Economic Growth: Theory and numerical solution methods

A.Novales, E.Fernandez, J.Ruiz

List of EXCEL files and MATLAB programs
(ordered by sections)

Chapter 2. The neoclassical growth model under a constant savings rate

- ***Solow_deterministic.xls***. Section 2.5.3
  A single realization for the constant savings rate, deterministic growth model, is analyzed.
  - *Discrete spreadsheet:* The solution to the model is computed from the exact solution as well as from linear and quadratic approximations. A numerical comparison between them is performed.
  - *Increasing time path* spreadsheet: the case of an economy converging to steady state from below
  - *Decreasing time path* spreadsheet: the case of an economy converging to steady state from above

- ***Change_steadystate.m***. Section 2.5.3
  Characterization of steady-state effects from changes in structural parameters other than the savings rate

- ***Change_savings.xls***. Section 2.5.4
  Short- and long-run effects of a permanent increase in the value of the savings rate. A large and a small increase in savings rate are considered in different spreadsheets.
  - *C-increases* (1) and (2) spreadsheets: it presents the case of an economy where steady-state consumption is higher after the increase in savings.
  - *C-decreases* (1) and (2) spreadsheets: it presents the case of an economy where steady-state consumption is lower after the increase in savings

- ***Change_savings.m***. Section 2.5.4
  It performs the same exercise as ***Change_savings.xls***

- ***Dynamic_inefficiency.xls***. Section 2.5.5
  Characterization of dynamically efficient and inefficient steady-states

- ***Solow_dynamic.m***. Section 2.5.5
  The program performs the same exercise as ***Dynamic_inefficiency.xls***.

- ***Solow_stochastic.xls***. Section 2.6.1
  An approximate and an exact solution to the stochastic version of the constant savings rate economy. Four different realizations are displayed in as many spreadsheets

- ***Solow_stochastic.m***. Sections 2.5.3 and 2.6.1
  The program performs the same exercise as ***Solow_stochastic.xls***. This program can also be used to reproduce the exercise in *Solow_deterministic.x*ls file by setting a variance parameter to zero.

Chapter 3. Optimal Growth in continuous time

- ***CK_continuous time.xls***. Section 3.1.6
  - ***Steady-state*** spreadsheet: steady-state levels for different parameterizations
  - ***Speed of convergence*** spreadsheet: Speed of convergence to steady state, as a function of structural parameter values
  - ***Convergence. Risk aversion*** and ***Risk aversion*** spreadsheets: Convergence trajectories for different values of the intertemporal elasticity of substitution of consumption. It shows the potential pitfalls in using discrete time observations from the solution to the continuous time economy
  - ***Change in output share of k*** spreadsheet: Steady-state effects of a change in the value of the output elasticity of capital

- ***CK_c_steady state.m.*** Section 3.1.6
  A program to analyze the sensitivity of the steady state levels of endogenous variables in the continuous time version of the Cass Koopmans model with respect to values of structural parameters.

- ***CK_c_transition.m.*** Section 3.1.6
  Impulse response functions in the continuous time version of the Cass Koopmans model. This program calls function ***ss_ck_c.m.***, which lays out first order conditions for the planner's problem and computes the transition matrix in the log-linear approximation to the model. Impulse response functions are obtained by taking an initial condition on the stock of capital different form the steady-state level.

- ***CK_c_change structural parameters.m.*** Section 3.1.6
  Short- and long-run analysis of effects of a change in a structural parameter in the continuous time Cass-Koopmans economy. The program is written to consider changes in the output elasticity of capital, but changes in any other parameter can be considered alternatively. Like ***CK_impulse responses .m***, this program calls function ***ss_ck_c.m***.

- ***CK_Taxes_deterministic.xls***. Section 3.8.2
  Solution to the deterministic planner's problem under consumption and income taxes. It shows the need to appropriately use stability conditions

Chapter 4. Optimal Growth in discrete time

- ***CK_d_transition.m.*** Section 3.8.2
  This program computes the transition in the discrete time version of the Cass-Koopmans model without uncertainty using different treatments of stability. This program calls function ***ss_ck_d.m,*** which lays out first order conditions for the discrete-time version of the Cass-Koopmans economy. These conditions can be used either to compute steady-state levels of the endogenous variables, as well as the transition matrix for the linear approximation to the model.

- ***CK_solution_changes in tax.xls.*** Section 3.8.3
  - ***Steady state*** spreadsheet: Long-run effects of different fiscal policy experiments
  - ***Transition*** spreadsheet: Short- and long-run effects of some fiscal policy experiments

- ***CK_d_long run tax changes.m.*** Section 3.8.3
  A program to analyze long-run (steady-state) effects of changes in either the consumption tax rate, the income tax rate or both in the deterministic, discrete time

Cass-Koopmans economy. The analysis can be done either keeping revenues constant, or maintaining a constant ratio of revenues to output. This program calls the following functions: *i*) ***ss_ck_d.m***, which solves for the steady-state, *ii*) ***ss_ck_d1.m***, which solves for the steady-state keeping revenues constant (in this case, the consumption tax becomes an endogenous variable, which is solved for so that revenues remain constant for any income tax rate), *iii*) ***ss_ck_d2.m***, which solves for the steady-state keeping constant the ratio of revenues to total income (in this case, the consumption tax is an endogenous variable which is solved for so that the revenues to output ratio is the same for any income tax rate).

- ***CK_d_long_short run tax changes.m.*** Section 3.8.3
  A program to analyze short- and long-run effects in the deterministic, discrete-time Cass-Koopmans economy of changes in either the consumption tax rate, the income tax rate, or both, while keeping constant either total revenues, or the ratio of revenues to income. This program calls the same functions as ***CK_d_long_run tax changes.m***.

Chapter 5. Numerical solution methods

- ***Simple_models.xls.*** Section 5.2.3
  A sample realization form the analytical solution for each of the two economies considered in this section

- ***Simple_planner_problemxls.*** Sections 5.3.1 and 5.3.4
  A single realization from the planner's problem without taxes, using the linear quadratic approximation as well as Uhlig's undetermined coefficients approach.

- ***lq.m.*** Section 5.3.1
  This program computes a single realization from the discrete-time version of the Cass-Koopmans growth model under uncertainty, using the linear-quadratic approximation solution method. The program calls function ***util.m***, which lays out the objective function so that the numerical derivatives needed to implement the solution method can be computed

- ***CK_solution_BK.xls.*** Section 5.3.3 and 5.4.2
  A single realization from the solution to the planner's problem with taxes using Blanchard and Kahn's approach

- ***Blanchard Kahn.m.*** Section 5.3.3
  The Blanchard-Kahn method is used to compute a numerical solution to the discrete-time version of the Cass-Koopmans economy subject to productivity shocks. A single realization is obtained and graphs for the simulated series for the main variables in the model are displayed.

- ***uhlig.m.*** Section 5.3.4
  This program computes a single realization from the discrete-time version of the Cass-Koopmans growth model under uncertainty using Uhlig's solution method, based on a log-linear approximation. The program calls function *ss_lsims.m*, as in ***LSIMS.m***.

- ***CK_Stochastic.xls.*** Section 5.3.5
  Numerical solutions for the planner's problem without taxes using the eigenvalue-eigenvector decomposition approach

- ***CK_Stochastic_taxes.xls.*** Section 5.3.5

Numerical solutions for the planner problem under consumption and income taxes, using the eigenvalue-eigenvector decomposition approach. A single realization from the stochastic economy is obtained for alternative values of the consumption and income tax rates, maintaining the same values for structural parameters.

- ***CK_Stochastic_taxes_structural_parameters.xls.*** Section 5.3.5
  Numerical solutions for the planner's problem using the eigenvalue-eigenvector decomposition approach. A single realization from the stochastic economy is obtained for alternative values of the vector of structural parameters, maintaining the same consumption and income tax rates.

- ***SIMS.m.*** Section 5.3.5
  This program computes a single realization of the discrete-time version of the Cass-Koopmans growth model under uncertainty, using Sims's eigenvalue-eigenvector decomposition method and a linear approximation. This program calls function ***ss_sims.m***, to solve for the steady-state as well as to compute the transition matrix in the linear approximation to the first order conditions for the optimization problem.

- ***LSIMS.m.*** Section 5.3.5
  This program computes a single realization from the discrete-time version of the Cass-Koopmans growth model under uncertainty, using Sims's eigenvalue-eigenvector decomposition method and a log-linear approximation. This program calls function ***ss_lsims.m*** to solve for the steady-state as well as to compute the transition matrix in the log-linear approximation to the first order conditions for the optimization problem.

- ***marcet.m.*** Section 5.3.6
  This program solves for the polynomial used in the den Hann-Marcet solution method to approximate the expectations term in the Euler equation in the discrete-time neoclassical growth model under uncertainty. The program calls ***srmarcet.m*** to compute this sum of squared residuals

- ***marcet1.m*** *and* ***marcet2.m.*** Section 5.3.6
  These two programs solve the discrete-time version of the Cass-Koopmans growth model under uncertainty, using the parameterized expectations method of den Hann and Marcet. *marcet1.m* uses command "fminunc to minimize the sum of squared residuals, while *marcet2.m* uses a Gauss-Newton algorithm. In both cases, *marcet.m* needs to be used in advance to compute initial values for the parameters in the polynomial used to represent the conditional expectations.

- ***BK impulse response.m*** Section 5.4.3
  This program computes responses to an impulse in the innovation in the autoregressive process for productivity in the stochastic, discrete-time version of the Cass-Koopmans economy. The Blanchard and Kahn method is used times series for the endogenous variables along the transition.

- ***methods.m*** Section 5.5
  This program solves solve the discrete-time version of the Cass-Koopmans growth model under uncertainty, using different solution methods: linear-quadratic approximation, Sims' method with either a linear or a log-linear approximation, Blanchard-Kahn method, Uhlig method, and the parameterized expectations method by den Hann-Marcet. The user must select the specific solution method by activating the corresponding line in the program, and deactivating the alternative lines. N simple realizations are obtained and the characteristics of the solution are summarized through sample statistics: mean, standard deviation, coefficient of variation, volatility relative to that of output, correlation coefficients with

contemporaneous, lagged or leaded output,…). We compute these statistics for each realization to then obtain the average and standard deviation for each one of them across the set of simulations (sample realizations). These basic statistics needed for a discussion along the lines of the Real Business Cycle literature. Depending on the solution method chosen, the program calls functions: ***MLQ.m, MSIMS.m, MLSIMS.m, MBK.m, MUHLIG.m, MMARCET.m.***

- ***coll_cheb.m.*** Section 5.5
  This program solves the deterministic, Cass-Koopmans optimal growth model using projection methods and Chebychev polynomials. It calls functions ***res.m*** and ***Cd.m***.

- ***g_cheb_s_3.m.*** Section 5.5
  This program solves the stochastic version of the optimal Cass-Koopmans growth model using projection methods and Chebychev polynomials of order 3. It calls functions ***res_s_3g.m, psi_3.m, Cds_3.m, root_h.m, herm.m***.

- ***g_cheb_s_4.m.*** Section 5.5
  This program solves the stochastic version of the optimal Cass-Koopmans growth model using projection methods and Chebychev polynomials of order 4. It calls functions ***res_s_4g.m, psi_4.m, Cds_4.m, root_h.m, herm.m***.

- ***g_cheb_s_5.m.*** Section 5.5
  This program solves the stochastic version of the optimal Cass-Koopmans growth model using projection methods and Chebychev polynomials of order 5. It calls functions ***res_s_5g.m, psi_5.m, Cds_5.m, root_h.m, herm.m***.

Chapters 6. Basic models of endogenous growth

- ***DynamicLaffer.xls***. Section 5.5.1
  The possibility of dynamic Laffer effects in the AK endogenous growth economy is shown in this file.

- ***AK_Stochastic.m***. Section 5.6.2
  This program computes a single realization of the numerical solution to the stochastic version of the AK economy.

- ***mAK_Stochastic.m***. Section 5.6.2
  This program computes an arbitrary number (chosen by the user) of realizations of the numerical solution to the stochastic version of the AK economy, from which the sample distribution for any statistic of interest can then be obtained.

- ***Dfase.m***. Section 5.9.3
  This program computes the phase diagram for the Jones and Manuelli endogenous growth economy.

- ***AKModel.xls***. Section 5.9.4
  A single realization from the numerical solution to the stochastic version of the AK economy.

- ***AK_JMs.m***. Section 5.9.4
  This program computes a single realization of the numerical solution to the stochastic version of the Jones and Manuelli endogenous growth economy, from which the sample

distribution for any statistic of interest can be obtained. Realizations from the AK economy can be obtained as a special case.

- ***mAK_JMs.m***. Section 5.9.4
  This program computes a number of realizations (chosen by the user) of the numerical solution to the stochastic version of the Jones and Manuelli endogenous growth economy, from which the sample distribution for any statistic of interest can be obtained. Realizations from the AK economy can be obtained as a special case.

Chapter 7. Additional models of endogenous growth

- ***Simul_diffus.m.*** Section 7.2.3
  This program computes the transmission of a shock in the leader country to the follower country. The program can also be used to generate a realization of the numerical solution to the problems of the leader and the follower countries, given a realization of the stochastic shocks for both countries.

- ***Lucas_ss_w.m.*** Section 7.4.5
  These three programs compute steady-state effects of changes in tax rates on labour income, in the endogenous growth model with human capital accumulation.

- ***Lucas_ss_r.m.*** Section 7.4.5
  These three programs compute steady-state effects of changes in tax rates on capital income in the endogenous growth model with human capital accumulation.

- ***Lucas_ss_c.m.*** Section 7.4.5
  These three programs compute steady-state effects of changes in tax rates on consumption expenditures in the endogenous growth model with human capital accumulation.

- ***Lucas_sim1.m.*** Section 7.4.7
  This program computes a single realization of the endogenous growth model with human capital accumulation, which may include an externality in the production of the final good through the average stock of human capital. It can also be used to solve this model in case of equilibrium indeterminacy. The program uses the full Blanchard-Kahn approach by representing the model in state-space form, described as ‘*First simulation approach’* in the section devoted to describing the solution approach for this model.

- ***Human capital.xls.*** Section 7.4.7
  This EXCEL spreadsheet performs the same exercise as the ***Lucas_sim1.m*** program file, but without the externality produced by the possible presence of the average stock of human capital in the production of the final good.

- ***Lucas_sim2.m.*** Section 7.4.7
  This program computes a single realization of the endogenous growth model with human capital accumulation, without externalities in the production of the final good. The Blanchard-Kahn approach is used to solve for control variables, but the original nonlinear model is used to solve for the variables in levels, described as ‘*Second simulation approach’* in the section devoted to describing the solution approach for this model**.**

- ***Mlucas_sim.m.*** Section 7.4.7
  This program computes multiple realizations of the endogenous growth model with human capital accumulation. The user may include or not externalities in the production of the final good. The program can also be used for the case of indeterminacy of equilibrium, and two

alternative benchmark parameterizations leading to either situation are again provided. The multiple realizations are used to present average values of a wide variety of statistics, together with their standard deviations across the set of simulations.

Chapters 8 and 9. Growth in monetary economies

- ***Timing real balances.xls***. Section 8.4.3
  Steady-state welfare cost of inflation in two economies differing in the timing of real balances in the utility function.

- ***Steady-state fiscal policy.xls***. Section 8.5.2
  Steady-state values under alternative monetary policy choices, the economy being under an income tax.
    - *Case* 1 spreadsheet: the government chooses the income tax rate as well as the steady-state level of outstanding debt and the rate of inflation, while the steady-state levels of real balances and lump-sum transfers to consumers become endogenous
    - *Case* 2 spreadsheet: the government chooses the income tax rate and the steady-state size of the transfer, while leaving the steady-state stock of bonds to be endogenously determined
    - *Case* 3 spreadsheet: the government chooses the level of the steady-state transfer and the steady-state stock of per-capita debt

- ***SS inflation endogenous leisure.xls***. Section 8.6.2
    - *Case* 1 spreadsheet: the government realizes a lump-sum transfer to consumers, which is fully financed by increasing the money supply, and we compute the welfare cost of inflation
    - *Case* 2 spreadsheet: examines the welfare cost of inflation under a different type of separability in the utility function
    - *Case* 3 spreadsheet: examines the welfare cost of inflation under a still different type of separability in the utility function
    - *Case* 4 spreadsheet: the government realizes a lump-sum transfer to consumers, which is financed by increasing the money supply and by a tax on consumption. The government has an inflation target and a constant tax rate on consumption
    - *Case* 5 spreadsheet: the government realizes a lump-sum transfer to consumers, which is financed by increasing the money supply and by a tax on consumption. The government has an inflation target and it keeps constant the level of public debt

- ***Money_M_d.m***. Section 9.4
  This program solves the deterministic monetary economy when the monetary authority chooses the rate of money growth. The program computes steady-state values and a single sample realization for the solution

- ***Money_i_d.m***. Section 9.5
  This program solves the deterministic monetary economy when the monetary authority chooses the nominal rate of interest. The program computes steady-state values and a single sample realization for the solution

- ***Short-run nonneutrality.xls***. Section 9.6
    - ***Change nominal rates*** spreadsheet: government finances a transfer to consumers by printing money, issuing debt and raising proportional taxes on output, while using nominal rates as a control variable for monetary policy
    - ***Once and-for-all money change*** spreadsheet: the government makes a lump-sum transfer to the representative agent, which is financed through seignoriage and bond

issuing. In this case, the government uses the rate of growth of money supply as policy variable and we consider the effects of an experiment by which the growth rate experiences a permanent increase.
    - ***Gradual money change*** spreadsheet: we consider a gradual, but permanent change in money growth.

- ***S_i_npi_s.m***. Section 9.7.1
  A single realization of the solution to the stochastic monetary economy, under the assumption that the government chooses the nominal rate of interest according to a Taylor's rule without the presence of the rate of inflation.

- ***S_i_pi_s.m***. Section 9.7.1
  A single realization of the solution to the stochastic monetary economy, under the assumption that the government chooses the nominal rate of interest according to a Taylor's rule that includes the rate of inflation together with the output gap.

- ***mS_i.m***. Section 9.7.1
  This program computes the solution to the stochastic monetary economy, under the assumption that the government chooses the nominal rate of interest according to a Taylor's rule that includes the rate of inflation together with the output gap. A large number of sample realizations (chosen by the user) is computed, and the sample distribution for any statistic of interest can be obtained.

- ***S_M_s.m.*** Section 9.7.2
  This program computes a single sample realization of the solution for the stochastic monetary economy, under the assumption that the government controls the rate of growth of the money supply.

- ***mS_M.m***. Section 9.7.2
  This program solves the stochastic monetary economy, under the assumption that the government controls the rate of growth of the money supply. A large number of sample realizations (chosen by the user) are computed, and the sample distribution for any statistic of interest can be obtained.

- ***Neokeyn.m***. Section 9.8
  This program computes a single realization of the Neo Keynesian economy described in the text. Monetary policy is implemented by the monetary authority choosing the nominal rate of interest according to a Taylor's rule. It calls function ***nkeyn.m***.

- ***Nkeyprg.m.*** Section 9.8
  This program computes multiple realizations of the same economy considered in the previous program. It computes the first two moments of the empirical distribution for a number of statistics, and it stores the simulated data.

Chapter 10: Empirical Methods 1: Frequentist Estimation.

- ***data_example1.xlsx***. (directory *chapter10/GMM/example1).* Section 10.2.3
  The Excel file 'data_example1.xlsx' contains data on the 3-month and the 10-year interest rates of the United States Treasury bonds (annualized monthly rates) [3-Month Treasury Bill: Secondary Market Rate, Percent, Monthly, 10-Year Treasury Constant Maturity Rate]. The third series is the personal consumption expenditure of private agents in the United States [Personal Consumption Expenditures]. All series have been obtained from the Federal Reserve Bank of St. Louis (FRED) database and the sample runs from January 1979 to November 2018
- ***data.prn***. (directory *chapter10/GMM/example1).* Section 10.2.3
  Ascii data
- ***main.m*** (directory *chapter10/GMM/example1).* Section 10.2.3
  This file is structured as follows: in the first lines we load the data file *data.prn*, and we choose the number of lags used as intruments (which will also be the lags used to compute the Newey-West covariance matrix S); then the parameter β is transformed to ensure that its estimate falls in the interval (0,1); after that, the function *gmmest.m*[1] is executed, and the results are presented in table form
- ***gmmest.m.*** (directory *chapter10/GMM/example1).* Section 10.2.3
  The gmmest.m function minimizes the objective function given in (10.4) while iteratively computing the weighting matrix W using the inverse of the Newey-West covariance matrix S. To do that, this function will call the HSmodel.m routine, which specifies the first order conditions (10.15) of the Hansen and Singleton model for both interest rates, and the moments used in (10.20), besides writing the objective function to be minimized. Throughout the estimation process, the weighting matrix is computed iteratively; the calculation of the moment weights function is calculated through the function gmmweightmatrix.m which is called from gmmest.m.
- ***gmmweightmatrix.m*** (directory *chapter10/GMM/example1).* Section 10.2.3
  This function computes the weighting matrix.
- ***HSmodel.m*** (directory *chapter10/GMM/example1).* Section 10.2.3
  This function computes the first order conditions (10.15)

- ***data_example2.xlsx***. (directory *chapter10/GMM/example2).* Section 10.2.3
  In this example we also use monthly data obtained from the FRED database (Federal Reserve Data from the Bank of St. Louis). The sample runs from January 1982 to December 2006.
  For the consumption variable, we use data on personal consumption expenditures on services and non-durable goods (real personal consumption expenditures on services, non-durable) divided by the population, thus obtaining a series of real private consumption per capita. We calculate the annual growth rate for that variable ($\gamma_{c,t}^{(12)} = (c_{t+12}/c_t) - 1$) and we deduce a "monthly" rate as:

---

[1] This a modified versión of "gmmestimation.m", which was written by Cao Zhiguang (see https://www.mathworks.com/matlabcentral/fileexchange/12114-gmm).

$\gamma_{c,t}^{(1)} = (1+\gamma_{c,t}^{(12)})^{1/12} - 1$, in order to obtain a smoothed monthly consumption growth, since the theoretical model represents monthly data. Thus, the variable (c_{t+1} / c_t) in the model will be measured by the time series data we have obtained for $(1+\gamma_{c,t}^{(1)})$.

As risk-free interest rate we use: i) the nominal yield of the 1-year Treasury bonds (denoted by $r_{rf,t}$ in basis points), on a monthly basis: $r_{rf,t}^{m} = (1+r_{rf,t})^{1/12} - 1$, and ii) the inflation measured by the inter-annual variation $\pi_t^{(12)} = (P_{t+12} / P_t) - 1$ of the Consumer Price Index for All Urban Consumers, on a monthly basis: $\pi_t^{(1)} = (1+\pi_t^{(12)})^{1/12} - 1$. Thus, the risk-free real interest rate will be denoted by: $R_{rf,t} = \dfrac{1+r_{rf,t}^{m}}{1+\pi_t^{(1)}} - 1$.

Finally, as real return on risky assets (stocks) we use the return on a value-weighted portfolio of the S&P 500 Universe (that includes dividends), and we denote by $R_{eq,t}$. It is obtained from the observed nominal equity market returns, correcting for inflation the same way we have just done with the interest rate.

- ***Data_ex2.txt***. (directory *chapter10/GMM/example2).* Section 10.2.3
  Ascii data
- ***main_cx2.m*** (directory *chapter10/GMM/example2).* Section 10.2.3
  The results of example 2 are obtained executing this file. This file, as well as the rest of the functions (gmmest.m, gmmweightmatrix.m, and HSmodel.m), have a structure similar to those described in Example 1.

- ***UNRATE_US.xls*** (directory *chapter10/maximum_likelihood/example1).* Section 10.3.3
  The file UNRATE_US.xls contains quarterly data for the US GDP and unemployment rate, from 1948Q1 to 2018Q1.
- ***gdp_u_us.prn*** (directory *chapter10/maximum_likelihood/example1).* Section 10.3.3
  File gdp_u_us.prn contains the data in ascii format, to be loaded by the Matlab program main_detrend.m.
- ***main_detrend.m*** (directory *chapter10/maximum_likelihood/example1).* Section 10.3.3
  The program is organized as follows: we first load the GDP and unemployment rate data. After that, we take logs of the GDP data and use the *hpjra.m* function to estimate the trend and cyclical components using the Hodrick and Prescott filter. This is the standard filter used by DSGE practitioners to estimate cyclical components. One of the graphs compares the cyclical components estimated by the Clark and the Hodrick-Prescott decompositions.
  After that, we define the parameter vector $\theta = [\phi_1, \phi_2, \sigma_\upsilon, \sigma_\omega, \sigma_e, \alpha_0, \alpha_1, \alpha_2, \sigma_{e_U}, \sigma_{\upsilon_U}]'$, which we will transform to avoid

having to impose constraints on the parameter values when maximizing the likelihood function. Indeed, if the parameters are transformed appropriately, the optimization process will generate estimates that satisfy the theoretical restrictions on their numerical support, without need of imposing any constraint on the optimization problem. Thus, for example, we will guarantee an estimate $\sigma_{\upsilon} > 0$ by estimating a transformed parameter $\tilde{\sigma}_{\upsilon} \in (-\infty, \infty)$ to guarante that $\sigma_{\upsilon} = \exp(-\tilde{\sigma}_{\upsilon} / 10) > 0$ holds. We will also estimate the transformed parameters $\left\{\tilde{\phi}_1, \tilde{\phi}_2\right\}$, defined on the real line by, $\phi_1 = \frac{\tilde{\phi}_1}{1+|\tilde{\phi}_1|} + \frac{\tilde{\phi}_2}{1+|\tilde{\phi}_2|}$, $\phi_2 = -\frac{\tilde{\phi}_1}{1+|\tilde{\phi}_1|}\frac{\tilde{\phi}_2}{1+|\tilde{\phi}_2|}$, thus guaranteeing the stationarity of the AR (2), since the estimates of $\left\{\phi_1, \phi_2\right\}$ that we will recover will make the roots of the characteristic equation of the AR (2) structure in (10.34) to have a modulus less than 1.
Then, the opposite of the log of the likelihood function that is specified in the Matlab function *kf_detrend.m* is minimized. Besides the likelihood function, this function uses the Kalman filter equations as specified above, also computes the period-by-period predictions of the unobservable variables and the covariance matrices of the prediction errors.
Afterwards, graphs with the estimated unobserved components are computed, the variance and covariance matrix of the untransformed estimated parameters is calculated, and the estimates and their standard deviations are tabulated.

- ***kf_detrend.m*** (directory *chapter10/maximum_likelihood/example1).* Section 10.3.3
  The log of the likelihood function is specified in this Matlab function.
- ***hpjra.m*** (directory *chapter10/maximum_likelihood/example1).* Section 10.3.3
  This file is a function to estimate the trend and cyclical components using the Hodrick and Prescott filter.

- ***data.xls*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4

  This file contains the data (downloaded from FRED) as well as the transformations carried out to convert the data to quarterly per capita. We use the following variables: GDP (net of public consumption and net exports, since the model has no public sector or foreign sector), Consumption and Hours Worked per capita downloaded from the Federal Reserve database Saint Louis (FRED). The data are: Real Personal Consumption Expenditures, Real Gross Private Domestic Investment (the sum of these two variables will be our output measure), Hours of Wage and Salary Workers on Private, Nonfarm Payrolls, and Civilian Noninstitutional Population, Ages 16 and Over. The sample runs from the first quarter of 1948 to the second quarter of 2002.

- ***ych.dat*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4
  Data in ascii format
- ***main.m*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4

The structure of this file is as follows: first the data from the ascii file "ych.dat" is uploaded. Then, output and consumption trends are eliminated. Once the parameters have been specified, we transform them to avoid having to impose constraints in the maximum likelihood procedure. Once the initial conditions are defined in the transformed parameters, the opposite of the logarithm of the likelihood function is minimized. The logarithm of the likelihood function is computed in the file *log_likelihood.m*. Once the transformed parameters have been estimated, we recover their untransformed values and calculate the standard deviations associated with each estimated parameter to make inference. The standard deviations are computed as the square root of the main diagonal of the inverse of the Hessian of the logarithm of the likelihood function evaluated on the estimated untransformed parameters. Additionally, this file calculates the impulse response function through the *irf.m* function. Finally, this file also computes the variance decomposition of the forecast errors for the main variables in the model through the decomp_var.m function.

- ***Log_likelihood.m*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4
  The file *log_likelihood.m* is organized as follows: taking as inputs the parametric values, the matrices of the log-linear approximation of the optimality conditions of the model are calculated to compose the matrices of the state-space representation. Then, using the equations of the Kalman Filter, we calculate the logarithm of the likelihood function and change its sign so that we can maximize it with the Matlab optimizer (which is actually a minimizer).
- ***irf.m*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4
  This function uses the estimated parameters to calculate the dynamic effects of a productivity shock or a demand shock on the main macro-magnitudes of the theoretical model
- ***decomp.m*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4
  This file computes the variance decomposition of the forecast errors for the main variables in the model using, as inputs, the matrices A, B, and C of the empirical model in the state-space representation.
- ***state_space_matrices.m*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4
  This function calculates the matrices A, B, and C of the empirical model in the state-space representation

Chapter 11: Empirical Methods 2: Bayesian Estimation.

- ***example_3.m*** (directory *chapter11/example_gibbs_sampling/).* Section 11.4

  In this file a simple linear regression model is estimated using the Gibbs-Sampling algorithm. The model specification is as follows:

$$y_t = \alpha_0 + \alpha_1 x_t + \varepsilon_t = X_t \beta + \varepsilon_t \, , \beta = [\alpha_0 \, , \, \alpha_1]' \, , \, X_t = [1, \, x_t], \, \varepsilon_t \underset{iid}{\sim} N(0, \sigma^2) \; .$$

Data for the dependent variable are first generated from: i) values of 2 and 0.9 for the constant and the slope of the regression model respectively; ii) generating 500 random numbers from a Normal distribution with zero expectation and variance equal to 0.006, as a time series for the disturbance; iii) generating 500 random numbers from a N(0,1) distribution as values for the exogenous variable. Data for the y_t variable are computed from $y_t = 2 + 0.9 x_t + \varepsilon_t$. The sample size is therefore $T = 500$.

Once we have the data $\{y_t \, , \, x_t\}_{t=1}^{T}$, the program starts by estimating the regression model by ordinary least squares as a benchmark for comparison. After that, a prior distribution is proposed for the parameters in β (conditional on σ^2) as follows:

$$\beta \,|\, \sigma^2 \sim N(\beta_0 \, , \, \Sigma_0), \; \beta_0 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \; \Sigma_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} ,$$

which implies very little informative capacity for the prior distribution, since the variances associated with the parameters are high.

Finally, we propose a Gamma distribution with shape and scale parameters equal to zero ($v_0 = \delta_0 = 0$) for the inverse of the variance of the disturbances σ^2, thus indicating a non-informative a priori distribution.

Afterwards, the program starts the Gibbs-Sampling algorithm with 24,000 realizations of which we will only use in estimation the last 16,000. The program ends with a set of graphs like those in Figure 1-Chapter 11.

- ***Cauchy_random.m*** (directory *chapter11/Cauchy_random/).* Section 11.5.1
  This file generates random numbers from a Cauchy distribution using Matropolis-Hastings sampler algorithm. We will use the Normal distribution as our proposed distribution, with a mean given by the previous value of the chain and a standard deviation given by $\sigma = 8$.

- ***Sp500.txt (***directory *chapter11/bayesian_garch11/).* Section 11.6
  Data in ascii format. These are daily logarithmic returns of the S&P500 index for a sample from December 30, 1994 to March 31, 2008.
- ***Bayesian_garch11.m (***directory *chapter11/bayesian_garch11/).* Section 11.6

  This is the main file to be executed. This file calls the *lfv1c.m* and *drchrnd.m* functions. The structure of file *bayesian_garch11.m* is as follows: First, the data for the S&P500 index is loaded and the first logged difference is calculated, to obtain the returns on the index. Next, initial conditions are established for the parameters, which are then transformed as described in (11.30). The logarithm of the likelihood function is minimized (with the opposite sign) to obtain frequentist estimates of the transformed parameters. Then, we recover the original parameters and calculate the Hessian of the likelihood function both for the original parameters (to make inference about them, obtaining their covariance matrix as the inverse of the Hessian), and for the transformed parameters, to use that covariance matrix to generate candidate points for the Bayesian estimation. We will also use that covariance matrix of the transformed parameters to generate candidate points for the transformed parameters in the Bayesian estimation. Finally, using the Metropolis-Hastings with Random Walk algorithm (*Random Walk Metropolis-Hastings*), the posterior distribution of the parameters is sampled. The values for the hyper-parameter are those obtained from the frequentist estimation. The covariance matrix used to generate the candidate points is constructed as a diagonal matrix whose elements are those obtained from the estimation of the covariance matrix of the transformed parameters using the values from the frequentist estimate.

- ***lfv1c.m (***directory *chapter11/bayesian_garch11/).* Section 11.6
  This function generates the logarithm of the likelihood function, with the opposite sign, since we are going to calculate both, the frequentist and the Bayesian estimates. Since the former is obtained maximizing the likelihood function and Matlab optimization algorithms solve minimization problems, we program the likelihood function with the opposite sign.
- ***drchrnd.m (***directory *chapter11/bayesian_garch11/).* Section 11.6

  This function generates random numbers from a Dirichlet distribution. We will use it to generate random numbers from the prior distribution of the parameters, and these realizations will allow us to compare the prior and the posterior empirical densities of these parameters.

- ***ych.dat (***directory *chapter11/bayesian_DSGE).* Section 11.7
- Data in asci format. This data is the same as in file "***data.xls*** (directory *chapter10/maximum_likelihood/example2).* Section 10.3.4" in chapter 10.
- ***main_bayesian.m (***directory *chapter11/bayesian_DSGE).* Section 11.7

  This file contains the code for the Bayesian estimation of the parameters of this model (Hansen's (1985) Model). This file is structured analogously to file *bayesian_garch11.m* from the previous section. First, we load data on consumption, output and hours and eliminate a linear trend from the logarithms of these time series. Next, we give initial values to the structural parameters of the model and decide the desired number of realizations of the posterior distribution. These will be 39,000 in our exercise, and we will keep for inference the last 30,000, discarding the first 9,000 to eliminate any possible dependence from the initial conditions. After that, we specify the hyperparameters of the selected prior distributions.

  A candidate point is then generated for the transformed parameters using the average of the realizations obtained from the assumed prior distributions, and transforming those averages according to (11.38) (See Chapter 11). With that candidate point we calculate the prior density using (11.39), as well as the likelihood of this initial candidate point, using (11.40), through the functions

Hansen_model.m, which computes the matrices that describe the model in state-space representation. The *lfvkf.m* function is then used to compute the likelihood function using the Kalman filter. From this starting point, we start the loop that will sample in the posterior distribution in the manner described in detail when we presented the Metropolis-Hastings algorithm. During the sampling we will collect, in addition to the posterior realizations, the variance decomposition and the impulse response functions of the main variables in the model to the supply and demand shocks, using the functions *state_space_matrices.m, decomp_var.m, and irf.m* (from chapter 10).

Finally, we draw a random sample from the prior distributions to compare their histogram with that of the posterior distributions obtained with the Metropolis-Hastings algorithm. This is an interesting comparison. The difference between both densities shows the extent to which the data are informative on a specific parameter. If the prior and posterior distributions for a given parameter are very similar, it means that the data are not very useful to identify the true value of that parameter: our uncertainty on its true value is the same before looking at the data than afterwards.

The file ends with the presentation of tables that collect the posterior empirical mean for each parameter and its 95% confidence bands, as well as the variance decompositions and their confidence bands. In Tables 4 and 5 we summarize those results. Finally, the program file displays graphs gathering the impulse response functions for the variables that make up the theoretical model to demand and supply shocks, as well as their confidence bands.

- ***hansen_model.m (***directory *chapter11/bayesian_DSGE).* Section 11.7
  This function computes the matrices that describe the model in state-space representation using a log-linear approximation of the optimality conditions.
- ***lfvkf.m (***directory *chapter11/bayesian_DSGE).* Section 11.7
  This function computes the likelihood function using the Kalman filter.
- ***irf.m (***directory *chapter11/bayesian_DSGE).* Section 11.7
- This function uses the estimated parameters to calculate the dynamic effects of a productivity shock or a demand shock on the main macro-magnitudes of the theoretical model
- ***decomp_var.m (***directory *chapter11/bayesian_DSGE).* Section 11.7
  This file computes the variance decomposition of the forecast errors for the main variables in the model using, as inputs, the matrices A, B, and C of the empirical model in the state-space representation
- ***state_space_matrices (***directory *chapter11/bayesian_DSGE).* Section 11.7
  This function calculates the matrices A, B, and C of the empirical model in the state-space representation