

# Handout for the Modeling Workshop

## Table of Contents

I. Introduction .....	2
II. Before You Start.....	2
III. The Model.....	3
A. Aggregate demand .....	3
B. Aggregate supply.....	3
C. Uncovered Interest Rate Parity .....	4
D. Policy Rule.....	6
IV. Model Solution .....	7
V. Model Calibration .....	9
A. Calibration of Long-term Parameters .....	9
B. Calibration: Business-Cycle Properties.....	10
C. Some Useful Calibration Hints .....	12
D. How to Check the Model Properties .....	13
VI. Forecasting.....	14
A. Database Preparation.....	14
B. The Forecast .....	17
C. Interpreting the Forecast.....	19
Software Installation .....	20
The Model Code .....	23

## I. INTRODUCTION

This note introduces a simple output gap model that can be calibrated to fit stylized facts in most countries and to prepare a simple inflation forecast. It is a trimmed-down version of a generic Quarterly Projection Model used in several central banks and other institutions such as the IMF (Berg, Karam and Laxton 2006a, 2006b). The objective of this class of models is to help decide on an appropriate level of the policy interest rate, given the inflation target and the current state of the economy.

The model is solved numerically using the Matlab software and IRIS Toolbox, however, no prior knowledge of either Matlab or IRIS is required. The solution algorithm and commands are explained in this note.<sup>1</sup> We focus on three practical aspects of the modeling process in the context of inflation targeting: (i) calibration; (ii) testing the model properties and data preparation; and (iii) inflation forecast, its analysis and interpretation. Software installation is described in [Appendix I](#) and computer codes in [Appendix II](#).

## II. BEFORE YOU START

Before you can start modeling in the IRIS environment, you need to install it and a few other freeware programs (MiKTeX, Ghostscript, Ghostgum, and X-12) on your computer. Detailed instructions are contained in [Appendix I](#) and we suggest that you follow them as closely as possible.

Start each session (after you open Matlab) by running the following line from the Matlab command window: `addpath c:/iris-toolbox; irisstartup`; replacing `c:/iris-toolbox` with your own main IRIS folder chosen during installation. Navigate to your working folder to check that it contains the following files (they will be described in more detail later):

1. `model.mod`
2. `readmodel.m`
3. `modelproperties.m`
4. `makedata.m`
5. `forecast.m`
6. `data.csv`

If the course is Monetary policy analysis course, the folder contains also files

7. `kalman.m`
8. `in_sample.m`

With the exception of '`model.mod`' and '`data.csv`', all files are Matlab programs (extension '.m') and work in the Matlab environment only. The programs get started in the following

---

<sup>1</sup> The IRIS Toolbox (<http://www.iris-toolbox.com/>) is a Matlab-based package for advanced macroeconomic modeling written by a development team led by Jaromír Beneš (current affiliation: IMF Research Department, e-mail: [jaromir.benes@gmail.com](mailto:jaromir.benes@gmail.com)).

way. Type the name of a program, for instance ‘`makedata`’, in the command window and push [enter]. Once the screen shows `>>` the program has been run successfully. You can check that all programs are functioning by executing them in the following sequence: ‘`modelproperties`’, ‘`makedata`’, ‘`forecast`’, ‘`kalmanfilter`’, ‘`in_sample`’. From Windows explorer, in the appropriate folder, double click on files that are generated to report simulation results: [Report\\_Shocks.ps](#), [Stylized\\_facts.ps](#), [Forecast.ps](#), [Filter.ps](#), and [In\\_sample](#). Click [OK] and you should see charts and tables of various model outputs.

### III. THE MODEL

The model we use for our exercises is a reduced-form, new-Keynesian model, sometimes also called the output gap model. More complicated versions of it are being utilized in many central banks all around the world. The model consists of four basic behavioral equations and several identities (the corresponding IRIS codes are contained in the [model.mod](#), i.e., [Appendix IV](#)). The behavioral equations are:

1. Aggregate demand (IS curve)
2. Aggregate supply (Phillips curve)
3. Uncovered Interest Rate Parity
4. Monetary Policy Rule

#### A. Aggregate demand

The aggregate spending relationship corresponds to the open economy version of the traditional IS curve and takes the form:

$$\hat{y}_t = a_1 \hat{y}_{t-1} - a_2 mci_t + a_3 \hat{y}_t^* + \varepsilon_t^y, \quad (1)$$

where  $\hat{y}_t$  is the deviation of the log of output from its noninflationary level, i.e., the output gap;  $mci_t$  is the real monetary condition index that is defined as a weighted average of deviations of the long-term real interest rate,  $r_t$ , from its neutral (noninflationary) level, and deviation of the real exchange rate,  $z_t$ , from its trend level;  $\hat{y}_t^*$  is the foreign output gap and  $\varepsilon_t^y$  is an aggregate demand shock. The aggregate demand shocks are governed by a Normal distribution and have no serial correlation. The coefficients  $a_1$ ,  $a_2$ , and  $a_3$  capture the persistence of output; the impact of monetary conditions on real economic activity; and the impact of foreign environment, respectively.

#### B. Aggregate supply

The aggregate supply equation (the Phillips curve) is defined as follows:

$$\pi_t = b_1 \pi_{t-1} + (1 - b_1) \pi_{t+1}^e + b_2 rmc_t + \varepsilon_t^\pi, \quad (2)$$

where  $\pi_t$  is the annualized quarterly change of the consumer price index, i.e., inflation;  $\pi_{t+1}^e$  denotes model-consistent inflation expectations;  $rmc_t$  is the gap in firms’ real

marginal costs; and  $\varepsilon_t^\pi$  is an aggregate supply shock. Similar to demand shocks, the aggregate supply shocks are governed by a Normal distribution and have no serial correlation.

The supply relationship encompasses multi-period, overlapping nominal contracts of domestic producers as well as importers. The latter is an important feature of small open economies that typically have a powerful exchange rate channel of monetary transmission. That is why we define in (3) the  $rmc_t$  as a weighted average of output gap (domestic producers) and the gap in real exchange rate (importers) with the coefficient  $(1 - b_3)$  approximating the weight of imported goods in the consumer basket. In the equation below,  $\hat{z}_t$  denotes deviations of the real exchange rate from its neutral (noninflationary) level.

$$rmc_t = b_3 \hat{y}_t + (1 - b_3) \hat{z}_t. \quad (3)$$

It is important to model expectations properly, capturing the behavior of agents, some of which may be forward-looking, using model consistent (rational) expectations,  $E_t \pi_{t+1}$ , while others are backward-looking. Economic agents who are assumed to be fully forward-looking comprise  $(1 - b_1)$  of the population and  $b_1$  agents follow the rule of thumb of past inflation. Another way of thinking about this parameter is the persistence of inflation—the more persistent inflation, the higher is  $b_1$ . The coefficient  $b_2$  captures the influence of the gap in the real marginal costs on inflation (the slope of the Phillips curve) and measures the sacrifice ratio, i.e., how much output will be lost in order to bring inflation down by 1 percentage point.

### C. Uncovered Interest Rate Parity

We capture the relationship with the rest of the world using alternative versions of the uncovered interest rate parity condition. In its pure forward-looking version, the UIP relates the behavior of domestic and foreign interest rates, and the nominal exchange rate:

$$s_t = s_{t+1}^e + (i_t^* - i_t + prem_t) / 4 + \varepsilon_t^s, \quad (4)$$

where  $s_t$  is the nominal exchange rate;  $s_{t+1}^e$  is the model consistent expectation of the nominal exchange rate in period  $t+1$ ;  $i_t$  is the domestic nominal interest rate (annualized);  $i_t^*$  is the foreign nominal interest rate (annualized);  $prem_t$  is the risk premium; and  $\varepsilon_t^s$  is the exchange rate shock. With the canonical UIP, the model has difficulty in matching the observed behavior of the economy (looking, say, at impulse responses), because the exchange rate has little persistence (Beneš, Hurník and Vávra, 2008). The pure forward-looking element in Equation (4) forces the current exchange rate level to adjust immediately to the sum of all future interest rate differentials implied by the model behavior. This is at odds with the observed exchange rates that tend to move more slowly.

There are a few alternative ways of increasing the persistence of the exchange rate. One way is to modify the pure UIP by containing its forward-looking nature. This approach substitutes the model-consistent exchange rate expectations in Equation (4) by a combination of backward-looking and model-consistent (forward-looking) expectations ( $s^e$ ). This can also

be interpreted as assuming backward-looking expectations for a portion of the agents in the economy and forward-looking expectations for the remaining. The approach below is more general than that of Berg, Karam and Laxton (2006a, 2006b), allowing for a nonzero rate of growth of the exchange rate in the long-run. Equation for exchange rate expectations ( $s^E$ ) takes the form:

$$s_{t+1}^E = e_1 s_{t+1}^e + (1 - e_1)(s_{t-1} + 2\Delta\bar{s}_t), \quad (5)$$

where

$$\Delta\bar{s}_t = \bar{\pi}_t - \bar{\pi}_t^* + \Delta\bar{z}_t. \quad (6)$$

In this setting,  $\bar{\pi}_t$  is domestic inflation target;  $\bar{\pi}_t^*$  is foreign inflation target; and  $\Delta\bar{z}_t$  is the trend (long-run) change in the real exchange rate. The coefficient  $e_1$  determines the degree of forward-looking behavior in the financial market or the portion of the agents that are assumed to be forward-looking.

The second element in Equation (5),  $(s_{t-1} + 2\Delta\bar{s}_t)$ , is the backward-looking exchange rate expectation, which projects the exchange rate in period  $t + 1$  as an extrapolation of the past ex-change rate using the trend rate of growth of the real exchange rate and the average inflation differential approximated by the difference in inflation targets. While such expectations are not model-consistent in the short-run, they are consistent in the long-run, in line with the finding that the PPP holds at longer horizons only. The term  $\Delta\bar{s}_t$  is the change in the exchange rate consistent with long-term economic fundamentals represented by the inflation targets and the real exchange rate trend. By construction,  $\Delta\bar{s} = \Delta s$  in the long-run, so the long-run properties of the model are intact. The above extension of the UIP is designed to make the nominal exchange rate more persistent, however, it does not reduce the overall volatility of the exchange rate of the simulated model (the fluctuations in the nominal exchange rate are more persistent, but its standard deviation is not necessarily smaller).

Alternatively, the UIP can be modified to account for the central banks that use the FOREX interventions actively (and the exchange rate channel) to meet the inflation target.<sup>2</sup> However, to model the exchange rate consistently, the change in the ER target must be defined as a “sum” of the inflation differential and trend appreciation as in Equation (6). Moreover, to avoid unrealistic exchange rate behavior in the first simulated period, the last observed level of the exchange rate needs to be taken as the target at time  $t-1$ . The targeted exchange rate is then defined as:

$$s_t^T = s_{t-1}^T + \Delta\bar{s}_t / 4, \quad (7)$$

$$s_{t-1}^T = s_{t-1}, \quad (8)$$

---

<sup>2</sup> A similar structure of the UIP is used in the forecasting models in Botswana, Belarus, Ukraine, and Serbia. (Beneš, Hurník and Vávra, 2008).

where  $s_t^T$  is the exchange rate target at time  $t$ . Thus, first, Equation (6) determines the long-run change in the nominal exchange rate as implied by the PPP theory and makes the *change* in the exchange rate target consistent with the chosen inflation target. Second, the target for the *level* of the nominal exchange rate is determined by its connection to the last observed level of the nominal exchange rate.

The UIP then becomes a “weighted average” of the exchange rate target and the canonical UIP

$$s_t = e_1 s_t^T + (1 - e_1)(s_{t+1}^e + (i_t^* - i_t + prem_t)/4) + \varepsilon_t^s, \quad (9)$$

with the coefficient  $e_1$  measuring the degree of the central bank intervention on the FOREX market. The extended UIP defined as in Equation (9) stabilizes the exchange rate in a way that resembles a managed exchange rate regime (Beneš, Hurník, and Vávra, 2008).

The stability of the exchange rate, however, comes at the cost of persistent deviations of inflation from the inflation target whenever a shock hits the economy. Intuitively, the introduction of an exchange rate target is equivalent to switching from an inflation target to a price level target. Even if the central bank publicly defines its target as a rate of growth of the price level, it simply means that the target is a price level growing at some constant rate. Any overshooting of such an inflation target must still be compensated by subsequent undershooting and *vice versa* to jointly satisfy the inflation and exchange rate targets.

The model also satisfies the long-run version of the UIP expressed in real terms. This version of the UIP links the trend values for real exchange rate appreciation (either due to the Balassa-Samuelson effect or some “convergence inflation” as in Čihák and Holub, 2003) to the trend values of domestic and foreign real interest rates:

$$\Delta \bar{z}_{t+1} = \bar{r}_t - \bar{r}_t^* - prem_t. \quad (10)$$

The steady-state values of  $\Delta \bar{z}$ ,  $\bar{r}$  and  $\bar{r}^*$  are set as parameters in this model, requiring prior assessment of these trend values,<sup>3</sup> while the risk premium is calculated endogenously, assuring the existence of a consistent steady state. Values of those parameters frame the forecast over the medium to long run.

#### D. Policy Rule

The model is closed by a policy reaction function of the monetary authority (Taylor, 1993). The rule is defined typically for a central bank maintaining a floating exchange rate regime, however, it can be modified to reflect the objective of controlling the exchange rate.

---

<sup>3</sup> For example, one could assume that a fast-converging transition economy would have an equilibrium real appreciation to the tune of 3-5 percent annually.

### “Floaters”

For simplicity, we take the three-month interest rate to be the instrument of monetary policy, and the authority is assumed to respond to deviations of next-period inflation from its target and to the output gap. In other words, we assume that the credit markets flawlessly transmit the changes in the policy rate into the money-market rates. The last-period policy stance affects the current policy stance allowing the authority to smooth interest rates by adjusting them gradually to the desired level implied by the deviations of inflation and output from equilibrium. Mathematically,

$$i_t = f_1 i_{t-1} + (1 - f_1)(i_t^n + f_2(\pi_{t+1}^e - \pi^T) + f_3 \hat{y}_t) + \varepsilon_t^i, \quad (11)$$

where  $i_t$  is the domestic short-term nominal interest rate and  $\varepsilon_t^i$  is a policy shock. The monetary authority is fully forward-looking and uses model-consistent inflation expectations,  $\pi_{t+1}^e$ . The policy-neutral rate,  $i_t^n$ , is such that would keep the output gap unchanged. We calculate it as the sum of the trend real interest rate and model-consistent inflation expectations:

$$i_t^n = \bar{r}_t + \pi_{t+1}^e. \quad (12)$$

### “Countries with Managed Float” and “Peggers”

Central banks attempting to control the exchange rate using foreign exchange interventions find it difficult to control simultaneously also the money market and hence the short-term nominal interest rate. The domestic money market interest rate  $i_t$  becomes partly determined by the UIP equation and the policy rule needs to be modified as follows:

$$i_t = g_1(\Delta s_{t+1} + i_t^* + prem_t) + (1 - g_1)(f_1 i_{t-1} + (1 - f_1)(i_t^n + f_2(\pi_{t+1}^e - \pi^T) + f_3 \hat{y}_t)) + \varepsilon_t^i, \quad (13)$$

The coefficient  $g_1$  reflects the degree of control the central bank retains over the domestic money market (if  $g_1 = 0$ , then the bank has retained full control; if  $g_1 = 1$  the bank has lost control). Of course, the extended version of the policy rule needs to be dovetailed with the uncovered interest parity equation (Equation (9)) and the  $e_1$  coefficient (“high” values of  $g_1$  require “high” values of  $e_1$ , and vice versa). Setting simultaneously a “low” value of  $e_1$  and a “high”  $g_1$  would lead to system indeterminacy. Intuitively, the central bank chooses whether to stabilize inflation through controlling either the exchange rate or interest rate.

For simplicity we work with the three-month interest rate and do not use long-term interest rates. The model includes further identities and transformations that are reported in the model codes.

## IV. MODEL SOLUTION

The solution of this model involves four steps. The main complication arises due to the need for calculating model consistent expectations as explained below. Without the calculation of model consistent expectations, the problem at hand is quite straightforward to solve in many mathematical software as it is a system of four linear equations with four unknowns. Solution

of the full model with expectations also requires solving the same set of four linear equations not just once but potentially many times.

To see the intuition of how the solution algorithm works, imagine that the only expectations operator is in the Phillips curve. See below that on the right hand side of the equation, one period ahead expectation of inflation appears while the left hand side is the contemporaneous inflation which we are trying to solve for. We run into a chicken and eggs problem because to be able to solve for contemporaneous inflation we need to know expected inflation that appears on the right hand side. Similarly, we cannot calculate expected inflation unless we know contemporaneous inflation as a function of other endogenous and exogenous variables.

$$\pi_t = b_1 \pi_{t-1} + (1 - b_1) \pi_{t+1}^e + b_2 rmc_t + \varepsilon_t^\pi,$$

In such circumstances typically the solution requires iterations, in other words, specifying a guess for the contemporaneous inflation and using this guess to calculate the right hand side of the Phillips curve which would imply a contemporaneous inflation that does not need to be the same as the initial guess. At this point the initial guess would be modified depending on how different the inflation implied by the Phillips curve is. Then this new guess would be used to recalculate the right hand side of the Phillips curve which will yield yet another inflation figure. The guess would be updated similarly and this procedure would continue until the initial guess is close enough to the inflation implied by the Phillips curve.

Below are more detailed and specific steps on how the algorithm works:

- 1- Guess inflation. Let's call it  $\pi_t^{old}$ . Note that here the superscript "old" does not refer to time, it is the guess used in the first iteration. If there is prior information about how the solution for inflation will look like, this information can be incorporated into the guess. If not, a guess of  $\pi_t^{old} = 0$  should work.
- 2- Use the Phillips curve to recalculate  $\pi_t$ . Specifically, calculate  $b_1 \pi_{t-1} + (1 - b_1) \pi_{t+1}^e + b_2 rmc_t + \varepsilon_t^\pi$  using  $\pi_t^{old}$  to evaluate the expected inflation. This calculation will deliver a different inflation:  $\pi_t^{new}$ . Note that at this point  $\pi_t^{old} \neq \pi_t^{new}$ . There is no reason for these two inflation values to be the same unless the initial inflation guess was "very educated."
- 3- Update initial inflation guess with the new guess:  $\pi_t^{old} = \pi_t^{new}$ .
- 4- Repeat steps 2-3 until  $|\pi_t^{old} - \pi_t^{new}| \leq \zeta$  where  $\zeta$  is a small number. Intuitively, one would need to continue iterating until the absolute value of the difference between the two values of inflation is small enough. This is called iterating until "convergence." The higher the accuracy needed from this numerical solution, the smaller  $\zeta$  needs to be.



## V. MODEL CALIBRATION

The baseline version of the model is based on “reasonable” values for all the parameters, however, the model exhibits comparatively low persistence with these parameters. The calibration process can be split into two parts: 1) the calibration of the trend variables, determining the steady-state levels of the model, 2) the calibration of individual equations parameters, pinning down the business-cycle properties of the model.

### A. Calibration of Long-term Parameters

The model converges to the long-term trend values of the real exchange rate, domestic and foreign real interest rates, and risk premium and we impose these values via parameters.<sup>4</sup> For simplicity, also the targets for domestic and foreign inflation are set as parameters. The choice of the inflation target predetermines the inflation forecast—the forecasted inflation converges to the target.

The model structure set in Equations (1) to (7) determines important long-run relationships. The nominal interest rate converges to the neutral rate that in turn equals to the sum of the trend real interest rate and the inflation target. (In the steady-state, inflation expectations equal the inflation target). The steady-state change in the nominal exchange rate becomes the sum of steady-state inflation differential vis-à-vis the world and steady-state real exchange rate appreciation. The risk premium is calculated endogenously given the parameterization of domestic and foreign real interest rates and the change in the real exchange rate. This approach not only keeps the model simple but also ensures that the trend values in Equation (5) are on a consistent path.

The model requires setting six long-run parameters ([Appendix III](#)):

1. Inflation target
2. Foreign long-run inflation (or foreign inflation target)
3. Domestic trend real interest rate
4. Foreign trend real interest rate
5. Equilibrium real exchange rate appreciation/depreciation
6. Potential output growth

We recommend that after changing to the model structure you check the steady-state consistency of the model. This is done by first typing `[m,p,mss] = readmodel` and pushing [ENTER] in the Matlab command window, instructing the program to solve the model steady-state and reduced form, while replacing the old model parameters with the new ones. The command returns information about the model solution, its parameterization and steady-state. The screen message is self-explanatory and we suggest checking the following:

1. The domestic and foreign inflation rates equal their respective inflation targets.

---

<sup>4</sup> Gap models do not ensure the stock-flow consistency of its variables and the steady-states have to be imposed. For this reason we prefer to call these variables long-run trends rather than steady states.

2. The domestic and foreign real interest rates equal their trend parameters.
3. The change in the real exchange rate equals the assumed trend.
4. The risk premium equals the difference between domestic and foreign real interest rates minus the change in the real exchange rate (equation 5).
5. The change in the nominal exchange rate must equal the inflation differential adjusted for the change in the real exchange rate.
6. All the gaps converge to zeros.

Violating any of these conditions will result in either inflation missing its target at the end of the forecast period or a non-convergence of the model. The most likely mistakes are:

1. A wrong parameter sign.
2. Violated linear homogeneity in one of the relevant equations. For example, you may have inadvertently set  $f_1 > 1$  in the Taylor rule. See Table 1 for further references on linear homogeneity conditions.

### B. Calibration: Business-Cycle Properties

In this section we discuss model calibration to reflect country-specific, business-cycle conditions by modifying parameters  $a_1$  to  $g_1$  in the file ‘[readmodel](#)’. Table 1 summarizes the baseline parameterization of the model.<sup>5</sup>

Following Berg, Karam, and Laxton (2006), in setting the parameters values we follow (i) the economic theory; (ii) international experience; and (iii) domestic-economy stylized facts, based, for example, on estimated parameter values in a structural VAR model. The estimated values may be used only to the extent that they do not violate model assumptions. For example, short-sample regressions for fast-growing emerging market economies are known to suggest that inflation declines when output expands above its potential (the estimated  $b_3$  coefficient in the Phillips curve is negative). However, using a negative value for  $b_3$  would violate the model convergence toward its steady.

Table 1. Baseline Parameters and Other Suggestions

$a_1$	<b>Output gap persistence</b> varies between 0.1 (extremely low persistence) to 0.95 (high persistence). The linear homogeneity condition: $0 < a_1 < 1$ . <i>Calibration:</i> Run an OLS regression of a log of output on its lagged value and a trend (linear, Hodrick-Prescott, etc.)
-------	--

<sup>5</sup> The model uses two additional parameters that we suggest keeping unchanged. First, the parameter  $h_1$  determines the speed of convergence of domestic variable trend values with respect to past estimated values and to their steady-state values and is set to 0.9. For example, the trend real interest rate reflects past trend rates and the steady-state real interest rate:  $\bar{r}_t = h_1 * \bar{r}_{t-1} + (1 - h_1) * \bar{r}_{SS}$ . Second, the parameter  $h_2$  determines the persistence of the foreign output gap:  $y_t^* = h_2 * y_{t-1}^* + \varepsilon_t^{y^*}$ . See the file [model.mod](#) for the code.

$a_2$	<b>Pass-through from monetary conditions to real economy.</b> The value varies between $-0.1$ (low impact) to $-0.5$ (high impact); the higher the parameter the more responsive is the output gap to changes in monetary policy and, hence, policy reactions need to be less pronounced. The linear homogeneity condition: $-0.9 < a_2 < -0.1$ . <i>Calibration:</i> We suggest basing this parameter either on an impulse-response function in a structural VAR (if available) or on expert assessment.
$a_3$	<b>Impact of foreign demand on the output gap</b> typically varies from 0.1 to 0.5. <i>Calibration:</i> Base the calibration on the export-to-GDP ratio.
$a_4$	<b>The relative weight of the real interest rate and real exchange rate in real monetary conditions in the IS curve (<i>mci</i>).</b> The value varies between 0.3 (open economy) and 0.8 (closed economy). <i>Calibration:</i> Expert assessment.
$b_1$	<b>Inflation persistence</b> determines the share of forward-looking versus backward-looking agents on the goods markets. The value varies between 0.4 (low persistence) and 0.9 (high persistence), the higher is the share of forward looking agents (the lower is the parameter). The less persistent the model becomes, the less pronounced policy reactions are required for a given disinflation goal. The linear homogeneity condition: $0 < b_2 < 1$ , otherwise monetary policy cannot bring inflation toward the target with zero gaps. <i>Calibration:</i> Run an OLS regression of the rate of inflation (quarter-on-quarter) on its lagged value and complement with expert assessment.
$b_2$	<b>The impact of real marginal costs on inflation (policy pass-through).</b> The value typically varies between 0.1 (low impact and high sacrifice ratio) to 0.5 (strong impact and low sacrifice ratio). The higher the parameter the less costly is disinflation. <i>Calibration:</i> Run an OLS regression of the rate of inflation on output gap and complement with expert assessment. Alternatively, data permitting, you can compare the output gap and decline in inflation during a clearly defined disinflation period (Ball, 1993). The latter technique is unlikely to produce meaningful results for emerging/transition economies.
$b_3$	<b>The relative weight of output gap and real exchange rate gap in firms' real marginal costs.</b> [(1- $b_3$ ) is the share of imported goods in the consumption good basket]. The value typically varies between 0.9 (relatively closed economy) and 0.6 (open economy). <i>Calibration:</i> Base the calibration on the share of imported goods in the CPI basket. While such a number is rarely published by the statistical office, some central banks calculated such breakdowns.
$e_1$	<b>Exchange rate persistence or central bank presence on the FOREX market.</b> The value varies between zero (forward-looking FOREX market or no central bank interventions) to 0.9 (either heavily backward-looking FOREX agents or a central bank heavily intervening on the FOREX market). <i>Caveat:</i> Interventions stabilize the exchange rate but cannot violate the steady-state consistency of the model—e.g., the exchange rate will never appreciate as long as there is on average a positive inflation differential ( $\pi_t - \pi_t^* \geq 0$ ) and no trend real exchange rate appreciation. The linear homogeneity condition: $0 < e_1 < 1$ . <i>Calibration:</i> Expert assessment.
$f_1$	<b>Policy rate persistence in the Taylor rule.</b> The value varies between zero (no persistence in policy setting) to 0.8 (“wait-and-see” policy). <i>Caveat:</i> a value above 0.7 may be too high when the model is calibrated as backward-looking. The linear homogeneity condition imposes $0 < f_1 < 1$ , otherwise the model does not converge. <i>Calibration:</i> Run an OLS regression of the policy rate on its lagged value and complement with expert assessment.

$f_2$	<b>Weight put by the policy maker on deviations of inflation from the target in the policy rule.</b> Ranges typically vary from 0.3 to 1. The linear homogeneity condition: $f_2 > 0$ (Taylor principle), otherwise monetary policy does not stabilize the economy. <i>Calibration:</i> Expert assessment.
$f_3$	<b>Weight put by the policy maker on output gap in the policy rule.</b> The linear homogeneity condition: $f_3 > 0$ , otherwise the model does not converge. Typically varies from 0.3 to 1. <i>Calibration:</i> Expert assessment.
$g_1$	<b>Central bank's control of the domestic money market and its short-term nominal interest rate.</b> The value varies between 0 (full control of the short-term rates) and 1 (no control of the short-term rates). The latter case ( $g_1=1$ ) implies that the central bank uses nominal exchange rate to stabilize inflation by intervening in the FOREX market. <i>Caveat:</i> high values of this coefficient in combination with low values of the $e_1$ parameter invalidate the model solution. <i>Calibration:</i> Expert assessment.
$t_1$	<b>Speed of exchange rate adjustment</b> from actual data to the exchange rate target ( $t_1$ ). This coefficient is used only if an exchange-rate-target adjusted UIP is used. Plausible values are: $0 < t_1 < 0.5$

In addition to bivariate regressions, structural VAR models and their impulse responses may be used for parameter calibration of exchange rate or output shocks. Many such VARs have been published on national central bank websites. We would caution, however, against using the impulse responses to the policy shock as the model we present is designed to capture systematic policy behavior. In other words, the monetary authority is assumed to fight against the shocks as opposed to creating them and the policy pass-through estimated in VAR model is likely to be underestimated.

To summarize, the calibration exercise includes the following steps. First, start with the parameter values established in the theory. Second, look for a reasonable estimate of the IS curve, Phillips curve, or policy rule. If there is no such estimate available, run a simple bivariate regression yourself to get a sense of the persistence in the economy. Third, look for a structural VAR model already estimated for the economy you are calibrating, focusing especially on the exchange rate pass-through.

### C. Some Useful Calibration Hints

1. *To make the policy response more (less) sluggish*, increase (decrease) the persistence in the policy rule ( $f_1$ ). Be aware, however, that by increasing (decreasing) persistence in the policy rule you *ceteris paribus* increase (decrease) the impact of monetary policy on the real economy (via both the real interest and exchange rates).
2. *To make the whole economy more persistent*, increase (decrease) the persistence parameter in the Phillips curve ( $b_1$ ). As a result, a more (less) pronounced policy reaction will be needed for a given disinflation goal.
3. *To force a faster (slower) convergence toward the trend ("steady-state") values*, increase (decrease) the policy pass-through either through the real monetary condition parameter

in the IS curve ( $a_2$ ) or the real marginal cost parameter in the Phillips curve ( $b_2$ ). As a result, monetary policy becomes more (less) powerful.

4. *To make the exchange rate path more (less) persistent or stable*, increase (decrease) the persistence (interventions) parameter in the UIP equation ( $e_1$ ).
5. *To make the transmission mechanism really sluggish*, you may introduce a lag for monetary conditions in the IS curve ( $a_2$ ) or a lag for the real marginal cost in the Phillips curve ( $b_2$ ) instead of the current values present in the canonical version. Similarly, to make the exchange rate pass-through really slow, introduce a lag for import prices in the Phillips curve.

You should exercise caution in increasing persistence in the economy. There exists a point, not easily defined in advance, when the economic agents may become too persistent for the central bank to be able to stabilize the economy. For example, if you make both the Phillips curve and the UIP fully backward looking, you should leave at least the policy rule to be forward looking, otherwise the model may become unstable.

#### **D. How to Check the Model Properties**

After the model has been calibrated, you may want to check visually the behavior of the model economy. You will do it by running the program ‘[modelproperties](#)’ ([Appendix V](#)) in the Matlab command window<sup>6</sup> and then inspecting six pages of printouts in a file ‘[Report\\_shocks.ps](#) or [.pdf](#) (PostScript of Acrobat, respectively):<sup>7</sup>

1. Aggregate demand shock;
2. Inflationary shock;
3. Exchange rate shock;
4. Monetary policy shock;
5. Foreign demand shock;
6. Foreign inflation shock.

Note that if your option is set for PostScript (.ps), you can (i) change the model calibration by ‘edit [readmodel](#)’; (ii) save it; (iii) run ‘[modelproperties](#)’ without having to close and open the ‘[Report\\_shocks.ps](#)’ file every time you run the ‘[modelproperties](#)’ program. The ‘[Report\\_shocks.ps](#)’ file changes with the change in model calibration while being open—a useful property when fine tuning the model.

If you decide to change the model structure, for instance by including the lagged output gap in the Phillips curve, change the code in the ‘[model.mod](#)’ file as follows:

---

<sup>6</sup> You can open the file by typing the ‘edit [modelproperties](#)’ [enter] in the command window.

<sup>7</sup> You can change the printout format in the very last line of the file ‘[modelproperties](#)’ by replacing .pdf (Acrobat) with .ps (PostScript): `compile(x, 'Report_Shocks.pdf')`.

```
%% Phillips Curve (Aggregate supply)
dl_cpi = b1*dl_cpi{-1} + (1-b1)*dl_cpi{+1} + b2*rmc{-1} + shk_dl_cpi;
```

Save the file and run the file ‘[modelproperties](#)’.

## VI. FORECASTING

This section discusses the use of the model for a basic forecasting exercise, the goal of which is to decide on an appropriate level of the policy rate. Rather than providing an unconstrained forecast of inflation, this class of models brings inflation close to the target by construction and the focus is on the path of the policy rate that would be consistent with such an inflation rate. The policy rate path depends on (i) the model structure, (ii) model parameterization, and (iii) initial conditions. We have discussed the model structure and parameterization in the preceding section and discuss below the importance of initial conditions.

### A. Database Preparation

A historical database, which provides initial conditions, is necessary to run the model forward. As a minimum, the database must contain a value for any variable that appears with a lag in the model structure. Thus, for a forecast starting in 2007Q1, we need initial conditions for 2006Q4 for all the variables with the lag of order one. In practice, however, the database is prepared and maintained using a longer time series data for all the relevant variables. For example, some of the variables needed for the initial conditions may be unobserved (output gap) and have to be estimated using the historical series. The longer the data set, the more precise is the estimation of unobserved variables.

The program ‘[makedata](#)’ is set to prepare the database for participant inspection without manual calculations of seasonal adjustment, de-trending or chart preparation. Nevertheless, before running ‘[makedata](#)’ in Matlab a basic data set needs to be compiled:

Table. Data Requirements

Series	Format
<i>Domestic economy</i>	
GDP at constant prices	Billions of national currency or an index
CPI	Index (not a rate of growth)
Nominal interest rate	Annualized rate, i.e., 5%
Nominal exchange rate	Units of domestic currency per one unit of foreign currency
<i>Foreign economy</i>	
GDP at constant prices	Billions of national currency or an index
CPI	Index (not a rate of growth)
Nominal interest rate	Annualized rate, i.e., 5%

These series—compiled in any program, including Microsoft Excel—need to be saved into the file ‘[data.csv](#)’ in the text format (‘[csv](#)’).<sup>8</sup> For more details see the template file ‘[data.csv](#)’. For your country series you have to use exactly the same names as in ‘[data.csv](#)’, otherwise the code would not recognize your series.

If any of your series is not seasonally adjusted, extend the series title with a suffix ‘\_u’ (for instance from ‘cpi’ to ‘cpi\_u’). The program ‘[makedata](#)’ will recognize seasonally unadjusted series and will run the seasonal adjustment (Census X12) automatically, saving the newly seasonally adjusted series without the suffix ‘\_u’. The program ‘[makedata](#)’ calculates the natural logarithm of all series (except interest rates), growth rates for relevant variables, and the real interest rate. It also estimates the trend and gap for real output, real exchange rate and domestic and foreign real interest rate. Here band-pass filter is used for the estimation of the output gap (domestic and foreign) and Hodrick-Prescott filter for the estimation of gap and trend for real exchange rate and domestic and foreign real interest rate.

Visual inspection of the data is a useful check of your prior intuition about the economy in question. Program ‘[makedata](#)’ creates panels of domestic and foreign inflation, nominal interest rates, exchange rates and estimated trends and gaps. It should be possible to recognize most mistakes or omissions in the ‘[data.csv](#)’ file just by inspecting the figures. We recommend checking especially the estimates of trends and gaps, as these will influence the forecast the most. The estimates of trends and gaps should correspond to your priors about the economy.

<sup>8</sup> In Excel, select Save As; in the “Save As” window click on “Save as type”; scroll down to CSV (MS-DOS); and select this option.

If the estimated trends or gaps seem biased, you may overwrite them manually (see below).<sup>9</sup> In fact, it suffices to change the last observation (initial condition) only, as only this information matters for the forecast. Note that with the exception of the year-on-year inflation rate the model contains no lags of order higher than one.

For example, the Hodrick-Prescott filter may generate an unreasonable trend for domestic real interest rates, while your priors suggest that this trend rate is in fact equal to, say, 1 percent annually. To manually change the database:

1. Open the program ‘`makedata`’ typing ‘`edit makedata`’ in Matlab command window [enter]
2. Go to the section ‘Expert change in the database’ (row 55) and type:  
`d.rr_eq(qq(2010,4)) = 1;`
3. Save the program ‘`makedata`’ and run it again typing ‘`makedata`’ in Matlab command window.

Your domestic trend real interest rate is now set to equal one in the fourth quarter of 2010. It is sufficient to change the last observation only as previous observations do not influence the forecast—the forecast depends only on the initial conditions.

## B. The Method of Kalman filtration

The final initial conditions (final database) may be based on the method of Kalman filtration rather than band-pass and Hodrick-Prescott filters. After running the program ‘`makedata`’ you may run program ‘`kalmanfilter`’. The program estimates all the unobserved variables, gaps and trends, based on the observed variables and the model structure, ensuring that the results are model- and calibration-dependent.

A reduced-form of the model serves as a starting point for the estimation of gaps and trends based on the method of Kalman filtration. The Kalman filter applies a reduced-form of the model extended for measurement equations that map observed variables to the unobserved. Together they represent the ‘state description of model’. The form is as follows:

$$y_t = Zx_t + \varepsilon_t \quad (1)$$

$$x_t = Tx_{t-1} + v_t \quad (2)$$

where  $x$  denotes the vector of unobserved state variables,  $y$  denotes the vector of observed (measurement) variables,  $\varepsilon$  is a random vector we call process noise, and  $v$  is the measurement noise. Furthermore, we assume a Gaussian distribution of the random vectors and of the  $x$ -state vector’s initial state.

Based on the state form of the model and using observed variables, the Kalman filter identifies all unobserved variables that are a part of the model, i.e. gaps, trends, and shocks.

---

<sup>9</sup> For example, the Hodrick-Prescott filter may be an inappropriate technique given the presence of a structural break in the series or because of end-period measurement error.



For linear systems it represents an optimum estimate in terms of the least squares criterion. Application of the filter itself takes on the recursive algorithm form, wherein the conditional probability density of state variables gets updated based on observed variables. Variables used as observed in estimation for all countries are set out in Table 5.

**Table 5: Observed variables for estimation**

CPI (level)	Foreign GDP (gap, band-pass filter)
GDP (level)	Foreign CPI (level)
Nominal exchange rate (level)	Foreign 3M interest rates (level)
3M nominal interest rates (level)	

The first step of the algorithm under the Kalman filtration is the prediction step. During the step, equation (2) is used to estimate the predictive probability density of states at time  $t$  based on the previous conditional probability density at time  $t-1$ . This probability density is however inexistent in the first period of the data sample and is therefore substituted with a random vector with a mean value and the unconditional variance of state variables described by the equation (2). Due to the presence of trends within the model and resulting non-stationarity of certain variables, the unconditional variance does not have a finite value and a diffuse Kalman filter need to be applied.

The filtration step follows after the prediction step, representing an update of the predictive probability density based on the information contained in the observed data. The measurement equation (1) is used for that purpose. Additional information drawn from the observed data enables a refined estimate of the state variables, also including the estimate of shocks. In addition to the above Kalman filter steps, we use also a smoothing step of the filter which, as opposed to the prediction and filtration steps, uses complete information from the observed data.

Results are saved in a graphical form in to files ‘[Filter.ps](#)’ and ‘[Filter.pdf](#)’ and as data in to files ‘[kalm\\_his.csv](#)’ and ‘[kalm\\_his.mat](#)’. The latter serves as your forecasting database.

### C. The Forecast

Mechanically, the making of the forecast is the shortest part of the whole process, however, its quality depends on previous work (calibration and data preparation) and ex post analysis. The forecast is produced by running the program ‘[forecast](#)’. This program reads the model (‘[\[m,p,mss\] = readmodel](#)’) and historical data (initial conditions) from the database and simulates the model forward. Furthermore, the program produces indicator figures and tables that can be viewed by doubleclicking on the file ‘[Forecast.ps](#)’ (or ‘[Forecast.pdf](#)’). As the data source you may use either the database based on statistical filtration (‘[history.csv](#)’) or method of Kalman filtration (‘[kalm\\_his.csv](#)’). You switch between those two databases by changing the name of the respective file in command ‘[h = dbload\('...'\)](#)’.

Reminder: You need to run ‘[makedata](#)’ and ‘[kalmanfilter](#)’ any time you change your data before you run the program ‘[forecast](#)’.

## The Programming Code of ‘forecast’

The following section briefly summarizes the code of the program ‘forecast’ and can be skipped by most readers who are not interested in Matlab and IRIS programming. The program starts with reading the model and data into the objects ‘m’ and ‘h’ respectively and sets the forecast range. You can change the range depending on your current data sample. The key command for running the forecast is `simulate` with the following syntax:

```
S = simulate(m,h,fcastrange)
```

The simulation results are saved into the database object ‘s’ and the command ‘`dboverlay`’ adds ‘h’ to ‘s’, overlaying the historical data with the current projections. The whole database is saved into the file ‘`fcastdata.mat`’ (by the command `savestruct`) and the figure and tables are saved into the ‘`Forecast.ps`’ file (or ‘`Forecast.pdf`’).

### How to Pre-Set Future Values

You may want to set one or more future variables to an exact number, either because you do not believe that the model’s AR(1) process forecasts a particular variable well (for example, some of the foreign variables) or because some near-term variables are better forecast by time-series models (for example, inflation one period ahead). You save the dataset ‘`data.csv`’ with these future values and tell the program ‘forecast’ to use these values using the command ‘`plan`’ with the following syntax:

```
simplan = plan(m,fcastrange)
```

The command `plan` creates an object ‘`simplan`’ that initially contains an empty plan for model ‘m’ over the forecasting range. In the next step you specify which variable are to be taken from ‘`data.csv`’ using commands `exogenize` and `endogenize`. Assuming we want to pre-set the external rate of inflation and interest rates, the syntax is as follows:

```
simplan = exogenize(simplan,{'dl_x_cpi','x_rn'},fcastrange)
```

The command `exogenize` specifies which variables are going to be taken as exogenous variables and decides over which horizon this will hold (in this case over the whole forecast horizon; if you want to pre-set these variables only for the next two quarters, replace `fcastrange` with `qq(2007,1):qq(2007,2)`, and so on). Pre-setting a variable causes, however, a discrepancy between the left-hand side of the respective equations (the pre-set values) and their right-hand sides (the data generating processes). Both sides of the equation are equalized through the residual term, which now becomes an endogenous variable, see `endogenize`:

```
Simplan = endogenize(simplan,{'shk_dl_x_cpi','shk_x_rn'},fcastrange)
```

Intuitively, you can pre-set only those variables that have a residual term in their equations specified in `model.mod`.

To summarize, if you decide to pre-set the two variables above, in the program ‘forecast’ put a percentage sign ‘%’ in front of `S = simulate(m,h,fcastrange)` and remove the percentage signs from the following four lines:

```
simplan = plan(m,fcastrange);
simplan = exogenize(simplan,{'dl_x_cpi','x_rn'},fcastrange);
```

```

simplan = endogenize(simplan, {'shk_dl_x_cpi', 'shk_x_rn'}, fcastrange);
s = simulate(m, h, fcastrange, 'plan', simplan);

```

#### D. Interpreting the Forecast

The simulations of your model need to be carefully interpreted and this section suggests some steps toward this objective. All relevant results are contained in file ‘[Forecast.ps](#)’ (or ‘[Forecast.pdf](#)’) in the form of a figure and two tables.

Note that the inflation forecast converges to the targeted inflation, possibly with some initial fluctuation. This is a normal property of the so-called unconditional forecast where the monetary authority does all it can to keep the inflation rate close to the targeted one within the transmission period (usually 6-8 quarters). The most interesting variable is thus the trajectory of nominal interest rate set by the central bank rather than the forecasted inflation rate. The simulated path of the policy interest rate provides a policy reaction given the past data and the model structure.

In addition to inflation and the nominal interest rate, the figure shows the forecast of the nominal exchange rate, output gap, and monetary conditions. The last chart—the real interest and exchange rate gaps—plots both the direct and indirect exchange rate channels and the interest channel. Based on this chart, one can observe how tight the monetary stance must be (the real interest and exchange rate gaps) in order to meet the inflation target and what would be the cost of this stance in terms of real economic activity (the output gap).

The forecast printout also contains two tables: Main Indicators and Decomposition. The former table is self-explanatory and all variables correspond directly to the variables discussed earlier. The latter table provides a breakdown of the contributing factors to your forecasts based on the relevant equations. For example, the aggregate demand (the IS curve) is a function of the past output gap, real monetary conditions, and external demand and the second block of the forecast decomposition table provides this information (Output Gap Decomposition). Similarly, the Phillips curve breakdown is provided in the third block (Inflation Factor Decomposition). You should be able to extract an economic story from this table to back up your inflation scenario.

## Software Installation

The Matlab software does not provide a specific toolbox for economic modeling and serves only as the environment for the necessary matrix algebra. For the economic modeling it is necessary to install the IRIS toolbox, which is a Matlab-based package for advanced macroeconomic modeling. Although the current IRIS toolbox can be downloaded from [www.iris-toolbox.com](http://www.iris-toolbox.com), we provide you with a package that contains the version of the IRIS that is used for all presented simulations. The package can be downloaded from [www.ales-bulir.wbs.cz](http://www.ales-bulir.wbs.cz). The installation of IRIS toolbox requires copying two files from the package in a specific directory (recommended: `c:/Modeling/iris_install`):

1. the zip file ‘`iris-toolbox.zip`’;
2. the Matlab file ‘`installiris.m`’

Note that both files must be copied exactly in this form. Before you start IRIS installation it is useful to install on your computer X12 Census package for seasonal adjustment and the MiKTeX, Ghostscript and Ghostgum (GSView) packages for reporting. No previous knowledge of X12, MiKTeX or Ghostscript - Ghostgum is required. The IRIS Toolbox uses all of them automatically.

We recommend creating a folder `c:/Modeling` that contains the following folders:

1. FOLDER ‘`iris-install`’, i.e., `c:/Modeling/iris-install`, with ‘`iris-toolbox.zip`’ and ‘`installiris.m`’ files.
2. FOLDER ‘`FS`’ (free software), i.e., `c:/Modeling/FS`, where the installation folder for MiKTeX, and files for Ghostscript and Ghostgum are saved. Copy all the installation files from the downloaded package. If interested, you can download the current versions of the MiKTeX Installer at [www.miktex.org](http://www.miktex.org) and [www.cs.wisc.edu/~ghost/doc/AFPL/get853.htm](http://www.cs.wisc.edu/~ghost/doc/AFPL/get853.htm) and <http://www.cs.wisc.edu/~ghost/gsview/> for Ghostscript and Ghostgum.
3. FOLDER ‘`x12`’, i.e., `c:/Modeling/x12`, where the installation file for X12 is copied. After executing the ‘`omegaxp.exe`’ file, the X12 program is installed automatically in the same folder. The current version can be downloaded from [http://www.census.gov/srd/www/x12a/x12down\\_pc.html#x12progXP](http://www.census.gov/srd/www/x12a/x12down_pc.html#x12progXP) (‘`omegaxp.exe`’).
4. FOLDER ‘`iris-toolbox`’, i.e., `c:/Modeling/iris-toolbox`, where the IRIS Toolbox is installed.
5. FOLDER ‘`texmf`’, i.e., `c:/Modeling/texmf`, where the MiKTeX is installed.

### *X12 Installation*

We start with the X12 Census installation:

1. Go to the `c:/Modeling/x12`;
2. Execute the downloaded ‘`omegaxp.exe`’ file. X12 Census is automatically installed in the same folder.

### *MiKTeX Installation*

1. Go to the [c:/Modeling/FS/MikTeX](#);
2. Execute the downloaded file '[setup-2.4.1705.exe](#)' file.
3. When asked for the installation folder, do not install the MiKTeX into the automatically offered 'Program Files', but change the folder via browse option to [c:/Modeling/texmf](#);
4. Install.

### ***Ghostscript and Ghostgum Installation***

1. Go to the [c:/Modeling/FS](#);
2. Execute the respective '.exe' files (their names may vary with the version you downloaded).
3. Both programs can be installed into the Program Files.

### ***IRIS Installation***

1. For this we start Matlab first.
2. Set the 'current directory' (on the top and in the middle of the screen) for the directory where the IRIS installation files are saved, i.e., [c:/Modeling/iris\\_install](#);
3. Write the command '[installiris](#)' in the Matlab's command window, [enter].
4. You will be asked for the desired IRIS Toolbox folder. Type [c:/Modeling/iris-toolbox](#), [enter]. IRIS Toolbox is installed in this folder. Matlab is case sensitive and you must write your folder names exactly.
5. You will be asked for the path to the LaTeX (MiKTeX) program (installed in the previous step). Type [c:/Modeling/texmf/miktex/bin](#), [enter].
6. You will be asked for the path to the X12. Type [c:/Modeling/x12](#), [enter].
7. Now the IRIS Toolbox has been installed and the paths to MikTeX and X12 have been set.

### ***Starting the IRIS Toolbox***

Anytime you open Matlab for a modeling exercise you must path the IRIS Toolbox to the Matlab. This must be done via Matlab's command window typing:

[addpath c:/Modeling/iris-toolbox; irisstartup](#), [enter].<sup>10</sup> You receive the message '[The IRIS Toolbox Version 4. 20061115 ready](#)'. Note that you should not path the IRIS Toolbox via Set Path option in the Matlab's File menu.

### ***Model Files***

Next step is to create subfolder 'Model' in your 'Modeling' folder, i.e., [c:/Modeling/Model](#). Unzip the file '[model.zip](#)' that is a part of the downloaded package. Once unzipped the folder 'Model' should contain the following files:

---

<sup>10</sup> Matlab will remember the command and next time you can type 'a' and push arrow up, Matlab will bring up this command.

1. model.mod
2. readmodel.m
3. modelproperties.m
4. makedata.m
5. forecast.m
6. data.csv

We describe these files in the section Modeling.

With the exception of `model.mod` and `data.csv`, all files are Matlab programs (extension `.m`) and work in the Matlab environment only. Before running them the Matlab's 'current directory' must be switched to the folder `Model` where the programs are saved. Thus switch the current directory from `c:/Modeling/iris_install` to `c:/Modeling/Model` (`cd c:/Modeling/Model`, [enter]). You may check that you are in the right directory typing the command `ls`, [enter]. Matlab will give you a list of files saved in its current directory.

Now you can execute the programs `readmodel`, `modelproperties`, `makedata` and `forecast`. The programs get started in the following way. Type the name of the program, for instance `readmodel` in the command window and push [enter]. Once the screen shows `>>` the program has been run successfully. Note that if you run the programs for the first time you should follow sequence: `readmodel`, `modelproperties`, `makedata`, `forecast`. Be aware that while running the program `modelproperties` for the first time, the MiKTeX application may ask you twice for additional installation. Choose yes as the installation uses the package installed previously and calls for some sub-packages that were installed automatically.

Please check that all is running:

1. Execute `readmodel`, `modelproperties`, `makedata` and `forecast`,
2. Start Windows explorer,
3. From `c:/Modeling/Model` double click on files `Report_Shocks.ps`, `Stylized_facts.ps`, `Forecast.ps`.
4. [OK] and you should see charts and tables of various model outputs.

## The Model Code

For the model code see the file ‘[model.mod](#)’ and you may open it in the Matlab’s editor typing the command ‘`edit model.mod`’ in the command window.<sup>11</sup>

The model code has four basic parts: endogenous variables, equation residuals, parameters, and equations:

1. **@variables:transition** The list of all endogenous (transition) variables.
2. **@variables:innovation** The list of all residuals defined in individual equations.
3. **@parameters** The list of parameters. Note that this list must be identical to the parameters defined within the ‘[readmodel](#)’ program—any parameter change in ‘[model.mod](#)’ must be reflected in ‘[readmodel](#)’, and vice versa.
4. **@equations:transition** The list of equations defining the endogenous (transition) variables. The number of transition variables must equal the number of transition equations.

The program ‘[readmodel](#)’ (as a Matlab file it has an extension ‘.m’) takes the code written in the ‘[model.mod](#)’, assigns the parameters values, and solves the model for its reduced form. To modify model parameters and steady-state values you need to open the ‘[readmodel](#)’ in the Matlab editor typing the command ‘`edit readmodel`’ in the Matlab command window or using the File menu. The IRIS commands used for handling the model are as follows:

The first command is [model](#):

```
m = model('model.mod', 'linear', true, 'assign', p);
```

That reads the text file ‘[model.mod](#)’ containing the model code and assigns the model parameters and trend values preset in the database ‘[p](#)’ to the model. In addition it transforms the model for the matrix algebra. Transformed model is written in the object ‘[m](#)’. The option ‘[linear](#)’ is logical function. If your model is linear you assign the value true. This makes the transformations and calculations easier and saves time.

Second, the model is solved for its steady-state using the command [sstate](#):

```
m = sstate(m);
```

The command ‘[sstate](#)’ takes the transformed model written in object ‘[m](#)’ and calculates its steady-state. This is done via iterations using the Matlab’s optimization toolbox. The steady-state values are written in object ‘[mss](#)’ that may be viewed by typing the ‘[mss](#)’ [enter] in the Matlab command window. It is useful to check the steady-state values any time you change the model structure (see section Calibration for further details).

Finally, the model is transformed in its reduced form using the command [solve](#). Exact quotation looks as follows:

---

<sup>11</sup> Anything that is written after % signs is a comment (or an inactive portion of the code) and does not constitute a part of the model.

```
m = solve(m);
```

For the transformation the Blanchard-Kahn algorithm is used. The object 'm' contains the reduced form of the model together with the steady-state values.

Using the command `savestruct` the object 'm' is saved in the database file 'model.mat'. This is used later for shock analysis and forecasting. You may run the 'readmodel' simply by typing: `[m,p,mss] = readmodel` [enter] in the Matlab command window. The evidence that the 'readmodel' works properly is either the appearance of file 'model.mat' in your current directory (if it was not there before) and you see the list of steady-state values in the Matlab command window once the 'readmodel' has been run. You do not need run the 'readmodel', however, as the other programs call the 'readmodel' as a function jointly with the model and its parameterization and steady-state as the outcome.