Solving DSGE Models I: Local Methods

— Week 1 —

Vivaldo Mendes

Dep. of Economics — Instituto Universitário de Lisboa

18 September 2017

Summary

I – Introducing the problem

The simplest possible model

1. Consider a model with a forward looking variable (y_t), a predetermined variable (x_t) and an exogenous shocks (ε_t)

$$y_t = \alpha + \beta E_t y_{t+1} + x_t$$

$$x_t = \phi + \lambda x_{t-1} + v_t$$

$$v_t \tilde{} iid(0, \sigma^2)$$

2. This simple model is very easy to solve if we impose two conditions on the solution:
   1. If the **law of iterated expectation** holds
   2. If "certainty equivalence" holds: the optimal (or true) values of y_t and x_t are the same as if we knew y_t and x_t for certain (if there were no uncertainty).

Solution by iterating forward: forward looking variable

1. The solution to the forward looking variable at the nth iteration will be

$$y_t = (\beta)^n E_t y_{t+n} + \sum_{i=0}^{n-1} (\beta)^i \alpha + \sum_{i=0}^{n-1} (\beta)^i E_t x_{t+i}$$

2. Avoiding explosive behavior of y_t when $n \longrightarrow \infty$, we have to impose

$$|\beta| < 1, \quad \text{or} \quad |1/\beta| > 1$$

3. Then we obtain

$$y_t = \sum_{i=0}^{n-1} (\beta)^i \alpha + \sum_{i=0}^{n-1} (\beta)^i E_t x_{t+i} \tag{1}$$

4. Therefore the optimal/true value of y_t depends only upon the value of a constant and the the expected values of the predetermined variable

Solution by iterating forward: predetermined variable

1. The solution to the backward looking variable at the nth forward iteration will be

$$x_t = (\lambda)^n x_0 + \sum_{i=0}^{n-1} (\lambda)^i \phi + \sum_{i=0}^{n-1} (\lambda)^i \varepsilon_{t-i}$$

2. Avoiding explosive behavior of x_t when $n \to \infty$, we have to impose

$$|\lambda| < 1$$

3. And get

$$x_t = \sum_{i=0}^{n-1} (\lambda)^i \phi + \sum_{i=0}^{n-1} (\lambda)^i \varepsilon_{t-i}$$

4. Assuming certainty equivalence

$$x_t = \sum_{i=0}^{n-1} (\lambda)^i \phi = \frac{\phi}{1-\lambda}$$

Solution by iterating forward: the entire model

1. Knowing the equilibrium level of x_t

$$x_t = \bar{x} = \frac{\phi}{1-\lambda}$$

2. Plugging it into the equilibrium level of y_t

$$y_t = \sum_{i=0}^{n-1} (\beta)^i \alpha + \sum_{i=0}^{n-1} (\beta)^i E_t x_{t+i}$$

3. Leads to

$$\begin{aligned} y_t &= \bar{y} = \frac{\alpha}{1-\beta} + \sum_{i=0}^{n-1} \beta^i \bar{x} = \frac{\alpha}{1-\beta} + \frac{1}{1-\beta}\bar{x} \\ &= \frac{\alpha + \phi/(1-\lambda)}{1-\beta} \end{aligned}$$

Numerical simulation

1. Now consider that the shock follows an AR(1) process

$$v_t = \rho v_{t-1} + \varepsilon_t, \quad |\rho| < 1$$

2. What happens to the whole model?
3. Parameters

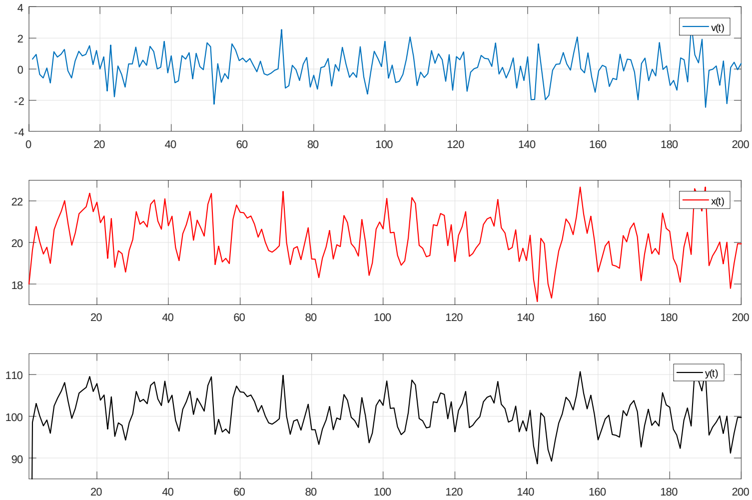
$$\beta = 0.75, \phi = 10, \lambda = 0.5, \alpha = 5, \rho = 0.8$$

1. Deterministic steady state

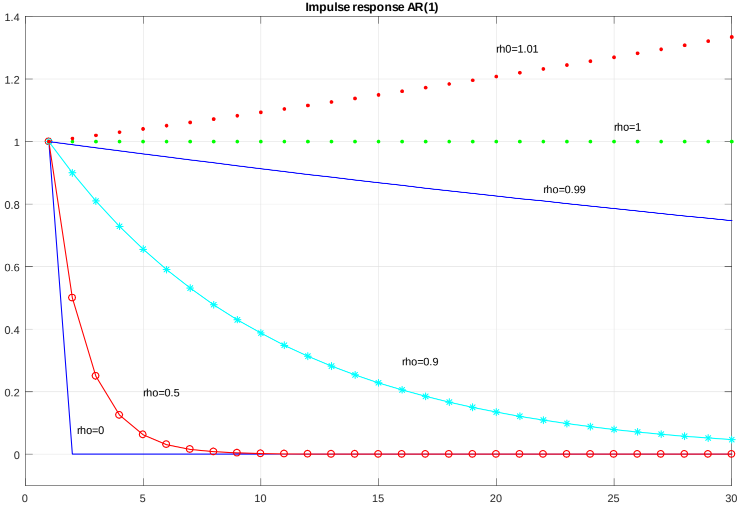
$$\bar{x} = 20 \ \ ; \ \ \bar{y} = 100$$

2. Time series and impulse response functions (next figures)

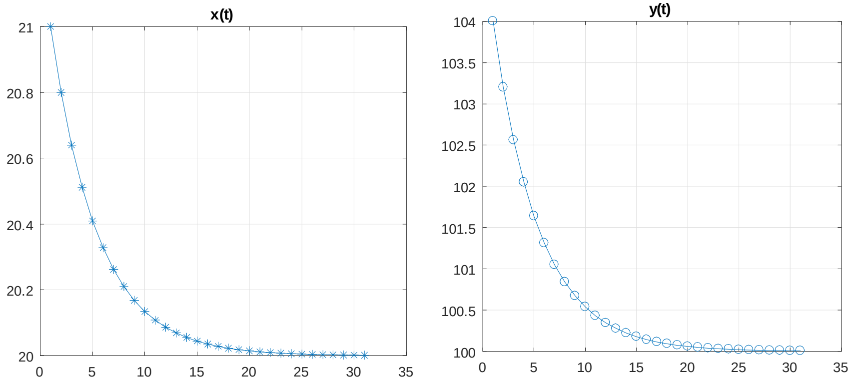
Time series


v(t)
x(t)
y(t)
4
2
0
-2
-4
0
20
40
60
80
100
120
140
160
180
200
22
20
18
110
100
90

Imp. response functions v(t): different values for rho

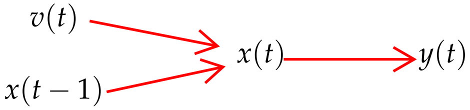


Impulse response functions: rho=0.8



The fundamental strategy to solve DSGE Models

1. Iterate forward and guarantee stability:
   1. Forward looking (also named as "control" or "jump") variables: $|1/\beta| > 1$
   2. Backward looking or predetermined variables: $|\lambda| < 1$
2. Solve the predetermined block
3. Then insert this solution into the forward looking block
4. We get a solution like this



II – More complicated types of models

The New Keynesian Model: one version

1. The baseline version includes five equations:

$$
\begin{aligned}
x_t &= E_t x_{t+1} - \frac{1}{\sigma}\left(i_t - E_t \pi_{t+1}\right) && \text{(IS)} \\
\pi_t &= \beta \cdot E_t \pi_{t+1} + \kappa x_t + u_t && \text{(AS)} \\
i_t &= \delta \pi_t + u_t && \text{(Rule)} \\
v_{t+1} &= \rho_v v_t + \epsilon^v_{t+1} && \text{(Shock v)} \\
u_{t+1} &= \rho_u u_t + \epsilon^u_{t+1} && \text{(Shock u)}
\end{aligned}
$$

2. The model is linear
3. Can we apply the same strategy?

The simple RBC model

1. The baseline version includes seven equations:

$$
\begin{aligned}
R_{t+1} &\equiv \alpha\left(Y_{t+1}/K_t\right) + 1 - \delta && \text{(S1)} \\
C_t^{-\eta} &= \beta E_t(C_{t+1}^{-\eta} R_{t+1}) && \text{(S2)} \\
Y_t/N_t &= \left[\xi/\left(1-\alpha\right)\right] C_t^{\eta} && \text{(S3)} \\
K_t &= (1-\delta)K_{t-1} + I_t && \text{(S4)} \\
Y_t &= A_t K_{t-1}^{\alpha} N_t^{1-\alpha} && \text{(S5)} \\
C_t + I_t &= Y_t && \text{(S6)} \\
\ln A_t &= (1-\rho)\ln A^* + \rho \ln A_{t-1} + \varepsilon_t && \text{(S7)}
\end{aligned}
$$

2. **A large nonlinear system** of stochastic difference equations
3. **Closed form solution is not possible** to be obtained for this model
4. Can we apply the strategy presented above in this case?

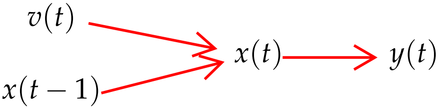
Two general approaches to solve this type of models

1. Local methods
    1. Linearization around steady state
    2. Perturbation (very similar to linearization: uses higher order approximations)
2. Global methods
    1. Projection methods
    2. Weighted residual methods
3. In this course we will cover **Linearization and Projection**

III – Linearization and the Blanchard-Kahn approach

The basic idea behind the B-K method

1. Apply the Jordan decomposition to transform our complicated models into two distinct blocks:
   1. The block including only predetermined variables
   2. The other block including only forward looking variables
2. Then we can apply exactly the same strategy followed in part I.
   1. Iterate forward
   2. Firstly, the predetermined block
   3. Then, the forward looking block.
3. And we get the same type of results



Blanchard, O., and C.M. Kahn. (1980). The solution of linear difference models under rational expectations. *Econometrica* 48(5), 1305–1311.

The Jordan decomposition

1. Compute the Jordan canonical form (also called Jordan normal form) of a symbolic or numeric matrix A
2. Our model comprises: a set of predetermined variables (x_t), a set of forward looking variables (y_t), and a set of exogenous shocks (v_t)
3. Write the model in state space form

$$\left[\begin{array}{c} w_{t+1} \\ E_t y_{t+1} \end{array}\right] = A \left[\begin{array}{c} w_t \\ y_t \end{array}\right] + B v_{t+1} \tag{2}$$

4. The Jordan decomposition of (A)

$$A = P \Lambda P^{-1}$$

5. Λ is a diagonal matrix with the **eigenvalues** of A along its leading diagonal and zeros in the remaining entries.
6. P contains the inverse matrix of the generalized **eigenvectors** of A as columns

The model with the Jordan decomposition

1. Apply the decomposition

$$\begin{bmatrix} w_{t+1} \\ E_t y_{t+1} \end{bmatrix} = P \Lambda P^{-1} \begin{bmatrix} w_t \\ y_t \end{bmatrix} + B v_{t+1} \tag{3}$$

2. Multiply both sides by P^{-1}

$$P^{-1} \begin{bmatrix} w_{t+1} \\ E_t y_{t+1} \end{bmatrix} = \Lambda P^{-1} \begin{bmatrix} w_t \\ y_t \end{bmatrix} + \underbrace{P^{-1} B}_{=R} \cdot v_{t+1} \tag{4}$$

The model with the Jordan decomposition

1. Partition P^{-1} and Λ to get

$$\underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w_{t+1} \\ E_t y_{t+1} \end{bmatrix}}_{Et \begin{bmatrix} \tilde{w}_{t+1} \\ \tilde{y}_{t+1} \end{bmatrix}} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w_t \\ y_t \end{bmatrix}}_{\begin{bmatrix} \tilde{w}_t \\ \tilde{y}_t \end{bmatrix}} + \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} v_{t+1} \tag{5}$$

2. So our transformed model looks much easier now

$$\begin{bmatrix} \tilde{w}_{t+1} \\ E_t \tilde{y}_{t+1} \end{bmatrix} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} \tilde{w}_t \\ \tilde{y}_t \end{bmatrix} + \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} v_{t+1}$$

The two decoupled blocks

Transformed model written down as a set of decoupled equations

$$\tilde{w}_{t+1} = \Lambda_1 \tilde{w}_t + R_1 v_{t+1} \qquad \text{(Stable block)}$$

$$E_t \tilde{y}_{t+1} = \Lambda_2 \tilde{y}_t + R_2 v_{t+1} \qquad \text{(Unstable block)}$$

1. We can now apply our old strategy
   1. Solve the unstable transformed block forward and get: $\tilde{y}_t^*$
   2. Solve the stable transformed block backwards and get: $\tilde{w}_t^*$
2. Insert the results back into the original problem

Solving the unstable block

1. Iterating forward this block, we get

$$E_t \tilde{y}_{t+n} = (\Lambda_2)^n \tilde{y}_t$$

2. If we have

$$|\Lambda_2| > 1$$

3. Then, the only stable solution will be

$$\tilde{y}_t^* = 0, \ \forall t$$

4. Now from our definition in eq. (5), we know that

$$\tilde{y}_t^* = P_{21} \cdot w_t^* + P_{22} \cdot y_t^* = 0$$

5. From which

$$y_t^* = \left[-P_{22}^{-1} P_{21} \right] \cdot w_t^* \qquad (6)$$

6. **Notice that this our old result:** forward looking variables depending upon predetermined ones.

Solving the stable block

1. Iterating forward this block, we get

$$\tilde{w}_{t+n} = (\Lambda_1)^n \tilde{w}_t \quad , \qquad |\Lambda_1| < 1$$

2. If we assume that

$$|\Lambda_1| < 1$$

3. The process is stable, and from eq. (5), we get

$$\tilde{w}_t^* = P_{11} \cdot w_t^* + P_{12} \cdot y_t^* \tag{7}$$

4. Now insert eq. (6) into (7), and get

$$\tilde{w}_t^* = \underbrace{\left[P_{11} - P_{12} P_{22}^{-1} P_{21} \right]}_{D} \cdot w_t^* \tag{8}$$

Solving the stable block (cont.)

1. But as from eq. (Stable block), we have

$$\tilde{w}_{t+1} = \Lambda_1 \tilde{w}_t + R_1 v_{t+1}$$

2. And as from eq.(8) we have

$$\tilde{w}_t^* = D \cdot w_t^*$$

3. Then

$$D \cdot w_{t+1}^* = \tilde{w}_{t+1}^* \qquad\qquad \tilde{w}_t^* = D \cdot w_t^*$$

$$\tilde{w}_{t+1} = \Lambda_1 \tilde{w}_t + R_1 v_{t+1}$$

4. From which we finally get

$$w_{t+1}^* = \left[D^{-1} \Lambda_1 D \right] w_t^* + \left[D^{-1} R_1 \right] v_{t+1} \tag{9}$$

Summarizing

1. Write down your model in state space form
2. Apply the Jordan decomposition
3. Decouple the system into two blocks
4. Make sure one eigenvalue is larger than 1 in modulus, the other lower than.1 in modulus.
5. End up with the two fundamental results

$$
\begin{aligned}
y_t^* &= \left[-P_{22}^{-1} P_{21}\right] \cdot w_t^* \\
w_{t+1}^* &= \left[D^{-1} \Lambda_1 D\right] w_t^* + \left[D^{-1} R_1\right] v_{t+1}
\end{aligned}
$$

with $D = P_{11} - P_{12} \left(P_{22}\right)^{-1} P_{21}$

IV – Matlab code for the B-K method

A simple model

1. Consider a simple version of the RBC model with no shocks. Write it down in state space form

$$E_t A x_{t+1} = B x_t$$

2. Assuming that B is invertible

$$C = B^{-1} A$$

3. We get

$$E_t C x_{t+1} = x_t \tag{10}$$

A simple model

1. Our case follows the paper by Martin Ellison ("Real Business Cycle Theory")
2. This example is interesting because it shows the characteristic matrix of the system on its **left hand side**
3. Notice that the model has

$$\begin{aligned} \text{Predetermined variables} &: A_t, k_t \\ \text{Forward looking variables} &: l_t, c_t \end{aligned}$$

4. The model has been previously linearized in the vicinity of the steady state

Model in state space form

1. The model written in state space form

$$\begin{bmatrix} E_t \end{bmatrix} \begin{bmatrix} 0.145 & -0.087 & 0.087 & -1 \\ 0 & 2.0252 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{A}_{t+1} \\ \hat{k}_{t+1} \\ \hat{l}_{t+1} \\ \hat{c}_{t+1} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0.7728 & 2.1318 & 0.4637 & -0.5702 \\ 0.95 & 0 & 0 & 0 \\ 1 & 0.4 & -0.4 & -1 \end{bmatrix} \begin{bmatrix} \hat{A}_t \\ \hat{k}_t \\ \hat{l}_t \\ \hat{c}_t \end{bmatrix}$$

The model ready to the computer

1. The model looks like

$$\left[\ E_t\ \right]\underbrace{\begin{bmatrix} 1.0526 & 0 & 0 & 0 \\ -0.8801 & 0.8383 & -0.058 & 0.6663 \\ 2.1139 & 0.6208 & 0.1595 & -1.8337 \\ -0.145 & 0.087 & -0.087 & 1 \end{bmatrix}}_{C}\begin{bmatrix} \hat{A}_{t+1} \\ \hat{k}_{t+1} \\ \hat{l}_{t+1} \\ \hat{c}_{t+1} \end{bmatrix}=\begin{bmatrix} \hat{A}_{t} \\ \hat{k}_{t} \\ \hat{l}_{t} \\ \hat{c}_{t} \end{bmatrix}$$

2. Now we can start writing down the Matlab code
3. Notice that the characteristic matrix is on the left hand side of the system
4. This implies that the solution to the predetermined block has Λ_1^{-1} (not Λ_1) on the right hand side (see handwritten notes)

$$w_{t+1}^{*}=\underbrace{\left[D^{-1}\Lambda_1^{-1}D\right]}_{z}w_t^{*}$$

The Jordan decomposition

Matlab code

```
[ve,MU]=eig(C);
% ve:  generalized eigenvectors; MU: eigenvalues.
```

Matlab output

$$ve = \begin{bmatrix} -0.0000 & 0.0000 & -0.2708 & 0.0000 \\ -0.0000 & 0.8418 & -0.7543 & -0.7071 \\ 0.9962 & -0.2924 & 0.0572 & -0.7071 \\ 0.0867 & 0.4537 & -0.5954 & 0.0000 \end{bmatrix}$$

$$MU = \begin{bmatrix} -0.000 & 0 & 0 & 0 \\ 0 & 1.2175 & 0 & 0 \\ 0 & 0 & 1.0526 & 0 \\ 0 & 0 & 0 & 0.7803 \end{bmatrix}$$

Sorting the eigenvalues in descending order

1. Two stable eigenvalues in the first two rows (predetermined variables)
2. Two unstable eigenvalues in the last two rows (forward looking variables)

Matlab code

```
t=flipud(sortrows([diag(MU) ve′]))
%Eigenvalues in the first column, transposed eigenvectors in each eigenvalue line
```

Matlab output

$$t = \begin{bmatrix} 1.2175 & 0.0000 & 0.8418 & -0.2924 & 0.4537 \\ 1.0526 & -0.2708 & -0.7543 & 0.0572 & -0.5954 \\ 0.7803 & 0.0000 & -0.7071 & -0.7071 & 0.0000 \\ -0.0000 & -0.0000 & -0.0000 & 0.9962 & 0.0867 \end{bmatrix}$$

Rearranging eigenvalues and eigenvectors

Matlab code

```
MU=diag(t(:,1)) % Eigenvalues in descending order along the..
ve=t(:,2:5) % Eigenvectors sorted out according to eigenvalue
```

Matlab output

$$MU = \begin{bmatrix} 1.2175 & 0 & 0 & 0 \\ 0 & 1.0526 & 0 & 0 \\ 0 & 0 & 0.7803 & 0 \\ 0 & 0 & 0 & -0.000 \end{bmatrix}$$

$$ve = \begin{bmatrix} -0.0000 & 0.8418 & -0.2924 & 0.4537 \\ -0.2708 & -0.7543 & 0.0572 & -0.5954 \\ 0.0000 & -0.7071 & -0.7071 & 0.0000 \\ -0.0000 & -0.0000 & 0.9962 & 0.0867 \end{bmatrix}$$

Rearranging eigenvectors in the right form

Matlab code

```
P=inv(ve') % Eigenvectors in the right form
      % (in columns) and their inverse
```

Matlab output

$$P = \begin{bmatrix} -4.4526 & 0.1575 & -0.1575 & 1.8104 \\ -3.6930 & -0.0000 & -0.0000 & 0.0000 \\ -1.3614 & -1.2267 & -0.1875 & 2.1553 \\ -2.0611 & -0.8245 & 0.8245 & 2.0611 \end{bmatrix}$$

Partioning MU

Matlab code

```
MU1=MU(1:2,1:2) % Elements of MU: lines 1&2; columns 1&2
MU2=MU(3:4,3:4) % Elements of MU: lines 3&4; columns 3&4
```

Matlab output

$$MU1 = \begin{bmatrix} 1.2175 & 0 \\ 0 & 1.0526 \end{bmatrix} \quad , \quad MU2 = \begin{bmatrix} 0.7803 & 0 \\ 0 & -0.000 \end{bmatrix}$$

Partioning of P

Matlab code

```
P11=P(1:2,1:2)
P12=P(1:2,3:4)
P21=P(3:4,1:2)
P22=P(3:4,3:4)
```

Matlab output

$$P11 = \begin{bmatrix} -4.4526 & 0.1575 \\ -3.6930 & -0.0000 \end{bmatrix} \quad , \quad P12 = \begin{bmatrix} -0.1575 & 1.8104 \\ -0.0000 & 0.0000 \end{bmatrix}$$

$$P21 = \begin{bmatrix} -1.3614 & -1.2267 \\ -2.0611 & -0.8245 \end{bmatrix} \quad , \quad P22 = \begin{bmatrix} -0.1875 & 2.1553 \\ 0.8245 & 2.0611 \end{bmatrix}$$

Final step: the dynamics of the four variables (Algebra)

1. We can now apply the two fundamental results

$$y_t^* = \underbrace{\left[-P_{22}^{-1}P_{21}\right]}_{f} \cdot w_t^*$$

$$w_{t+1}^* = \underbrace{\left[D^{-1}\Lambda_1 D\right]}_{z} w_t^* + \underbrace{\left[D^{-1}R_1\right]}_{g} v_{t+1}$$

   with $P_{11} - P_{12}\left(P_{22}\right)^{-1} P_{21} = D$

2. Notice that in this current example there are no shocks: $v_{t+1} = 0$
3. Notice also that our matrix D was on the left hand side of eq. (10): so in z instead of having Λ_1^{-1} we should have Λ_1 (see handwritten notes).

Final step: marginal coefficients (Matlab)

Matlab code

```
f=-inv(P22)*P21
z=inv(P11-P12*inv(P22)*P21)*inv(MU1)*(P11-P12*inv(P22)*P21)
```

Matlab output

$$f = \begin{bmatrix} 0.7563 & -0.3473 \\ 0.6975 & 0.5389 \end{bmatrix} \quad , \quad z = \begin{bmatrix} 0.9500 & -0.0000 \\ 0.3584 & 0.8214 \end{bmatrix}$$

1. Translate that into the model language
   1. f is related with the forward looking variables
   2. z is related to the predetermined variables

Final step: dynamics of the 4 variables (solution)

1. As in our model

$$y_t^* = \begin{bmatrix} \hat{l}_t \\ \hat{c}_t \end{bmatrix}, \quad w_t^* = \begin{bmatrix} \hat{A}_t \\ \hat{k}_t \end{bmatrix}$$

2. Then

$$y_t^* = f \cdot w_t^* \quad , \quad w_{t+1}^* = z \cdot w_t^*$$

Solution

$$\begin{aligned} \hat{l}_t &= 0.7563\hat{A}_t - 03473\hat{k}_t \\ \hat{c}_t &= 0.6975\hat{A}_t + 0.5389\hat{k}_t \\ \hat{A}_{t+1} &= 0.95\hat{A}_t + 0\hat{k}_t \\ \hat{k}_{t+1} &= 0.3584\hat{A}_t + 0.8214\hat{k}_t \end{aligned}$$

1. Check this solution is correct: in the structural form of the model we had $\hat{A}_{t+1} = \rho\hat{A}_t$ and we assumed $\rho = 0.95$. So ...

V – The NKM in the Blanchard-Kahn framework

The NKM in the Blanchard-Kahn framework

1. The baseline version includes five equations:

$$
\begin{aligned}
x_t &= E_t x_{t+1} - \frac{1}{\sigma}\left(i_t - E_t \pi_{t+1}\right) && \text{(IS)} \\
\pi_t &= \beta \cdot E_t \pi_{t+1} + \kappa x_t + u_t && \text{(AS)} \\
i_t &= \delta \pi_t + v_t && \text{(Rule)} \\
v_{t+1} &= \rho_v v_t + \epsilon_{t+1}^{v} && \text{(Shock v)} \\
u_{t+1} &= \rho_u u_t + \epsilon_{t+1}^{u} && \text{(Shock u)}
\end{aligned}
$$

2. Can we apply the same strategy?

$$
\begin{aligned}
\text{Predetermined} &: \quad v_t, u_t \\
\text{Forward looking} &: \quad x_t, \pi_t
\end{aligned}
$$

The NKM in state space

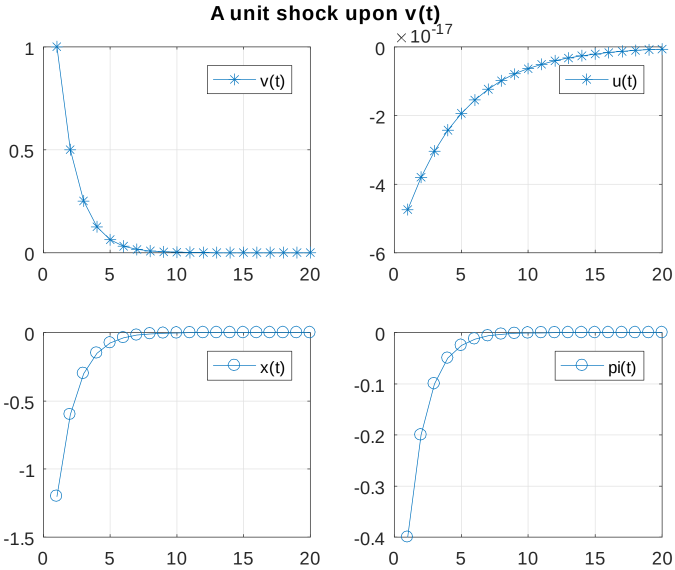
1. The model can be written as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{\sigma} \\ 0 & 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} v_{t+1} \\ u_{t+1} \\ E_t x_{t+1} \\ E_t \pi_{t+1} \end{bmatrix} =$$

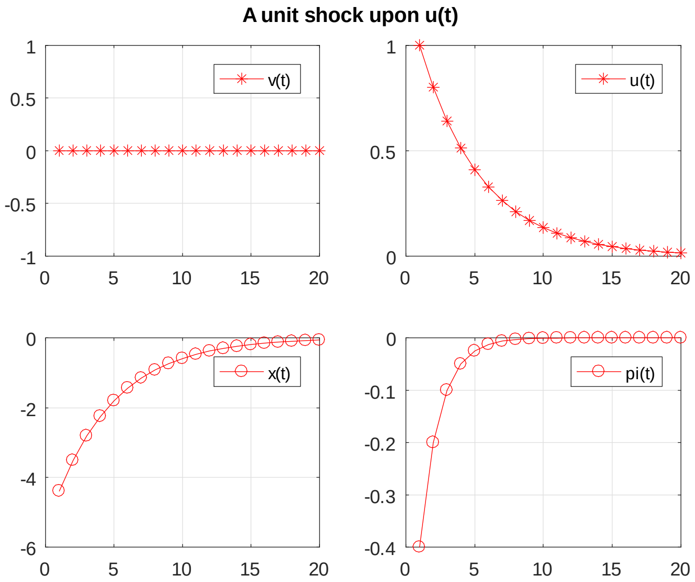
$$\begin{bmatrix} \rho_v & 0 & 0 & 0 \\ 0 & \rho_u & 0 & 0 \\ \frac{1}{\sigma} & 0 & 1 & \frac{1}{\sigma}\delta \\ 0 & -1 & -\kappa & 1 \end{bmatrix} \begin{bmatrix} v_t \\ u_t \\ x_t \\ \pi_t \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon^v_{t+1} \\ \epsilon^u_{t+1} \end{bmatrix}$$

2. Look at the routine **NKM_Topics_Macro_V2.m** and the IRF that come out of the model

The NKM: IRF from a shock upon v(t)



The NKM: IRF from a shock upon u(t)



VI – Exercise and Required readings

Exercise: the NKM with shocks to the natural real interest rate

1. The baseline version includes five equations:

$$
\begin{aligned}
x_t &= E_t x_{t+1} - \frac{1}{\sigma}\left(i_t - E_t \pi_{t+1} - r_t^n\right) && \text{(IS)} \\
\pi_t &= \beta \cdot E_t \pi_{t+1} + \kappa x_t && \text{(AS)} \\
i_t &= \phi_\pi \pi_t + \phi_x x_t && \text{(Monetary policy rule)} \\
r_t^n &= \rho r_{t-1}^n + \epsilon_t && \text{(Natural real interest rate)}
\end{aligned}
$$

2. with

$$
k = \frac{(1-\theta)(1-\theta\beta)}{\theta(\sigma+1)}
$$

Exercise: the NKM with shocks to the natural real interest rate

1. Notice that in this case

$$\begin{aligned} \text{Predetermined} &: \quad r_t^n \\ \text{Forward looking} &: \quad x_t, \pi_t \end{aligned}$$

2. With the following calibration

$$\beta = 0.99, \sigma = 1, \theta = 2/3, \phi_\pi = 1.5, \phi_x = 0.1, \rho = 0.8$$

develop a routine that is capable of producing the IRF of the model. (If you feel confident, try also to represent the time series associated with the model).

Required reading

Blanchard, O., and C.M. Kahn. (1980). The solution of linear difference models under rational expectations. *Econometrica* 48(5), 1305–1311.

Ellison, Martin (2009). Real Business Cycle Theory, mimeo, University of Oxford.