \sigma_{v^2}^2}$$

7. Assume that the welfare of agents in the economy can be measured by the function

$$\min \sum_{i=0}^{\infty} \beta^i (\hat{\pi}_t^2 + \hat{x}_t^2 + 0.1 \hat{i}_t^2)$$

Which type of shock is most detrimental to welfare in this economy?

15. DSGE model with persistence

This exercise is about the effects of introducing ad hoc persistence factors in the baseline model. Doing so is particularly prevalent in policy circles, following the lead of authors such as Del Negro-Schorfeide and Smets-Wouters. The persistence factors introduce extra lagged terms in the dynamic IS equation and the Phillips curve.

$$\hat{x}_t = \frac{h}{1+h}\hat{x}_{t-1} + \frac{1}{1+h}E_t\hat{x}_{t+1} - \sigma^{-1}(\hat{i}_t - E_t\hat{\pi}_{t+1})$$

$$\hat{\pi}_t = \frac{\gamma_p}{1+\beta\gamma_p}\hat{\pi}_{t-1} + \frac{\beta}{1+\beta\gamma_p}E_t\hat{\pi}_{t+1} + \kappa\hat{x}_t$$

$$\hat{i}_t = \delta\hat{\pi}_t + v_t$$

$$\kappa = \frac{(1-\omega)(1-\beta\omega)}{\alpha\omega}$$

Since we are attempting to explain the persistence of the output gap and inflation by the lagged terms, it no longer makes sense to have persistent interest rate shocks. We therefore proceed with interest rate shocks being white noise.

$$v_{t+1} = \varepsilon_{t+1}$$

The calibration of the economy is standard, except for the parameters on lagged endogenous variables in the IS and Phillips curves. As a baseline, we take the numbers estimated by Smets-Wouters for the euroarea economy. The full calibration is

Parameter	Calibrated value	Explanation
β	0.99	Discount rate
σ	1	Intertemporal elasticity of substitution in consumption
ω	0.5	Percentage of firms unable to change their price each period
α	3	(1/α) is elasticity of wages w.r.t output gap
$\frac{h}{1+h}$	0.35	Coefficient on lagged output gap in IS curve
$\frac{1}{1+h}$	0.65	Coefficient on expected future output gap in IS curve
$\frac{\gamma_p}{1+\beta\gamma_p}$	0.29	Coefficient on lagged inflation in Phillips curve
$\frac{\beta}{1+\beta\gamma_p}$	0.7	Coefficient on expected future inflation in Phillips curve
δ	1.5	Coefficient on inflation in simple rule for interest rate
σ_ε^2	1	Variance of interest rate shocks
ρ_v	0	Persistence of interest rate shocks

1. Substitute out for the interest rate and $v_{t+1} = \varepsilon_{t+1}$ in the IS curve and write down the state-space form of the model $A_0 E_t X_{t+1} = A_1 X_t + B_0 \varepsilon_{t+1}$. You should find that X_t is a 4×1 vector $(\hat{x}_{t-1} \quad \hat{\pi}_{t-1} \quad \hat{x}_t \quad \hat{\pi}_t)'$ and ε_t is a 1×1 vector (ε_{t+1}). Two of the equations in the state-space form simply connect the second and third terms $(\hat{x}_t \quad \hat{\pi}_t)'$ of X_{t+1} to the fourth and fifth terms $(\hat{x}_t \quad \hat{\pi}_t)'$ of X_t, for example the first and second rows of the state-space form take on the following form:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ & & & \\ & & & \end{pmatrix} \begin{bmatrix} \hat{x}_t \\ \hat{\pi}_t \\ E_t \hat{x}_{t+1} \\ E_t \hat{\pi}_{t+1} \end{bmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ & & & \\ & & & \end{pmatrix} \begin{bmatrix} \hat{x}_{t-1} \\ \hat{\pi}_{t-1} \\ \hat{x}_t \\ \hat{\pi}_t \end{bmatrix} + \begin{pmatrix} 0 \\ 0 \\ \\ \end{pmatrix} \varepsilon_{t+1}$$

The parameter matrices A_0, A_1 and B_0 are of dimension 4×4, 4×4 and 4×1 respectively.

2. Invert the state space form into the alternative state space form $E_t X_{t+1} = AX_t + B\varepsilon_{t+1}$ and perform the Jordan decomposition of the 4×4 matrix A. Check that the Blanchard-Kahn conditions are satisfied. As there are two backward-looking variables $(\hat{x}_{t-1} \quad \hat{\pi}_{t-1})$ and two forward-looking variables $(\hat{x}_t \quad \hat{\pi}_t)$ in the model, you should have two stable eigenvalues of magnitude less than 1 and two unstable eigenvalues of magnitude greater than 1.

3. Partition the X_t vector into backward-looking variables $\tilde{w}_t = (\hat{x}_{t-1} \quad \hat{\pi}_{t-1})'$ and forward looking variables $\tilde{y}_t = (\hat{x}_t \quad \hat{\pi}_t)'$. Partition also the matrices Λ and P from the Jordan decomposition. Calculate the solution for the equilibrium behaviour of the economy using the equations

$$y_t = -P_{22}^{*-1} P_{21}^{*} w_t$$
$$w_{t+1} = (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} \Lambda_1 (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*}) w_t + (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} R_1 \varepsilon_{t+1}$$

4. Calculate via simulation a series of stylised facts which summarise the behaviour of this model economy. Pay particular attention to the autocorrelations of the output gap and inflation. How do these autocorrelations change as the degree of calibrated persistence in the IS and Phillips varies from 0 to close to 1?

5. Derive the responses of interest rates, the output gap and inflation to the interest rate shock for different calibrations of the persistence parameters. What degree of persistence is needed to come close to your view of the degree of persistence of inflation and the output gap in the UK economy?

16. Optimised Taylor rule

This final exercise takes you through the steps necessary to optimise the coefficients on a simple Taylor rule. The structure of the economy is given as before by a dynamic IS curve and Phillips curve, with a Taylor rule for monetary policy.

$$\hat{x}_t = E_t \hat{x}_{t+1} - \sigma^{-1}(\hat{i}_t - E_t \hat{\pi}_{t+1}) + g_t$$
$$\hat{\pi}_t = \beta E_t \hat{\pi}_{t+1} + \kappa \hat{x}_t + u_t$$
$$\hat{i}_t = \delta_\pi \hat{\pi}_t + \delta_x \hat{x}_t + v_t$$

$$\kappa = \frac{(1-\omega)(1-\beta\omega)}{\alpha\omega}$$

The interest rate, cost-push and AD shocks are assumed to be an AR(1) processes

$$\begin{pmatrix} v_{t+1} \\ u_{t+1} \\ g_{t+1} \end{pmatrix} = \begin{pmatrix} \rho_v & 0 & 0 \\ 0 & \rho_u & 0 \\ 0 & 0 & \rho_g \end{pmatrix} \begin{pmatrix} v_t \\ u_t \\ g_t \end{pmatrix} + \begin{pmatrix} \upsilon_v & 0 & 0 \\ 0 & \upsilon_u & 0 \\ 0 & 0 & \upsilon_g \end{pmatrix} \begin{pmatrix} \varepsilon^v_{t+1} \\ \varepsilon^u_{t+1} \\ \varepsilon^g_{t+1} \end{pmatrix}$$

The aim of monetary policy is assumed to be minimisation of a weighted average of the variances of inflation, the output gap and interest rates. A suitable objective function is therefore

$$\min \sum_{i=0}^{\infty} \beta^i (\hat{\pi}_t^2 + \lambda_x \hat{x}_t^2 + \lambda_i \hat{i}_t^2)$$

where λ_x and λ_i are suitable weights. A full calibration of the economy is given below.

Parameter	Calibrated value	Explanation
β	0.99	Discount rate
σ	1	Intertemporal elasticity of substitution in consumption
ω	0.5	Percentage of firms unable to change their price each period
α	3	(1/α) is elasticity of wages w.r.t output gap
ρ_v	0.5	Persistence of interest rate shocks
ρ_u	0.5	Persistence of cost-push shocks
ρ_g	0.5	Persistence of AD shocks
υ_v	1	Standard deviation of interest rate shocks
υ_u	1	Standard deviation of cost-push shocks
υ_g	1	Standard deviation of AD shocks
λ_x	0.1	Weight on output gap variance in objective function
λ_i	1	Weight on interest rate variance in objective function

1. Substitute out for the interest rate in the IS curve and write down the state-space form of the model $A_0 E_t X_{t+1} = A_1 X_t + B_0 \varepsilon_{t+1}$ for given Taylor rule parameters. You should find that X_t is a 5×1 vector $\begin{pmatrix} v_t & u_t & g_t & \hat{x}_t & \hat{\pi}_t \end{pmatrix}'$ and ε_t is a 3×1 vector $\begin{pmatrix} \varepsilon^v_{t+1} & \varepsilon^u_{t+1} & \varepsilon^g_{t+1} \end{pmatrix}'$. The parameter matrices A_0, A_1 and B_0 are of dimension 5×5, 5×5 and 5×3 respectively.

2. Set up a grid for the parameters in the Taylor rule. You will need a row vector **indx = 0:0.1:2** to act as an index for the grid points, varying from 0 to 2 in steps of 0.1.

3. Proceed to check the Blanchard-Kahn conditions for each combination of Taylor rule parameters. The easiest way to do this is to loop through the combinations by a pair of nested **for … end** loops.

```
for i = 1:21
        for j = 1:21

        % Your code to check Blanchard-Kahn conditions
        % for δπ = indx(i) and δx = indx(j)

        end
end
```

For each combination of Taylor rule parameters indexed by *i* and *j*, calculate the state space form $A_0 E_t X_{t+1} = A_1 X_t + B_0 \varepsilon_{t+1}$, recycling the code in part 1. Invert the state space form into the alternative state space form $E_t X_{t+1} = A X_t + B \varepsilon_{t+1}$ and perform the Jordan decomposition of the 5×5 matrix *A*. Check that the Blanchard-Kahn conditions are satisfied. As there are three backward-looking variables $\begin{pmatrix} v_t & u_t & g_t \end{pmatrix}$ and two forward-looking variables $\begin{pmatrix} \hat{x}_t & \hat{\pi}_t \end{pmatrix}$ in the model, you should have three stable eigenvalues of magnitude less than 1 and two unstable eigenvalues of magnitude greater than 1. If the Blanchard-Kahn conditions are satisfied for parameters indexed by *i* and *j*, then store the value 1 in the *i*th row and *j*th column of a matrix **bkij**. Otherwise store 0 in **bkij**. Storing 1 or 0 in **bkij** can be achieved easily using the **if … else … end** statements.

4. Graph a surface plot of the matrix **bkij** using the command **surface(indx,indx,bkij).** The combinations of Taylor rule parameters for which the Blanchard-Kahn conditions hold should appear in a different colour than those for which it fails. Can you explain why the region where the Blanchard-Kahn conditions are satisfied is shaped how it is?

5. The mathematical condition for the Blanchard-Kahn conditions to hold under a general Taylor rule is $\kappa(\delta_{\pi} - 1) + (1 - \beta)\delta_x > 0$. Do your numerical results confirm this? Do your results shed any further light on the results obtained in exercise 9?

6. Simulate the economy for all combinations of Taylor rule parameters that satisfy the Blanchard-Kahn conditions. There are two obvious ways to do this. Firstly, you could loop through all combinations *i* and *j* of parameters using a double **for … end** loop again, performing a simulation only for those combinations for which **bkij** is equal to 1. Alternatively, you can combine checking the Blanchard-Kahn conditions and simulating in the same loops - simply include the simulation in the branch of the **if … else … end** that is selected when the Blanchard-Kahn conditions are satisfied.

   To simulate the economy for relevant Taylor rule parameters, you will need to partition the X_t vector into backward-looking variables $\tilde{w}_t = \begin{pmatrix} v_t & u_t & g_t \end{pmatrix}'$ and forward looking variables $\tilde{y}_t = \begin{pmatrix} \hat{x}_t & \hat{\pi}_t \end{pmatrix}'$. Partition also the matrices Λ and P from the Jordan decomposition and calculate the solution for the equilibrium behaviour of the economy using the equations

$$y_t = -P_{22}^{*-1} P_{21}^{*} w_t$$
$$w_{t+1} = (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} \Lambda_1 (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*}) w_t + (P_{11}^{*} - P_{12}^{*} P_{22}^{*-1} P_{21}^{*})^{-1} R_1 \varepsilon_{t+1}$$

   For each combination of Taylor rule parameters, simulate the economy and calculate the variance of simulated inflation, output gap and interest rates. Unconditional expected welfare is then a weighted average $\sigma_{\hat{\pi}}^2 + \lambda_x \sigma_{\hat{x}}^2 + \lambda_i \sigma_{\hat{i}}^2$ of these variances. Store this number in the *i*th row and *j*th column of a matrix **lossij**. If the Blanchard-Kahn conditions are not satisfied for the row *i* and column *j* combination then store a very high value such as +9999 in **lossij**.

7. Calculate the optimal combination of Taylor rule parameters. The best combination is that which minimises the loss in **lossij**. To find out the location of the minimum, use the MATLAB command **[C, I] = max(-lossij)** to find the indices of the minimum. The optimal Taylor rule parameters can then be read off from the elements of **I** using the **indx** vector. Do the parameters you obtain look familiar?

8. This approach to optimisation is rather slow because it relies heavily on brute force to find the optimal coefficients. A lot of time is wasted particularly in simulating the economy for each combination of Taylor rule parameters. One way to speed up the process is to recognise that, for each parameter combination, the equilibrium law of motion for the economy is linear:

$$y_t = -P_{22}^{*-1}P_{21}^{*}w_t$$
$$w_{t+1} = (P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})^{-1}\Lambda_1(P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})w_t + (P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})^{-1}R_1\varepsilon_{t+1}$$

Due to linearity, we can derive the variances of y_t and w_t directly, without simulation.

$$\Omega_y = (P_{22}^{*-1}P_{21}^{*})\Omega_w(P_{22}^{*-1}P_{21}^{*})'$$
$$\Omega_w = (I - KK')^{-1}(P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})^{-1}R_1\Omega_\varepsilon[(P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})^{-1}R_1]'$$
$$K = (P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})^{-1}\Lambda_1(P_{11}^{*} - P_{12}^{*}P_{22}^{*-1}P_{21}^{*})$$

Use these analytical formulae to replace the simulations in your code. Do you get the same results? How much faster is the search for the optimal Taylor rule coefficients now?

9. Investigate how the optimised Taylor rule coefficients change with the parameters of the model.