Computational Macroeconomics
Exercises

Week 2

Willi Mutschler
willi@mutschler.eu

Version: August 16, 2022

Contents

1. RBC model: preprocessing and steady-state in Dynare

Consider the basic Real Business Cycle (RBC) model with leisure. The representative household maximizes present as well as expected future utility

$$\max E_t \sum_{j=0}^{\infty} \beta^j U_{t+j}$$

with $\beta < 1$ denoting the discount factor and E_t is expectation given information at time t. The contemporaneous utility function

$$U_t = \gamma \log(C_t) + \psi \log(1 - L_t)$$

is additively separable and has two arguments: consumption C_t and labor L_t. The marginal utility of consumption is positive, whereas more labor reduces utility. Accordingly, γ is the consumption utility parameter and ψ the labor disutility parameter. In each period the household takes the real wage W_t as given and supplies perfectly elastic labor service to the representative firm. In return, she receives real labor income in the amount of $W_t L_t$ and, additionally, real profits DIV_t from the firm as well as revenue from lending capital K_{t-1} at interest rate R_t to the firms, as it is assumed that the firm and capital stock are owned by the household. Income and wealth are used to finance consumption C_t and investment I_t. In total, this defines the (real) budget constraint of the household:

$$C_t + I_t = W_t L_t + R_t K_{t-1} + DIV_t$$

The law of motion for capital K_t at the end of period t is given by

$$K_t = (1 - \delta) K_{t-1} + I_t$$

and δ is controlling depreciations. [1] Assume that the transversality condition is full-filled.

Productivity A_t is the driving force of the economy and evolves according to

$$\log A_t = \rho_A \log A_{t-1} + \varepsilon_t^A$$

where ρ_A denotes the persistence parameter and ε_t^A is assumed to be normally distributed with mean zero and variance σ_A^2.

Real profits DIV_t of the representative firm are revenues from selling output Y_t minus costs from labor $W_t L_t$ and renting capital $R_t K_{t-1}$:

$$DIV_t = Y_t - W_t L_t - R_t K_{t-1}$$

The representative firm maximizes expected profits

$$\max E_t \sum_{j=0}^{\infty} \beta^j Q_{t+j} DIV_{t+j}$$

subject to a Cobb-Douglas production function

$$f(K_{t-1}, L_t) = Y_t = A_t K_{t-1}^{\alpha} L_t^{1-\alpha}$$

The discount factor takes into account that firms are owned by the household, i.e. $\beta^j Q_{t+j}$ is the present value of a unit of consumption in period $t + j$ or, respectively, the marginal utility of an additional unit of profit; therefore $Q_{t+j} = \frac{\partial U_{t+j}/\partial C_{t+j}}{\partial U_t/\partial C_t}$.

Finally, we have the non-negativity constraints $K_t \geq 0$, $C_t \geq 0$ and $0 \leq L_t \leq 1$ and clearing of the labor as well as goods market in equilibrium, i.e.

$$Y_t = C_t + I_t$$

[1]Note that we use the end-of-period timing for capital, i.e. K_t instead of K_{t+1}, because the investment decision is done in period t and hence capital is also determined in t. In older papers you will often find beginning-of-period timing for capital, so always think about when it is decided and determined.

1. Briefly provide intuition behind the transversality condition.
2. Show that the first-order conditions of the representative household are given by
$$U_t^C = \beta E_t \left[U_{t+1}^C \left(1 - \delta + R_{t+1}\right)\right]$$
$$W_t = -\frac{U_t^L}{U_t^C}$$
where $U_t^C = \gamma C_t^{-1}$ and $U_t^L = \frac{-\psi}{1-L_t}$. Interpret these equations in economic terms.
3. Show that the first-order conditions of the representative firm are given by
$$W_t = f_L$$
$$R_t = f_K$$
where $f_L = (1-\alpha) A_t \left(\frac{K_{t-1}}{L_t}\right)^{\alpha}$ and $f_K = \alpha A_t \left(\frac{K_{t-1}}{L_t}\right)^{1-\alpha}$. Interpret these equations in economic terms.
4. Derive the steady-state of the model, in the sense that there is a set of values for the endogenous variables that in equilibrium remain constant over time.
5. Discuss how to calibrate the following parameters α, β, δ, γ, ψ, ρ_A and σ_A.
6. Write a script for this RBC model with a feasible calibration for an OECD country that computes the steady-state of the model.
7. Write a Dynare mod file for this RBC model with a feasible calibration for an OECD country and compute the steady-state of the model by using a `steady_state_model` block. Compare this to the steady-state computed above.
8. Now assume a contemporaneous utility function of the CRRA (constant Relative Risk Aversion) type: [2]
$$U_t = \gamma \frac{C_t^{1-\eta_C} - 1}{1-\eta_C} + \psi \frac{(1-L_t)^{1-\eta_L} - 1}{1-\eta_L}$$
   a) Derive the model equations and steady-state analytically.
   b) Write a script to compute the steady-state for this model.
   c) Write a Dynare mod file and compute the steady-state for this model by using a helper function in the `steady_state_model` block.

Readings

- McCandless (2008).

[2] Note that due to L'Hopital's rule $\eta_C = \eta_L = 1$ implies the original specification, $U_t = \gamma \log C_t + \psi \log(1 - L_t)$.

2. Fiscal Policy in General Equilibrium: preprocessing and steady-state

Model description

Consider a version of the Baxter and King (1993) model.

Households Let C_t denote real consumption, N_t real labor supply, I_t private investment and K_t the end-of-period private capital stock. The representative household maximizes its expected life-time utility

$$\max E_t \sum_{t=0}^{\infty} \beta^t \left[log(C_t) + \theta_l log(1 - N_t) + \Gamma(G_t^B, I_t^B) \right]$$

subject to

$$C_t + I_t = (1 - \tau_t)(w_t N_t + r_t K_{t-1}) + TR_t$$

where β denotes the discount rate, θ_l the Frisch elasticity of labor, w_t the real wage, r_t the real interest rate and TR_t real lump-sum transfers. $\Gamma(G_t^B, I_t^B)$ is a general function of public consumption G_t^B and public investment I_t^B such that it is non-decreasing in each of its arguments. Optimality is given by the consumption-leisure choice

$$(1 - \tau_t) w_t = \theta_l \frac{C_t}{1 - N_t} \tag{1}$$

and the savings decision

$$\lambda_t = \beta E_t \left\{ \lambda_{t+1} \left[(1 - \delta) + (1 - \tau_{t+1}) r_{t+1} \right] \right\} \tag{2}$$

where

$$\lambda_t = \frac{1}{C_t} \tag{3}$$

denotes marginal utility of consumption. The private and public capital stocks evolve according to

$$K_t = (1 - \delta) K_{t-1} + I_t \tag{4}$$

$$K_t^B = (1 - \delta) K_{t-1}^B + I_t^B \tag{5}$$

with δ denoting the depreciation rate.

Firms Firms maximize profits by choosing factor inputs according to

$$\max_{N_t, K_{t-1}} Y_t - w_t N_t - r_t K_{t-1}$$

subject to

$$Y_t = z_t (K_{t-1}^B)^\eta (K_{t-1})^\alpha (N_t)^{1-\alpha} \tag{6}$$

where η denotes productivity of public capital and α the share of capital in production. Taking factor prices as given, factor demand is determined by

$$w_t N_t = (1 - \alpha) Y_t \tag{7}$$

$$r_t K_{t-1} = \alpha Y_t \tag{8}$$

Productivity evolves according to

$$\log\left(\frac{z_t}{\bar{z}}\right) = \rho_z \log\left(\frac{z_{t-1}}{\bar{z}}\right) + \varepsilon_t^z \tag{9}$$

where ρ_z is a smoothing parameter and $\varepsilon_t^z \overset{iid}{\sim} N(0, \sigma_z^2)$.

Fiscal authority The fiscal authority faces the budget constraint

$$G_t^B + I_t^B + TR_t = \tau_t(w_t N_t + r_t K_{t-1}) \tag{10}$$

and its behavior is described by exogenous AR(1) processes

$$G_t^B - \bar{G}^B = \rho_{G^B}\left({G^B}_{t-1} - \bar{G}^B\right) + \varepsilon_t^{G^B} \tag{11}$$

$$I_t^B - \bar{I}^B = \rho_{I^B}\left({I^B}_{t-1} - \bar{I}^B\right) + \varepsilon_t^{I^B} \tag{12}$$

$$\log\left(\frac{\tau_t}{\bar{\tau}}\right) = \rho_\tau \log\left(\frac{\tau_{t-1}}{\bar{\tau}}\right) + \varepsilon_t^\tau \tag{13}$$

where $\rho_{G^B}, \rho_{I^B}, \rho_\tau$ are smoothing parameters and

$$\varepsilon_t^{G^B} \overset{iid}{\sim} N(0, \sigma_{G^B}^2), \qquad \varepsilon_t^{I^B} \overset{iid}{\sim} N(0, \sigma_{I^B}^2), \qquad \varepsilon_t^\tau \overset{iid}{\sim} N(0, \sigma_\tau^2)$$

Notice that the inclusion of TR_t implies a balanced budget rule, i.e. there is no debt in the model.

Market clearing Market clearing implies that whatever is consumed by households must be produced

$$Y_t = C_t + I_t + G_t^B + I_t^B \tag{14}$$

Summary Overall, the Baxter and King (1993) model can be summarized through equations (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), and (14).

Exercises

Willi, a fellow student, wants to assess how changes in fiscal policy (taxation & spending) affect the real economy.

1. From his class on introductory macroeconomics Willi remembers that as a first step it is always important to distinguish between endogenous and exogenous variables as well as model parameters. Can you help him with that?

2. Willi is quite clever about calibrating the model parameters. In particular, he is interested in targeting steady-state values of the model:

Target	Symbol	Value
steady-state output level	$\bar{Y}$	1
steady-state public consumption	$\bar{G^B}$	$0.2\bar{Y}$
steady-state public investment	$\bar{I^B}$	$0.02\bar{Y}$
steady-state transfers	$\bar{TR}$	0
steady-state real wage	$\bar{w}$	2
steady-state labor supply	$\bar{N}$	1/3

   Furthermore, he thinks that public-capital productivity should be **lower** than the capital share in production. Regarding the exogenous processes he would like mild persistence (ρ's equal to 0.75) and small shock standard errors of 1%. Can you provide a calibration for all model parameters meeting his targets and economic intuition?
   Hint: First, set some reasonable values for β, δ and η. Then begin with the target values and try to derive the steady-states of all other endogenous variables and the implied parameter values.

3. Willi has heard of the powerful toolbox DYNARE, so he asks you to help him set up this model in DYNARE. Write a DYNARE mod-file for this model, commenting each step such that Willi clearly understands each block.

Readings

- Baxter and King (1993)

3. Matrix Algebra

Let

$$A = \begin{pmatrix} 0.5 & 0 & 0 \\ 0.1 & 0.1 & 0.3 \\ 0 & 0.2 & 0.3 \end{pmatrix} \qquad \Sigma_u = \begin{pmatrix} 2.25 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 0.74 \end{pmatrix} \qquad R = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}$$

1. Compute the eigenvalues of A. What would this imply for the system $y_t = c + Ay_{t-1} + u_t$ with u_t being white noise?
2. Consider the matrices D: $m \times n$, E: $n \times p$ and F: $p \times k$. Show that
$$vec(DEF) = (F' \otimes D)\, vec(E),$$
where $\otimes$ is the Kronecker product and vec the vectorization operator.
3. Show that R is an orthogonal matrix. Why is this matrix called a rotation matrix?
4. Compute a regular lower triangular matrix $W \in \mathbb{R}^{3\times 3}$ and a diagonal matrix $\Sigma_\varepsilon \in \mathbb{R}^{3\times 3}$ such that $\Sigma_u = W\Sigma_\varepsilon W'$.
Hint: Use the Cholesky factorization $\Sigma_u = PP' = W\Sigma_\varepsilon^{\frac{1}{2}}(W\Sigma_\varepsilon^{\frac{1}{2}})'$.
5. Solve the discrete Lyapunov matrix equation $\Sigma_y = A\Sigma_y A' + \Sigma_u$ using
   a) the Kronecker product and vectorization
   b) the following iterative algorithm:
$$\begin{aligned} \Sigma_{y,0} &= I, A_0 = A, \Sigma_{u,0} = \Sigma_u \\ \Sigma_{y,i+1} &= A_i \Sigma_{y,i} A_i' + \Sigma_{u,i} \\ \Sigma_{u,i+1} &= A_i \Sigma_{u,i} A_i' + \Sigma_{u,i} \\ A_{i+1} &= A_i A_i \end{aligned}$$
   Write a loop until either a maximal number of iterations (say 500) is reached or each element of $\Sigma_{y,i+1} - \Sigma_{y,i}$ is less than 10^{-25} in absolute terms.
   c) Compare both approaches for A and Σ_u given above.

Readings:

- E. W. Anderson et al. (1996, Ch. 4.2)
- B. D. O. Anderson and Moore (2005, Ch. 6.7)
- Uribe and Schmitt-Grohe (2017, Ch. 4.10)

4. Law Of Large Numbers

Let $Y_1, Y_2, \ldots$ be an i.i.d. sequence of arbitrarily distributed random variables with finite variance σ_Y^2 and expectation μ. Define the sequence of random variables

$$\overline{Y}_T = \frac{1}{T}\sum_{t=1}^{T} Y_t$$

1. Briefly outline the intuition behind the "law of large numbers".

2. Write a program to illustrate the law of large numbers for uniformly distributed random variables (you may also try different distributions such as normal, gamma, geometric, student's t with finite or infinite variance). To this end, do the following:

   - Set $T = 10000$ and initialize the $T \times 1$ output vector u.
   - Choose values for the parameters of the uniform distribution. Note that $E[u] = (a+b)/2$, where a is the lower and b the upper bound of the uniform distribution.
   - For $t = 1, ..., T$ do the following computations:
     - Draw t random variables from the uniform distribution with lower bound a and upper bound b.
     - Compute and store the mean of the drawn values in your output vector at position t.
   - Plot your output vector and add a line to indicate the theoretical mean of the uniform distribution.

3. Now suppose that the sequence $Y_1, Y_2, \ldots$ is an $AR(1)$ process:

$$Y_t = \phi Y_{t-1} + \varepsilon_t$$

   where $\varepsilon_t \sim iid(0, \sigma_\varepsilon^2)$ is not necessarily normally distributed and $|\phi| < 1$. Illustrate numerically that the law of large numbers still holds despite the intertemporal dependence.

Readings

- Ploberger (2010)
- White (2001, Ch. 3)

References

Anderson, Brian D. O. and John B. Moore (2005). *Optimal Filtering.* Dover ed., unabridged republ. Dover Books on Engineering. Mineola, NY: Dover Publ. ISBN: 978-0-486-43938-9.

Anderson, Evan W. et al. (1996). "Chapter 4 Mechanics of Forming and Estimating Dynamic Linear Economies". In: *Handbook of Computational Economics.* Vol. 1. Elsevier, pp. 171–252. ISBN: 978-0-444-89857-9. DOI: 10.1016/S1574-0021(96)01006-4.

Baxter, Marianne and Robert G. King (1993). "Fiscal Policy in General Equilibrium". In: *The American Economic Review* 83.3, pp. 315–334. ISSN: 00028282.

McCandless, George T. (2008). *The ABCs of RBCs: An Introduction to Dynamic Macroeconomic Models.* Cambridge, MA: Harvard University Pres. ISBN: 978-0-674-02814-2.

Ploberger, Werner (2010). "Law(s) of Large Numbers". In: *Macroeconometrics and Time Series Analysis.* Ed. by Steven N. Durlauf and Lawrence E. Blume. London: Palgrave Macmillan UK, pp. 158–162. ISBN: 978-0-230-23885-5 978-0-230-28083-0. DOI: 10.1057/9780230280830_33.

Uribe, Martin and Stephanie Schmitt-Grohe (2017). *Open Economy Macroeconomics.* Princeton, NJ: Princeton University Press. ISBN: 978-0-691-15877-8.

White, Halbert (2001). *Asymptotic Theory for Econometricians.* Rev. ed. San Diego: Academic Press. ISBN: 978-0-12-746652-1.

A. Solutions

1 Solution to RBC model: preprocessing and steady-state in Dynare

1. The transversality condition for an infinite horizon dynamic optimization problem is the boundary condition determining a solution to the problem's first-order conditions together with the initial condition. The transversality condition requires the present value of the state variables (here K_t and A_t) to converge to zero as the planning horizon recedes towards infinity. The first-order and transversality conditions are sufficient to identify an optimum in a concave optimization problem. Given an optimal path, the necessity of the transversality condition reflects the impossibility of finding an alternative feasible path for which each state variable deviates from the optimum at each time and increases discounted utility. These conditions are implicit only, we don't enter them in a computer program. But implicitly we do consider them when we focus on unique and stable solutions or when we pick certain steady-state values.

2. Due to our assumptions , we will not have corner solutions and can neglect the non-negativity constraints. Due to the transversality condition and the concave optimization problem, we only need to focus on the first-order conditions. The Lagrangian for the household problem is

$$\begin{aligned}
\pounds = E_t \sum_{j=0}^{\infty} & \beta^j \left\{\gamma \log(C_{t+j}) + \psi \log(1 - L_{t+j})\right\} \\
& + \beta^j \lambda_{t+j} \left\{(W_{t+j} L_{t+j} + R_{t+j} K_{t-1+j} - C_{t+j} - I_{t+j})\right\} \\
& + \beta^j \mu_{t+j} \left\{((1-\delta) K_{t-1+j} + I_{t+j} - K_{t+j})\right\}
\end{aligned}$$

Note that the problem is not to choose $\{C_t, I_t, L_t, K_t\}_{t=0}^{\infty}$ all at once in an open-loop policy, but to choose these variables sequentially given the information at time t in a closed-loop policy, i.e. at period t decision rules for $\{C_t, I_t, L_t, K_t\}$ given the information set at period t; at period $t+1$ decision rules for $\{C_{t+1}, I_{t+1}, L_{t+1}, K_{t+1}\}$ given the information set at period $t+1$.

The first-order condition w.r.t. C_t is given by

$$\begin{aligned}
& \frac{\partial \pounds}{\partial C_t} = E_t \left(\gamma C_t^{-1} - \lambda_t\right) = 0 \\
& \Leftrightarrow \lambda_t = \gamma C_t^{-1} \qquad (I)
\end{aligned}$$

The first-order condition w.r.t. L_t is given by

$$\begin{aligned}
& \frac{\partial \pounds}{\partial L_t} = E_t \left(\frac{-\psi}{1 - L_t} + \lambda_t W_t\right) = 0 \\
& \Leftrightarrow \lambda_t W_t = \frac{\psi}{1 - L_t} \qquad (II)
\end{aligned}$$

The first-order condition w.r.t. I_t is given by

$$\begin{aligned}
& \frac{\partial \pounds}{\partial I_t} = E_t \left(-\lambda_t + \mu_t\right) = 0 \\
& \Leftrightarrow \lambda_t = \mu_t \qquad (III)
\end{aligned}$$

The first-order condition w.r.t. K_t is given by

$$\begin{aligned}
& \frac{\partial \pounds}{\partial K_t} = E_t(-\mu_t) + E_t \beta \left(\lambda_{t+1} R_{t+1} + \mu_{t+1}(1-\delta)\right) = 0 \\
& \Leftrightarrow \mu_t = E_t \beta (\mu_{t+1}(1-\delta) + \lambda_{t+1} R_{t+1}) \qquad (IV)
\end{aligned}$$

(I) and (III) in (IV) yields

$$\underbrace{\gamma C_t^{-1}}_{U_t^c} = \beta E_t \underbrace{\gamma C_{t+1}^{-1}}_{U_{t+1}^c} (1 - \delta + R_{t+1})$$

This is the Euler equation of intertemporal optimality. It reflects the trade-off between consumption and savings. If the household saves a (marginal) unit of consumption, she can consume the gross rate of return on capital, i.e. $(1-\delta+R_{t+1})$ units, in the following period. The marginal utility of consuming today is equal to U_t^c, whereas consuming tomorrow has expected utility $E_t(U_{t+1}^c)$. Discounting expected marginal utility with β the household must be indifferent between both choices in the optimum.

(I) in (II) yields:

$$W_t = -\frac{\frac{-\psi}{1-L_t}}{\gamma C_t^{-1}} \equiv -\frac{U_t^l}{U_t^c}$$

This equation reflects intratemporal optimality, particularly, the optimal choice for labor supply: the real wage must be equal to the marginal rate of substitution between labor and consumption.

3. First, we note that firms maximize expected profits, i.e. their horizon is the infinite time in principle. However, due to our assumptions, today's decisions don't impact tomorrow's decisions, so the maximization problem is actually a static problem as there are no forward-looking terms. That is, for every t, the objective is to maximize current profits:

$$DIV_t = A_t K_{t-1}^{\alpha} L_t^{1-\alpha} - W_t L_t - R_t K_{t-1}$$

The first-order condition w.r.t. L_t is given by

$$\begin{aligned}\frac{\partial DIV_t}{\partial L_t} &= (1-\alpha)A_t K_{t-1}^{\alpha} L_t^{-\alpha} - W_t = 0\\ \Leftrightarrow W_t &= (1-\alpha)A_t K_{t-1}^{\alpha} L_t^{-\alpha} = f_L = (1-\alpha)\frac{Y_t}{L_t}\end{aligned}$$

The real wage must be equal to the marginal product of labor. Due to the Cobb-Douglas production function it is a constant proportion $(1-\alpha)$ of the ratio of total output and labor. This is the labor demand function.

The first-order condition w.r.t. K_{t-1} is given by

$$\begin{aligned}\frac{\partial DIV_t}{\partial K_{t-1}} &= \alpha A_t K_{t-1}^{\alpha-1} L_t^{1-\alpha} - R_t = 0\\ \Leftrightarrow R_t &= \alpha A_t K_{t-1}^{\alpha-1} L_t^{1-\alpha} = f_K = \alpha\frac{Y_t}{K_{t-1}}\end{aligned}$$

The real interest rate must be equal to the marginal product of capital. Due to the Cobb-Douglas production function it is a constant proportion α of the ratio of total output and capital. This is the capital demand function.

4. First, consider the steady-state value of technology:

$$\log \bar{A} = \rho_A \log \bar{A} + 0 \Leftrightarrow \log \bar{A} = 0 \Leftrightarrow \bar{A} = 1$$

The Euler equation in steady-state becomes:

$$\begin{aligned}\bar{U}^C &= \beta \bar{U}^C(1-\delta+\bar{R})\\ \Leftrightarrow \bar{R} &= \frac{1}{\beta} + \delta - 1\end{aligned}$$

Now we will provide recursively closed-form expressions for all variables in relation to steady-state labor. That is the right-hand sides of the following equations are given in terms of parameters or previously computed expressions.

- The firms demand for capital in steady-state becomes

$$\bar{R} = \alpha \bar{A} \bar{K}^{\alpha-1} \bar{L}^{1-\alpha}$$
$$\Leftrightarrow \frac{\bar{K}}{\bar{L}} = \left(\frac{\alpha \bar{A}}{\bar{R}}\right)^{\frac{1}{1-\alpha}}$$

- The firms demand for labor in steady-state becomes:

$$W = (1-\alpha)\bar{A}\bar{K}^{\alpha}\bar{L}^{-\alpha} = (1-\alpha)\bar{A}\left(\frac{\bar{K}}{\bar{L}}\right)^{\alpha}$$

- The law of motion for capital in steady-state implies

$$\frac{\bar{I}}{\bar{L}} = \delta \frac{\bar{K}}{\bar{L}}$$

- The production function in steady-state becomes

$$\frac{\bar{Y}}{\bar{L}} = \bar{A}\left(\frac{\bar{K}}{\bar{L}}\right)^{\alpha}$$

- The clearing of the goods market in steady-state implies

$$\frac{\bar{C}}{\bar{L}} = \frac{\bar{Y}}{\bar{L}} - \frac{\bar{I}}{\bar{L}}$$

Now, we have expressions for all variables as a ratio to steady-state labor. Hence, once we compute L, we can revisit the above expressions to compute all values in closed-form. Due to the log-utility function, we can actually derive a closed-form expression for L:

$$\psi \frac{1}{1-\bar{L}} = \gamma \bar{C}^{-1} W$$
$$\Leftrightarrow \psi \frac{\bar{L}}{1-\bar{L}} = \gamma \left(\frac{\bar{C}}{\bar{L}}\right)^{-1} W$$
$$\Leftrightarrow \bar{L} = (1-\bar{L})\frac{\gamma}{\psi}\left(\frac{\bar{C}}{\bar{L}}\right)^{-1} W$$
$$\Leftrightarrow \bar{L} = \frac{\frac{\gamma}{\psi}\left(\frac{\bar{C}}{\bar{L}}\right)^{-1} W}{1+\frac{\gamma}{\psi}\left(\frac{\bar{C}}{\bar{L}}\right)^{-1} W}$$

Lastly, it is straightforward to compute the remaining steady-state values, i.e.

$$\bar{C} = \frac{\bar{C}}{\bar{L}}\bar{L}, \qquad \bar{I} = \frac{\bar{I}}{\bar{L}}\bar{L}, \qquad \bar{K} = \frac{\bar{K}}{\bar{L}}\bar{L}, \qquad \bar{Y} = \frac{\bar{Y}}{\bar{L}}\bar{L}$$

5. General hints: construct and parameterize the model such, that it corresponds to certain properties of the true economy. One often uses steady-state characteristics for choosing the parameters in accordance with observed data. For instance, long-run averages (wages, working-hours, interest rates, inflation, consumption-shares, government-spending-ratios, etc.) are used to fix steady-state values of the endogenous variables, which implies values for the parameters. You can also use micro-studies, however, one has to be careful about the aggregation and this is an on-going research agenda.

   We will focus on OECD countries and discuss one "possible" way to calibrate the model parameters (there are many other ways):

- $\boldsymbol{\alpha}$ productivity parameter of capital. Due to the Cobb Douglas production function thus should be equal to the proportion of capital income to total income of economy. So, one looks inside the national accounts for OECD countries and sets α to 1 minus the share of labor income over total income. For most OECD countries this implies a range of 0.25 to 0.35.
- $\boldsymbol{\beta}$ subjective intertemporal preference rate of households: it is the value of future utility in relation to present utility. Usually takes a value slightly less than unity, indicating that agents discount the future. For quarterly data, we typically set it around 0.99. A better way: fix this parameter by making use of the Euler equation in steady-state: $\beta = \frac{1}{\bar{R}+1-\delta}$ where $\bar{R} = \alpha\frac{\bar{Y}}{\bar{K}}$. Looking at OECD data one usually finds that average capital productivity $\bar{K}/\bar{Y}$ is in the range of 9 to 10.
- $\boldsymbol{\delta}$ depreciation rate of capital stock. For quarterly data the literature uses values in the range of 0.02 to 0.03. A better way: use steady-state implication that $\delta = \frac{\bar{I}}{\bar{K}} = \frac{\bar{I}/\bar{Y}}{\bar{K}/\bar{Y}}$. For OECD data one usually finds that average ratio of investment to output, $\bar{I}/\bar{Y}$, is around 0.25.
- $\boldsymbol{\gamma}$ **and** $\boldsymbol{\psi}$ individual's preference regarding consumption and leisure. Often a certain interpretation in terms of elasticities of substitutions is possible. Here we can make use of the First-Order-Conditions in steady-state, i.e.

$$\frac{\psi}{\gamma} = \bar{W}\frac{(1-\bar{L})}{\bar{C}} = (1-\alpha)\left(\frac{\bar{K}}{\bar{L}}\right)^{\alpha}\frac{(1-\bar{L})}{\bar{C}} = (1-\alpha)\left(\frac{\bar{K}}{\bar{L}}\right)^{\alpha}\frac{\frac{1}{\bar{L}}(1-\bar{L})}{\frac{\bar{C}}{\bar{L}}}$$

  and noting that $\bar{C}/\bar{L}$ as well as $\bar{K}/\bar{L}$ are given in terms of already calibrated parameters (see steady-state computations). Therefore, one possible way is to normalize one of the parameters to unity (e.g. $\gamma = 1$) and calibrate the other one in terms of steady-state ratios for which we would only require to calibrate steady-state hours worked $\bar{L}$. Note that labor time is normalized and usually corresponds to 8 hours a day, i.e. $\bar{L} = 1/3$.
- $\boldsymbol{\rho_A}$ **and** $\boldsymbol{\sigma_A}$ parameters of process for total factor productivity. These can be estimated based on a regression of the Solow Residual, i.e. production function residuals. ρ_A is mostly set above 0.9 to reflect persistence of the technological process and σ_A around 0.6 in the simple RBC model. Another way would be to try different values for σ_A and then try to match the shape of e.g. impulse-response-functions to corresponding (S)VAR models.

../progs/matlab/rbc_logutil_ss.m

```
function [SS,error_indicator] = rbc_logutil_ss(params)
error_indicator = 0; % initialize no error

% read-out parameters
alph  = params.alph;
betta = params.betta;
delt  = params.delt;
gam   = params.gam;
pssi  = params.pssi;
rhoA  = params.rhoA;

% compute steady-state
A = 1;
R = 1/betta+delt-1;
K_L = ((alph*A)/R)^(1/(1-alph));
if K_L <= 0
    error_indicator = 1;
end
W = (1-alph)*A*K_L^alph;
I_L = delt*K_L;
Y_L = A*K_L^alph;
```

```
C_L = Y_L - I_L;
if C_L <= 0
    error_indicator = 1;
end
L = gam/pssi*C_L^(-1)*W/(1+gam/pssi*C_L^(-1)*W); % closed-form expression for L

C = C_L*L;
I = I_L*L;
K = K_L*L;
Y = Y_L*L;
UC  = gam*C^(-1);
UL  = -pssi/(1-L);
fL  = (1-alph)*A*(K/L)^alph;
fK  = alph*A*(K/L)^(alph-1);

% write to output structure
SS.Y = Y;
SS.C = C;
SS.K = K;
SS.L = L;
SS.A = A;
SS.R = R;
SS.W = W;
SS.I = I;
SS.UC = UC;
SS.UL = UL;
SS.fL = fL;
SS.fK = fK;
end
```

You can try it out with the following parametrization (same as in the Dynare mod file):

../progs/matlab/rbc_logutil_ss_test.m

```
% this computes the steady-state of the rbc model

% calibration
params.alph = 0.35;
params.betta = 0.9901;
params.delt = 0.025;
params.gam = 1;
params.pssi = 1.7333;
params.rhoA = 0.9;

% compute steady-state
[SS,error_indicator] = rbc_logutil_ss(params);
if ~error_indicator
    disp(SS);
else
    error('steady-state could not be computed')
end
```

6. The mod file might look like this:

../progs/dynare/rbc_logutil.mod

```
%% Declare Variables and Parameters
var Y C K L A R W I UC UL fL fK;
varexo eps_A;
parameters alph betta delt gam pssi rhoA;

%% Calibration of parameters (simple)
alph = 0.35;
betta = 0.9901;
```

```
delt = 0.025;
gam = 1;
pssi = 1.7333;
rhoA = 0.9;

%% Calibration of parameters (advanced)
Abar  = 1;
K_o_Y = 10;    % avg capital productivity
I_o_Y = 0.25;  % avg investment ouput ratio
alph  = 0.35;  % cobb douglas
gam   = 1;     % normalize
Lbar  = 8/24;
rhoA  = 0.9;
delt = I_o_Y / K_o_Y;
betta = 1/(alph/K_o_Y+1-delt);
Rbar = 1/betta+delt-1;
K_o_L = ((alph*Abar)/Rbar)^(1/(1-alph));
Kbar = K_o_L*Lbar;
Ybar = Kbar/K_o_Y;
Ibar = delt*Kbar;
Wbar = (1-alph)*Abar*K_o_L^alph;
Cbar = Ybar - Ibar;
pssi = gam*(Cbar/Lbar)^(-1)*Wbar*(Lbar/(1-Lbar))^(-1);

%% Model Equations
model;
UC  = gam*C^(-1);
UL  = -pssi/(1-L);
fL  = (1-alph)*A*(K(-1)/L)^alph;
fK  = alph*A*(K(-1)/L)^(alph-1);

UC = betta*UC(+1)*(1-delt+R(+1));
W = - UL/UC;
W = fL;
R = fK;
Y = A*K(-1)^alph*L^(1-alph);
K = (1-delt)*K(-1) + I;
Y = C + I;
log(A) = rhoA*log(A(-1)) + eps_A;
end;

%% Steady State
steady_state_model;
A = 1;
R = 1/betta+delt-1;
K_L = ((alph*A)/R)^(1/(1-alph));
W = (1-alph)*A*K_L^alph;
I_L = delt*K_L;
Y_L = A*K_L^alph;
C_L = Y_L - I_L;
L = gam/pssi*C_L^(-1)*W/(1+gam/pssi*C_L^(-1)*W);
C = C_L*L;
I = I_L*L;
K = K_L*L;
Y = Y_L*L;
UC  = gam*C^(-1);
UL  = -pssi/(1-L);
fL  = (1-alph)*A*(K/L)^alph;
fK  = alph*A*(K/L)^(alph-1);
end;

steady;
```

Obviously, the results are the same.

8. a) For the first-order conditions of the household we know use

$$U_t^C = \gamma C_t^{-\eta_C}$$
$$U_t^L = -\psi(1-L_t)^{-\eta_L}$$

The steady-state for labor changes to

$$W\gamma\left(\frac{C}{L}\right)^{-\eta_C} = \psi(1-L)^{-\eta_L}L^{\eta_C}$$

This cannot be solved for L in closed-form. Rather, we need to condition on the values of the parameters and use an numerical optimizer to solve for L.

b) The function might look like this:

../progs/matlab/rbc_ss.m

```
%rbc_model_ss.m
function [SS,error_indicator] = rbc_ss(params,use_closed_form)
error_indicator = 0; % initialize no error
if nargin < 1
    use_closed_form = 1;
end

% read-out parameters
alph  = params.alph;
betta = params.betta;
delt  = params.delt;
gam   = params.gam;
pssi  = params.pssi;
rhoA  = params.rhoA;
etaC  = params.etaC;
etaL  = params.etaL;

% compute steady-state
A = 1;
R = 1/betta+delt-1;
K_L = ((alph*A)/R)^(1/(1-alph));
if K_L <= 0
    error_indicator = 1;
end
W = (1-alph)*A*K_L^alph;
I_L = delt*K_L;
Y_L = A*K_L^alph;
C_L = Y_L - I_L;
if C_L <= 0
    error_indicator = 1;
end
if (etaC == 1 && etaL == 1) && use_closed_form
    L = gam/pssi*C_L^(-1)*W/(1+gam/pssi*C_L^(-1)*W); % closed-form expression
     for L
else
    % No closed-form solution and we therefore use a fixed-point algorithm
    if error_indicator == 0
        options = optimset('Display','off','TolX',1e-12,'TolFun',1e-12);
        L0 = 1/3;
        [L,~,exitflag] = fsolve(@findL,L0,options);
        if exitflag <= 0
            error_indicator = 1;
        end
    else
        L = NaN;
    end
end
```

```
C = C_L*L;
I = I_L*L;
K = K_L*L;
Y = Y_L*L;
UC   = gam*C^(-1);
UL   = -pssi/(1-L);
fL   = (1-alph)*A*(K/L)^alph;
fK   = alph*A*(K/L)^(alph-1);

% write to output structure
SS.Y = Y;
SS.C = C;
SS.K = K;
SS.L = L;
SS.A = A;
SS.R = R;
SS.W = W;
SS.I = I;
SS.UC = UC;
SS.UL = UL;
SS.fL = fL;
SS.fK = fK;


%% Auxiliary function used in optimization
% note that some variables are not explicitly declared as input arguments but
    get their value from above,
% i.e. the scope of some variables spans multiple functions
function error = findL(L)
    error = W*gam*C_L^(-etaC) - pssi*(1-L)^(-etaL)*L^etaC;
end

end
```

You can try it out with the following parametrization (same as in the Dynare mod file):

../progs/matlab/rbc_ss_test.m

```
% this computes the steady-state of the rbc model

% calibration
params.alph = 0.35;
params.betta = 0.9901;
params.delt = 0.025;
params.gam = 1;
params.pssi = 1.7333;
params.rhoA = 0.9;
params.etaC = 2;
params.etaL = 1.5;

% compute steady-state
[SS,error_indicator] = rbc_ss(params);
if ~error_indicator
    disp(SS);
else
    error('steady-state could not be computed')
end

% double check
params.etaC = 1;
params.etaL = 1;

SS1 = rbc_ss(params,0);
SS2 = rbc_logutil_ss(params);
```

```
if ~isequal(SS1,SS2)
    error('steady-state is not computed correctly for etaC=etaL=1')
end
```

c) In Dynare we could use the following mod file:

../progs/dynare/rbc_ces.mod

```
%% Declare Variables and Parameters
var Y C K L A R W I UC UL fL fK;
varexo eps_A;
parameters alph betta delt gam pssi rhoA etaC etaL;

%% Calibration of parameters (simple)
alph = 0.35;
betta = 0.9901;
delt = 0.025;
gam = 1;
pssi = 1.7333;
rhoA = 0.9;
etaC = 2;
etaL = 1.5;


%% Model Equations
model;
UC  = gam*C^(-etaC);
UL  = -pssi*(1-L)^(-etaL);
fL  = (1-alph)*A*(K(-1)/L)^alph;
fK  = alph*A*(K(-1)/L)^(alph-1);

UC = betta*UC(+1)*(1-delt+R(+1));
W = - UL/UC;
W = fL;
R = fK;
Y = A*K(-1)^alph*L^(1-alph);
K = (1-delt)*K(-1) + I;
Y = C + I;
log(A) = rhoA*log(A(-1)) + eps_A;
end;

%% Steady State
steady_state_model;
A = 1;
R = 1/betta+delt-1;
K_L = ((alph*A)/R)^(1/(1-alph));
W = (1-alph)*A*K_L^alph;
I_L = delt*K_L;
Y_L = A*K_L^alph;
C_L = Y_L - I_L;
L0 = 1/3;
L = rbc_ces_helper_function(L0,pssi,etaL,etaC,gam,C_L,W);
C = C_L*L;
I = I_L*L;
K = K_L*L;
Y = Y_L*L;
UC  = gam*C^(-etaC);
UL  = -pssi*(1-L)^(-etaL);
fL  = (1-alph)*A*(K/L)^alph;
fK  = alph*A*(K/L)^(alph-1);
end;

steady;
```

and the following helper function:

../progs/dynare/rbc_ces_helper_function.m

```
function L = rbc_ces_helper_function(L0,pssi,etaL,etaC,gam,C_L,W)
    if etaC == 1 && etaL == 1
        L = gam/pssi*C_L^(-1)*W/(1+gam/pssi*C_L^(-1)*W);
    else
        options = optimset('Display','Final','TolX',1e-12,'TolFun',1e-12);
        L = fsolve(@(L) W*gam*C_L^(-etaC) - pssi*(1-L)^(-etaL)*L^etaC , L0,
    options);
    end
end
```

Obviously, the results are (numerically) the same.

2 Solution to Fiscal Policy in General Equilibrium: preprocessing and steady-state

../progs/dynare/Baxter_King.mod

```
%----------------------------------%
% declare variables and parameters %
%----------------------------------%
var
  y   % output
  c   % consumption
  iv  % private investment
  g   % government spending
  ivg % government investment
  lam % marginal utility of consumption
  k   % private capital stock
  kg  % public capital stock
  r   % real interest rate
  w   % real wage
  tau % net tax rate
  tr  % fiscal transfers
  z   % productivity process
  n   % labor
;

varexo
  e_z   % productivity shock
  e_g   % government spending shock
  e_ivg % government investment shock
  e_tau % tax rate shock
;

parameters
  BETA      % discount factor
  ETA       % public capital productivity
  ALPHA     % private capital productivity
  THETA_L   % utility weight for labor
  DELTA     % capital depreciation rate
  RHO_Z     % persistence parameter technology process
  RHO_G     % persistence parameter government spending process
  RHO_IVG   % persistence parameter government investment process
  RHO_TAU   % persistence parameter tax process
  Z_BAR     % target value of technology level
  G_BAR     % target value of government spending
  IVG_BAR   % target value of government investment
  TAU_BAR   % target value of net tax rate
;

%----------------------%
% calibrate parameters %
%----------------------%
%% parameter calibration using targeted steady-state values
Y_BAR   = 1;
G_BAR   = 0.2*Y_BAR;
IVG_BAR = 0.02*Y_BAR;
TR_BAR  = 0;
W_BAR   = 2;
N_BAR   = 1/3;
Z_BAR   = 1; %normalize

% public capital productivity ETA should be lower than capital share in production
    ALPHA
ALPHA = 1-W_BAR*N_BAR/Y_BAR; % from labor demand in steady-state
ETA = 0.3*ALPHA; % ETA is lower

% exogenous processes
RHO_Z   = 0.75;
```

```
RHO_G   = 0.75;
RHO_IVG = 0.75;
RHO_TAU = 0.75;

% Set some reasonable values for DELTA and BETA (see e.g. RBC model)
DELTA = 0.025;
KG_BAR = IVG_BAR/DELTA; % from public capital law of motion in steady-state
K_BAR = (Y_BAR/(Z_BAR*KG_BAR^ETA*N_BAR^(1-ALPHA)))^(1/ALPHA); % from production function
IV_BAR = K_BAR*DELTA;
R_BAR = ALPHA*Y_BAR/K_BAR; % from capital demand

TAU_BAR = (G_BAR + IVG_BAR + TR_BAR)/(W_BAR*N_BAR+R_BAR*K_BAR); % fiscal budget ins
    steady-state
BETA = 1/(1-DELTA+(1-TAU_BAR)*R_BAR); % from savings decision in steady-state
C_BAR = Y_BAR - IV_BAR - G_BAR - IVG_BAR; % from ressource constraint in steady-state

% labor utility weight from labor supply decision
THETA_L = (1-TAU_BAR)*W_BAR*(1-N_BAR)/C_BAR;

%-----------------%
% model equations %
%-----------------%
model;
[name='labor-leisure decision']
(1-tau)*w = THETA_L*c/(1-n);
[name='savings decision']
lam = BETA*lam(+1)*(1-DELTA + (1-tau(+1))*r(+1));
[name='marginal utility of consumption']
lam = 1/c;
[name='law of motion private capital stock']
k  = (1-DELTA)*k(-1)  + iv;
[name='law of motion public capital stock']
kg = (1-DELTA)*kg(-1) + ivg;
[name='production function']
y = z*kg(-1)^ETA*k(-1)^ALPHA*n^(1-ALPHA);
[name='labor demand']
w*n = (1-ALPHA)*y;
[name='private capital demand']
r*k(-1) = ALPHA*y;
[name='productivity process']
log(z/Z_BAR) = RHO_Z*log(z(-1)/Z_BAR) + e_z;
[name='government budget constraint']
g + ivg + tr = tau*(w*n + r*k(-1));
[name='fiscal rule: government spending']
g - G_BAR = RHO_G*(g(-1)-G_BAR) + e_g;
[name='fiscal rule: government investment']
ivg - IVG_BAR = RHO_IVG*(ivg(-1)-IVG_BAR) + e_ivg;
[name='fiscal rule: tax rate']
log(tau/TAU_BAR) = RHO_TAU*log(tau(-1)/TAU_BAR) + e_tau;
[name='market clearing']
y = c + iv + g + ivg;
end;


%--------------%
% computations %
%--------------%
initval;
y   = Y_BAR;
c   = C_BAR;
iv  = IV_BAR;
g   = G_BAR;
ivg = IVG_BAR;
lam = 1/C_BAR;
k   = K_BAR;
```

```
kg  = KG_BAR;
r   = R_BAR;
w   = W_BAR;
tau = TAU_BAR;
tr  = TR_BAR;
z   = Z_BAR;
n   = N_BAR;
end;
steady; % compute steady-state
```

3 Solution to Matrix Algebra

../progs/matlab/MatrixAlgebra_Eigenvalues.m

1.
```
% =========================================================================
% MatrixAlgebra_Eigenvalues.m
% =========================================================================
% Compute Eigenvalues of a Matrix
% =========================================================================
% Willi Mutschler, November 16, 2021
% willi@mutschler.eu
% =========================================================================
clearvars; clc; close all;

A = [0.5,    0,    0;
     0.1,  0.1,  0.3;
     0,    0.2,  0.3];

EV_A = eig(A);
disp(abs(EV_A)<1);
```

In the univariate AR(1) model we would check whether the autocorrelation coefficient is between -1 and 1, i.e. $|A| < 1$, such that $\sum_{j=0}^{\infty}(AL)^j = 1/(1-AL)$ where L is the lag operator. In the multivariate case, we want the same thing, i.e. $\sum_{j=0}^{\infty}(AL)^j = (1-AL)^{-1}$. Note that A is a square matrix and taking the power of a matrix is not a trivial task. One convenient way to do so, however, is to consider an eigenvalue decomposition (if it exists): $A = Q\Lambda Q^{-1}$, where Q is the square matrix whose columns contain the eigenvectors q_i corresponding to the eigenvalues λ_i found on the diagonal of $\Lambda = [\lambda_i]_{ii}$. Moreover, note that Λ is a diagonal matrix and Q is an orthogonal matrix $Q^{-1} = Q'$. Using this decomposition one can show that it is very easy to compute for instance a

- matrix inverse $A^{-1} = Q\Lambda^{-1}Q^{-1}$ where the inverse of $[\Lambda^{-1}]_{ii} = 1/\lambda_i$ is very easy to calculate as it is a diagonal matrix
- matrix powers: $A^2 = (Q\Lambda Q^{-1})(Q\Lambda Q^{-1}) = Q\Lambda(Q^{-1}Q)\Lambda Q^{-1} = Q\Lambda^2 Q^{-1}$ or more generally: $A^j = Q\Lambda^j Q^{-1}$.

So, for $\sum_{j=0}^{\infty}(AL)^j = (1-AL)^{-1}$ we need that $\lim_{j\to\infty}\Lambda^j = 0$. As this is a diagonal matrix, this simplifies to looking at each eigenvalue, whether it is between -1 and 1: $|\lambda_i| < 1$. In other words, for VAR(1) systems $y_t = c + Ay_{t-1} + u_t$ we need to check whether the eigenvalues are inside the unit circle, then the VAR(1) model is said both stable and covariance-stationary.

2. Example for vectorization and Kronecker product:

$$\underbrace{vec\begin{pmatrix}1&3&2\\1&0&0\\1&2&2\end{pmatrix}}_{3\times3} = \underbrace{\begin{pmatrix}1\\1\\1\\3\\0\\2\\2\\0\\2\end{pmatrix}}_{9\times1}, \qquad \underbrace{\begin{pmatrix}1&3&2\\1&0&0\\1&2&2\end{pmatrix}}_{3\times3} \otimes \underbrace{\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix}}_{3\times2} = \underbrace{\begin{pmatrix} 1\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix} & 3\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix} & 2\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix}\\ 1\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix} & 0\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix} & 0\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix}\\ 1\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix} & 2\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix} & 2\cdot\begin{pmatrix}0&5\\5&0\\1&1\end{pmatrix}\end{pmatrix}}_{9\times6}$$

Using this definition, we can show that $vec(DEF) = (F' \otimes D)vec(E)$ using e.g. a symbolic toolbox:

../progs/matlab/MatrixAlgebra_KroneckerFormula.m

```
% =========================================================================
```

```
% MatrixAlgebra_KroneckerFormula.m
% ======================================================================
% Show that vec(D*E*F) = kron(F',D)*vec(E) using MATLAB's symbolic toolbox
% ======================================================================
% Willi Mutschler, November 16, 2021
% willi@mutschler.eu
% ======================================================================
clearvars; clc; close all;

%% Some basics on the symbolic toolbox
syms x % create symbolic variable
% note that matlab does not simplify or expand by default! example:
f1 = (x - 2)^2;
f2 = x^2 - 4*x + 4;
expand(f1)
simplify(f2)
isequal(f1,f2)
isequal(expand(f1),f2)
isequal(f1,simplify(f2)) % note that simplify usually takes longer than expand

%% Show that vec(D*E*F) = kron(F',D)*vec(E)
dim = randi([1 10],1,4); % generate 4 random integers between 1 and 10 as
    dimensions
% create symbolic matrices
D = sym('d',[dim(1) dim(2)]);
E = sym('e',[dim(2) dim(3)]);
F = sym('f',[dim(3) dim(4)]);
DEF = D*E*F; % check whether matrix product is defined
vecDEF = DEF(:); %vectorization

% correct: compare expanded symbolic expressions
if isequal(expand(vecDEF),expand(kron(transpose(F),D)*E(:)))
    fprintf('Expanded expressions are identical\n');
else
    error('Expanded expressions are not identical');
end
```

Of course you can do this on paper as well:

$$
\begin{aligned}
DEF &= D\begin{pmatrix} e_1 & e_2 & \dots & e_p \end{pmatrix}\begin{pmatrix} f_{11} & f_{12} & \dots & f_{1k} \\ f_{21} & f_{22} & \dots & f_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ f_{p1} & f_{p2} & \dots & f_{pk} \end{pmatrix} \\
&= D\underbrace{\begin{pmatrix} e_1 f_{11} + e_2 f_{21} + \dots + e_p f_{p1}, & e_1 f_{12} + e_2 f_{22} + \dots + e_p f_{p2}, & \dots, & e_1 f_{1k} + e_2 f_{2k} + \dots + e_p f_{pk} \end{pmatrix}}_{n\times k}
\end{aligned}
$$

$$
\begin{aligned}
vec(DEF) &= \begin{pmatrix} f_{11}De_1 + f_{21}De_2 + \dots + f_{p1}De_p \\ f_{12}De_1 + f_{22}De_2 + \dots + f_{p2}De_p \\ \vdots \\ f_{1k}De_1 + f_{2k}De_2 + \dots + f_{pk}De_p \end{pmatrix} = \begin{pmatrix} f_{11}D & f_{21}D & \dots & f_{p1}D \\ f_{12}D & f_{22}D & \dots & f_{p2}D \\ \vdots & \vdots & \vdots & \vdots \\ f_{1k}D & f_{2k}D & \dots & f_{pk}D \end{pmatrix}\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{pmatrix} \\
&= (F' \otimes D)\, vec(E)
\end{aligned}
$$

3. An orthogonal matrix is characterized by $R' = R^{-1}$ and therefore $R'R = RR' = I$. Here:

$$
R'R = \begin{pmatrix} (\cos(\phi))^2 + (\sin(\phi))^2 & -\cos(\phi)\sin(\phi) + \sin(\phi)\cos(\phi) \\ -\sin(\phi)\cos(\phi) + \cos(\phi)\sin(\phi) & (\sin(\phi))^2 + (\cos(\phi))^2 \end{pmatrix}
$$

with $(\cos(\phi))^2 + (\sin(\phi))^2 = 1$ (so-called trigonometric Pythagoras).

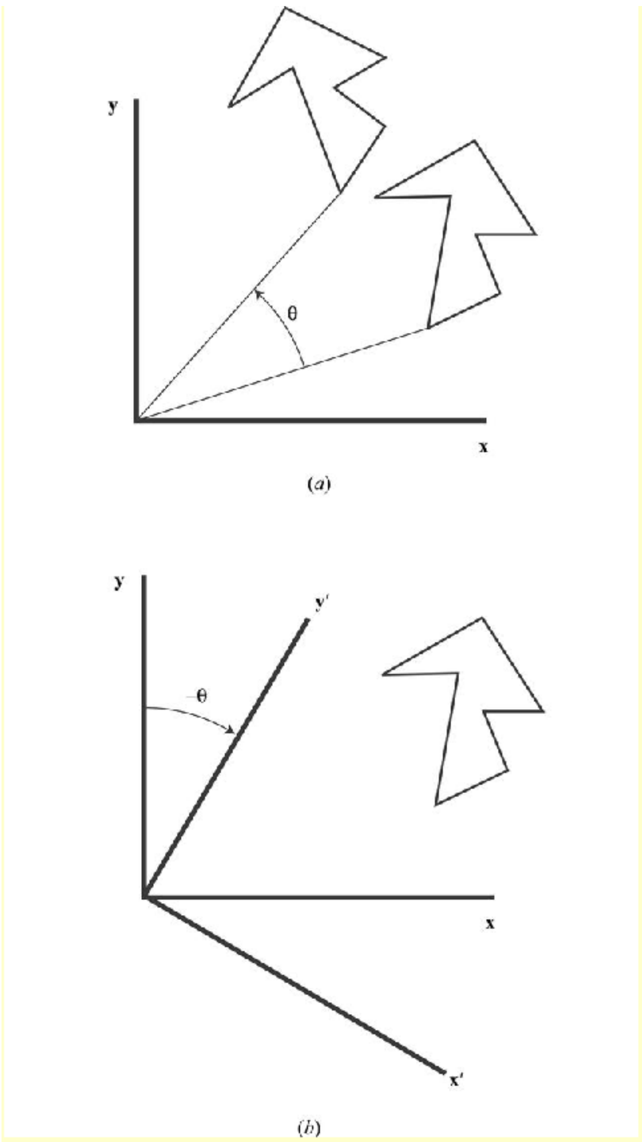
../progs/matlab/MatrixAlgebra_Rotation.m

```matlab
% ========================================================================
% MatrixAlgebra_Rotation.m
% ========================================================================
% Show orthogonality of 2-dimensional rotation matrix using MATLAB's
% symbolic toolbox
% ========================================================================
% Willi Mutschler, November 16, 2021
% willi@mutschler.eu
% ========================================================================
clearvars; clc; close all;

theta = sym('theta');
R = [cos(theta), -sin(theta);
     sin(theta), cos(theta)];

simplify(transpose(R)*R)
simplify(R*transpose(R))
Rinv = R\eye(size(R,1));
simplify(transpose(R) - Rinv)
```

Why rotation matrix? A rotation matrix rotates vectors or objects in the Euclidian space without stretching or shrinking it.



In this example the matrix R rotates the vector counter-clockwise given angle ϕ. An active rotation means that the vector is multiplied by the rotation matrix and this rotates the vector counterclockwise $x' = Rx$. A passive rotation means that the coordinate system is rotated and therefore the vector is also rotated: $x' = R^{-1}x$. Later on we will need rotation matrices for identification of structural shocks!

../progs/matlab/MatrixAlgebra_Cholesky.m

4.
```matlab
% ======================================================================
% MatrixAlgebra_Cholesky.m
% ======================================================================
% Decompose a covariance matrix using the Cholesky decomposition, i.e.
% SIGu = W*SIGe*W'
% ======================================================================
% Willi Mutschler, November 16, 2021
% willi@mutschler.eu
% ======================================================================
clearvars; clc; close all;

SIGu = [2.25 0 0; 0 1 0.5; 0 0.5 0.74];
P = chol(SIGu,'lower');
% Note that P = W*SIGe^(1/2)
SIGe_sqrt = diag(P);
SIGe = diag(SIGe_sqrt.^2);
% Find W which is solution to equation W*SIGe^(1/2) = P
% - A\B (mldivide) solves A*x = B
% - A/B (mrdivide) solves x*B = A <-- we want this to get W
W = P/diag(SIGe_sqrt);
isequal(W*SIGe*W',SIGu)
```

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{pmatrix}}_{W} \underbrace{\begin{pmatrix} 2.25 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.49 \end{pmatrix}}_{\Sigma_\varepsilon} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}}_{W'} = \underbrace{\begin{pmatrix} 2.25 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 0.74 \end{pmatrix}}_{\Sigma}$$

5. Solving this equation can be done either analytically or using an algorithm:

   a) Analytically:

$$\begin{aligned} vec(\Sigma_y) &= vec(A\Sigma_y A') + vec(\Sigma_u) = (A \otimes A)vec(\Sigma_y) + vec(\Sigma_u) \\ (I - A \otimes A)vec(\Sigma_y) &= vec(\Sigma_u) \\ vec(\Sigma_y) &= (I - A \otimes A)^{-1} vec(\Sigma_u) \end{aligned}$$

   b) Doubling algorithm:

../progs/matlab/dlyapdoubling.m

```matlab
% ======================================================================
% dlyapdoubling.m
% ======================================================================

function SIGy = dlyapdoubling(A,SIGu)
% ======================================================================
% Solves the Lyapunov equation SIGy = A*SIGy*A' + SIGu using the doubling
% algorithm
% =====================================================================
% SIGy = dlyapdoubling(A,SIGu)
% ---------------------------------------------------------------------
% INPUT
%   - A     : square matrix [n x n] (usually autoregressive or state space
    matrix)
%   - SIGu  : square matrix (n x n] (usually covariance matrix)
```

```
% --------------------------------------------------------------------
% OUTPUT
%  - SIGy: square matrix (usually covariance matrix) [n x n] that solves
%  the Lyapunov equation
% ====================================================================
% Willi Mutschler, October 30, 2021
% willi@mutschler.eu
% ====================================================================

max_iter    = 500;
A_old       = A;
SIGu_old    = SIGu;
SIGy_old    = eye(size(A));
difference = .1;
index1      = 1;
tol         = 1e-25;
while (difference > tol) && (index1 < max_iter)
    SIGy        = A_old*SIGy_old*transpose(A_old) + SIGu_old;
    difference  = max(abs(SIGy(:)-SIGy_old(:)));
    SIGu_old    = A_old*SIGu_old*transpose(A_old) + SIGu_old;
    A_old       = A_old*A_old;
    SIGy_old    = SIGy;
    index1      = index1 + 1;
end

end %function end
```

The basic idea of the doubling algorithm is to start at some $\Sigma_{y,0}$ and find new values for $\Sigma_{y,i+1}$ using the equation $A\Sigma_{y,i}A' + \Sigma_u$ until the difference $\Sigma_{y,i+1} - \Sigma_{y,i}$ becomes very small or a certain number of iterations is reached. The doubling algorithm, however, allows one to pass in one iteration from $\Sigma_{y,i}$ to $\Sigma_{y,2i}$ rather than $\Sigma_{y,i+1}$, provided that one updates three other matrices. There are also other (generalized) algorithms to solve such matrix Lyapunov (or Sylvester) equations.

c) Comparison:

../progs/matlab/MatrixAlgebra_Lyapunov.m

```
% ====================================================================
% MatrixAlgebra_Lyapunov.m
% ======================================================================
% Compares different ways to compute solution of Lyapunov equation
% SIGy = A*SIGy*A' + SIGu, i.e. analytically using the Kronecker formula
% and the doubling algorithm
% ====================================================================
% Willi Mutschler, November 16, 2021
% willi@mutschler.eu
% ======================================================================
clearvars; clc; close all;

A = [0.5,    0,    0;
     0.1,  0.1,  0.3;
     0,    0.2,  0.3];
SIGu = [2.25 0 0; 0 1 0.5; 0 0.5 0.74];

tic
vecSIGy = (eye(size(A,1)^2)-kron(A,A)) \ SIGu(:);
SIGy_kron     = reshape(vecSIGy,size(A));
toc

tic
SIGy_dlyap = dlyapdoubling(A,SIGu);
toc
fprintf('The maximum absolute difference of entries is %d\n',max(abs(
    SIGy_kron-SIGy_dlyap)))
```

```
% re-run to see that dlyapdoubling becomes faster (just-in-time compilation)
```

4 Solution to Law Of Large Numbers

1. In probability theory, the law of large numbers (LLN) is a theorem that describes the result of performing the same experiment a large number of times (repetitions, trials, experiments, or iterations). According to the LLN, the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed. The laws of large numbers are the cornerstones of asymptotic theory. In this exercise, the LLN is about determining what happens to $\overline{Y}_T$ as $T \rightarrow \infty$ (note that $\overline{Y}_T$ is a random variable). The LLN states that this series converges to the expected value μ. More precisely, the strong LLN implies that at the limit, we can exactly determine μ. The weak LLN implies that we can only approximately determine μ, even though we can make the approximation very very very close to μ. This implies:

   - Strong LLN means almost-sure convergence: At some point adding more observation does not matter at all for the average, it would be exactly equal to the expected value. That is, the sequence $\overline{Y}_1, \overline{Y}_2, \ldots$ of random variables converges **almost surely** to a variable μ, if

     $$Pr\left(\left\{\lim_{T\rightarrow\infty} \overline{Y}_T = \mu\right\}\right) = 1$$

     or simply:

     $$\overline{Y}_T \overset{a.s.}{\rightarrow} \mu$$

     This definition of convergence is not very important in econometrics.

   - Weak LLN means that the sample mean converges in probability to the population mean. That is, the sequence $\overline{Y}_1, \overline{Y}_2, \ldots$ of random variables converges **in probability** to a variable μ, if

     $$\lim_{T\rightarrow\infty} Pr\left(|Y_T - \mu| < \delta\right) = 1$$

     As $T \rightarrow \infty$, the probability is approaching 1 very closely, but it is not actually 1. In other words, the probability that the average is "far" (more than an arbitrary small number δ) from the expectation μ is zero. More compactly notation:

     $$\begin{aligned} \overline{Y}_T &\overset{p}{\rightarrow} \mu \\ plim\, \overline{Y}_T &= \mu \end{aligned}$$

     This definition of convergence is very important in econometrics. There exist different theorems that establish necessary and sufficient conditions depending on the process at study. In Quantitative Macroeconomics, we are mainly concerned with identically distributed process that are either independent of each other or homogenously dependent, i.e. autoregressive AR or VAR processes (aka homogeneous Markov processes).

2. Here is an extended illustration for several distributions:

../progs/matlab/LawOfLargeNumbers.m

```
% ======================================================================
% LawOfLargeNumbers.m
% ======================================================================
% Illustration of the weak law of large numbers for several distributions:
% normal,uniform,geometric,student's t (finite and infinite variance),gamma.
% ======================================================================
% Willi Mutschler, October 27, 2021
% willi@mutschler.eu
% ======================================================================
%% Housekeeping
clearvars; clc;close all;

%% Initializations
```

```
T = 10000;                                       % maximum horizon of periods
z = nan(1,T); sig_z = 2; mu_z = 10;              % normal distribution
u = nan(1,T); a = 2; b = 4; mu_u = (a+b)/2;      % uniform distribution
ge = nan(1,T); p = 0.2; mu_ge = (1-p)/p;         % geometric distribution
ga = nan(1,T); k=2; thet=2; mu_ga=k*thet;        % gamma distribution
st1 = nan(1,T); nu1 = 3; mu_st1 = 0;             % student t with finite variance
st2 = nan(1,T); nu2 = 2; mu_st2 = 0;             % student t with infinite variance

%% Draw random variables
ZZ = mu_z + sig_z.*randn(1,T); % normal distribution
UU = a + (b-a).*rand(1,T);     % uniform distribution
GeGe = geornd(p,1,T);          % geometric distribution
GaGa = gamrnd(k,thet,1,T);     % gamma distribution
ST1 = trnd(nu1,1,T);           % student t with finite variance
ST2 = trnd(nu2,1,T);           % student t with infinite variance

%% Compute and store mean for growing sample sizes
wait = waitbar(0,'Please wait...'); % open waitbar
for t = 1:T
    % get random numbers with growing sample size
    Zt = ZZ(1:t);                  % normal distribution
    Ut = UU(1:t);                  % uniform distribution
    Get = GeGe(1:t);               % geometric distribution
    Gat = GaGa(1:t);               % gamma distribution
    ST1t = ST1(1:t);               % student t with finite variance
    ST2t = ST2(1:t);               % student t with infinite variance

    % Compute and store averages
    z(t) = mean(Zt);
    u(t) = mean(Ut);
    ge(t) = mean(Get);
    ga(t) = mean(Gat);
    st1(t) = mean(ST1t);
    st2(t) = mean(ST2t);

    waitbar(t/T); % update waitbar
end
close(wait); % close waitbar

%% Create plot for different distributions
figure('name','Law of Large Numbers for different distributions');
subplot(2,3,1);
    plot(z);
    line(0:T,repmat(mu_z,1,T+1),'linestyle','--','color','black');
    title('Normal');
subplot(2,3,2);
    plot(u);
    line(0:T,repmat(mu_u,1,T+1),'linestyle','--','color','black');
    title('Uniform');
subplot(2,3,3);
    plot(ge);
    line(0:T,repmat(mu_ge,1,T+1),'linestyle','--','color','black');
    title('Geometric');
subplot(2,3,4);
    plot(ga);
    line(0:T,repmat(mu_ga,1,T+1),'linestyle','--','color','black');
    title('Gamma');
subplot(2,3,5);
    plot(st1);
    line(0:T,repmat(mu_st1,1,T+1),'linestyle','--','color','black');
    title('Student''s t finite variance');
subplot(2,3,6);
    plot(st2);
    line(0:T,repmat(mu_st2,1,T+1),'linestyle','--','color','black');
    title('Student''s t infinite variance');
```

Note that for the t-distribution with infinite variance the weak LLN does not apply.

3. Here is an extended illustration for different error term distributions:

../progs/matlab/LawOfLargeNumbersAR1.m

```matlab
% ======================================================================
% LawOfLargeNumbersAR1.m
% ======================================================================
% Illustration of the weak law of large numbers for the AR(1) process based
% on different error term distributions. Distributions considered:
% normal,uniform,geometric,student's t (finite and infinite variance),gamma.
% ======================================================================
% Willi Mutschler, October 27, 2021
% willi@mutschler.eu
% ======================================================================
%% Housekeeping
clearvars; clc;close all;

%% Initializations
T = 100000;                   % maximum horizon of periods
sig_z = 2; mu_z = 10;         % parameters for normal distribution
a = 2; b = 4; mu_u = (a+b)/2; % parameters for uniform distribution
p = 0.2; mu_ge = (1-p)/p;     % parameters for geometric distribution
k=2; thet=2; mu_ga=k*thet;    % parameters for gamma distribution
nu1 = 5; mu_st1 = 0;          % parameters for student t with finite variance
nu2 = 2; mu_st2 = 0;          % parameters for student t with infinite variance
phi=0.8; mu_y = 0;            % parameters for stable AR(1) process

Y = nan(T,6);                 % initialize output vector: sample size in rows,
    distributions in columns
%% Draw random variables
ZZ = mu_z + sig_z.*randn(1,T); % normal distribution
UU = a + (b-a).*rand(1,T);     % uniform distribution
GeGe = geornd(p,1,T);          % geometric distribution
GaGa = gamrnd(k,thet,1,T);     % gamma distribution
ST1 = trnd(nu1,1,T);           % student t with finite variance
ST2 = trnd(nu2,1,T);           % student t with infinite variance

%% Compute and store mean for growing sample sizes
wait = waitbar(0,'Please wait...'); % open waitbar
for t = 1:T
    Yt = nan(t,6);  % initialize matrix for simulated data of sample size t in the
     rows
                    % different processes due to the 6 different error term
    distributions are in the columns
    Yt(1,:) = mu_y; % initialize the first observation with the unconditional mean

    if t>1
        for tt=2:t
            % Note that we demean the errors
            Yt(tt,1) = phi*Yt(tt-1,1) + (ZZ(tt)-mu_z);     % normal distribution
    with mean zero
            Yt(tt,2) = phi*Yt(tt-1,2) + (UU(tt)-mu_u);     % uniform
            Yt(tt,3) = phi*Yt(tt-1,3) + (GeGe(tt)-mu_ge); % geometric
            Yt(tt,4) = phi*Yt(tt-1,4) + (GaGa(tt)-mu_ga); % gamma
            Yt(tt,5) = phi*Yt(tt-1,5) + (ST1(tt)-mu_st1); % finite variance
    student t
            Yt(tt,6) = phi*Yt(tt-1,6) + (ST2(tt)-mu_st2); % infinite variance
    student t
        end
    end
    % Compute and store averages
    Y(t,:) = mean(Yt,1);
    waitbar(t/T); % update waitbar
end
```

```
close(wait); % close waitbar

%% Create plot for AR(1) with different distributions
figure('name','Law of Large Numbers for stable AR(1)');
subplot(2,3,1);
    plot(Y(:,1));
    line(0:T,repmat(mu_y,1,T+1),'linestyle','--','color','black');
    title('Normal errors');
subplot(2,3,2);
    plot(Y(:,2));
    line(0:T,repmat(mu_y,1,T+1),'linestyle','--','color','black');
    title('Uniform errors');
subplot(2,3,3);
    plot(Y(:,3));
    line(0:T,repmat(mu_y,1,T+1),'linestyle','--','color','black');
    title('Geometric errors');
subplot(2,3,4);
    plot(Y(:,4));
    line(0:T,repmat(mu_y,1,T+1),'linestyle','--','color','black');
    title('Gamma errors');
subplot(2,3,5);
    plot(Y(:,5));
    line(0:T,repmat(mu_y,1,T+1),'linestyle','--','color','black');
    title('Student''s t finite variance errors');
subplot(2,3,6);
    plot(Y(:,6));
    line(0:T,repmat(mu_y,1,T+1),'linestyle','--','color','black');
    title('Student''s t infinite variance errors');
```

Note that we need to make sure that $E[\varepsilon_t] = 0$ when we simulate data. We see that the weak law of large numbers holds under weaker conditions than iid. For instance, one can show that for the stationary AR(1), necessary and sufficient conditions are: $Var[y_t] < \infty$ and $|\gamma(k)| \to 0$ as $k \to \infty$. This does not hold for all distributions considered in the code. Particularly, the student's t distribution with 2 degrees of freedom does not have a finite variance.