MATLAB routines for trend agnostic one step estimation of a small New Keynesian model

Filippo Ferroni*

December 30, 2008

1 Introduction

This document is aimed to provide a guide for the MATLAB routines for the estimation of a small New Keynesian model using the trend agnostic one step approach. Reference and details of the estimation procedure can be found in the paper *Trend agnostic one step estimation of DSGE models*. I wrote the routines with MATLAB version R2006a.

2 MATLAB files

The folder `MATLAB_routines` contains several .m files and two folder, `database` and `POSTERIORS`. `database` contains the data used for the estimation. The times series are GDP, hours worked, real wages and inflations from 01-04-1964 to 01-04-2007. `POSTERIORS` collects files, tables and graphs for the posterior analysis.

`MAIN.m` is the main programm, which calls the different econometric procedures to estimate the structural parameters of the DSGE model. The program loads the data and calls the RWM algorithm to compute the posterior distributions under different settings. Data is loaded from the folder `database`. For the two step setup, data is first filtered and stored in a $(174 \times 4 \times 3)$ matrix, `DATA`, where the third dimension refers to the filter. So, the matrix `DATA(:,:,1)` contains linear detrended data, `DATA(:,:,2)` the HP filtered data, `DATA(:,:,3)` the first difference of the data. For the one step, data is raw, `data`. I consider

*Department of Economics, Universitat Pompeu Fabra, Carrer Ramón Trias Fargas, 25-27, 08005 Barcelona, Spain. Office: 20.159, Jaume I building. Phone: 0034 93 542 2689. e-mail: filippo.ferroni@upf.edu.

two types of estimation approaches: two step, where data is first filtered and then DSGE structural parameters are estimated, and one step, where structural and filtering parameters are jointly estimated. For each approach I consider three filters: linear detrending, HP filter and first difference. Thus, we have 6 setups. For each setup, parameters are estimated using Monte Carlo Markov Chain simulators, in particular using a RWM algorithm (see below). The output is saved in a `.mat` file according to the econometric approach (one step, `1s`, or two step, `2s`) and to type of filter (linear detrending, `lt`, HP filter, `hp`, and first difference, `fd`); for example, `1s_output_hp.mat` means that we estimate the parameters with the one step method using the HP filter. Output is stored in the folder `.\POSTERIORS\output`.

`RWM.m` is the Random Walk Metropolis-Hastings algorithm. Details on the algorithm can be found in the paper. The inputs are: the set of times series, `dy`, an initial value for the parameters, `theta`, a value for the variance matrix of the algorithm, `VAR`, the type of estimate, where two step = `2s`, hp-dsge = `hp`, lt-dsge = `lt` and fd-dsge = `fd`. The main output of the estimations is a vector, called `theta_p`, which contains all the accepted draws and whose dimension depends on the number of iterations, `N`. Notice that the value for `N` is set in `MAIN.m`. A 'safe' number of repetition is 1 million draws; that ensures convergence for all the parameters. It takes about 4-5 hours for each setup depending on the RAM of your computer. If you want to see how the program works, 200,000 draws are fine, but you might need to change some stuff in the files that compute the posterior distributions and statistics. Other outputs are `loglike_p` and `logprior_p`; they are vectors that collects the values of the likelihood and of the prior computed at each raw elements of `theta_p`, respectively. Finally, `acc` is the frequency by which we accept the draws; we tried to keep it around 20-35%.

`initialvalues.m` is a function that loads the initial value for the parameters to start the RWM algorithm. Clearly, the dimension of the vector of parameters depends on the type of estimation. Whereas for the two step the dimensionality of the parameter vector is the same regardless of the filter used, in a one step approach the number of parameters depends on the filter used. In particular, in a two step approach the parameters vector contains the DSGE structural parameters, in the hp-dsge setup the parameters vector contains the DSGE structural parameters and four filter standard deviations, in the lt-dsge setup the parameters vector contains the DSGE structural parameters, four standard deviations and four slopes[1], in the fd-dsge setup the parameters vector contains the DSGE structural pa-

[1] I do not estimate the intercept because they do not converge. I fix them to the first observation of the

rameters, four standard deviations and four constants. In the lt-dsge setup, the initial value for the slope of the linear trend is the OLS estimator. In the fd-dsge setup, the initial value for the constant of the unit root specification is given by the mean of the first difference data.

`model.m` rewrites the DSGE linearized equilibrium conditions in matrix form and in a way that suits the Uhlig (1998) algorithm, `solve.m`. The output is the solution of the DSGE linearized model in form of matrices, `PP,QQ,RR,SS,NN`. If there is no stable solution, then `PROBLEM=0` and the solution matrices `PP,QQ,RR,SS,NN` are set to zero. The DSGE model equilibrium conditions are

$$\begin{aligned}
\lambda_t &= \epsilon_t^{\chi} - \sigma_c y_t \\
y_t &= \epsilon_t^a + n_t \\
mc_t &= \omega_t + n_t - y_t \\
mrs_t &= -\lambda_t + \sigma_n n_t \\
\omega_t &= mrs_t \\
r_t &= \rho_r r_{t-1} + (1-\rho_r)(\rho_\pi \pi_t + \rho_y y_t) + \epsilon_t^r \\
\lambda_t &= E_t(\lambda_{t+1} + r_t - \pi_{t+1}) \\
\pi_t &= k_p(mc_t + \epsilon_t^{\mu}) + \beta E_t \pi_{t+1} \\
\epsilon_t^{\chi} &= \rho_\chi \epsilon_{t-1}^{\chi} + \nu_t^{\chi} \\
\epsilon_t^a &= \rho_a \epsilon_{t-1}^{\chi} + \nu_t^a
\end{aligned}$$

λ_t is the marginal utility of consumption and σ_c elasticity of intertemporal substitution. The shadow value of consumption is hit also by a preference shock, ϵ_t^{χ}, which I assume to follow an AR(1) process. Total factor productivity, ϵ_t^a, is assumed to be a stationary AR(1) process. The difference between real wage, ω_t, and the marginal product of labor, $y_t - n_t$, defines the marginal cost, mc_t. Since labor market is perfectly competitive and frictionless, there is no wage markup and the marginal rate of substitution, mrs_t, is equal to the real wage. The marginal rate of substitution between working and consumption depends positively on hours worked, where σ_n is the inverse of the Frish elasticity of labor supply. β is the time discount factor. The NKP curve is hit by a cost push shock, ϵ_t^{μ}. The slope of the Phillips curve is $k_p = (1-\zeta_p)\frac{1-\beta\zeta_p}{\zeta_p}$, where ζ_p is the probability of keeping the price fixed. The four exogenous processes are driven by mutually uncorrelated, zero mean innovations, i.e. $\nu_t = [\nu_t^{\chi}, \nu_t^a, \nu_t^r, \nu_t^{\mu}]$.

data.

I define x_t as the vector of endogenous states,

$$x_t = [\lambda_t, mc_t, mrs_t, r_t].$$

y_t as the endogenous variables vector,

$$y_t = [y_t, n_t, \omega_t, \pi_t]$$

z_t as the vectors of exogenous processes,

$$z_t = [\epsilon_t^{\chi}, \epsilon_t^{a}, \epsilon_t^{r}, \epsilon_t^{\mu}]$$

`model.m` rewrites the above system of equation such that

$$\begin{aligned} 0 &= AAx_t + BBx_{t-1} + CCy_t + DDz_t \\ 0 &= E_t[FFx_{t+1} + GGx_t + HHx_{t-1} + JJy_{t+1} + KKy_t + LLz_{t+1} + MMz_t] \\ z_{t+1} &= NNz_t + \nu_{t+1} \end{aligned}$$

where $E_t[\nu_{t+1}] = 0$. Then, `solve.m` (joint with `calc_qst.m`, `qzdiv.m`, `qzswitch.m` and `solve_qz.m`) solves for the equilibrium law of motion

$$\begin{aligned} x_t &= PPx_{t-1} + QQz_t \\ y_t &= RRx_{t-1} + SSz_t \end{aligned}$$

`like.m` computes the likelihood of the data, `dy`, with the Kalman filter given a value for the parameters vector, `theta`. `like.m` first calls a function, `model.m`, which solves the DSGE and gives the equilibrium law of motion of the economy, the matrices `PP,QQ,RR,SS,NN`. In case there does not exist a stable solution to the DSGE model the likelihood is set to $-\infty$. Otherwise, it computes the likelihood of a linear state space system, i.e.

$$s_{t+1} = Fs_t + G\omega_{t+1} \tag{1}$$

$$Y_t = Hs_t + \eta_t \tag{2}$$

where $\eta_t \sim N(0, R)$ and $\omega_{t+1} \sim N(0, \Sigma)$.
In a two step approach (regardless of the filter), `2s`, the state space system is defined as

follows

$$
\begin{aligned}
Y_t &= \mathbf{DATA}(t,:,i)' \\
s_t &= \begin{pmatrix} x_{t-1} & z_t \end{pmatrix} \\
F &= \begin{pmatrix} PP & QQ \\ \mathbf{0} & NN \end{pmatrix} \\
G &= \begin{pmatrix} \mathbf{0} & I \end{pmatrix}' \\
H &= \begin{pmatrix} RR & SS \end{pmatrix}
\end{aligned}
$$

The variance covariance matrix, R, of the measurement equation, (2), is set to zero. The variance matrix, Σ, of the transition equation, (1), is a diagonal matrix with entries the standard deviations of the DSGE model structural shock.
In a one step approach with a linear detrending filter, `lt`, the state space is defined as follows

$$
\begin{aligned}
Y_t &= \mathbf{data}(t,:)' - a - bt \\
s_t &= \begin{pmatrix} x_{t-1} & z_t \end{pmatrix} \\
F &= \begin{pmatrix} PP & QQ \\ \mathbf{0} & NN \end{pmatrix} \\
G &= \begin{pmatrix} \mathbf{0} & I \end{pmatrix}' \\
H &= \begin{pmatrix} RR & SS \end{pmatrix}
\end{aligned}
$$

I fix the intercept a because it is difficult to estimate; in particular, the intercept does not converge using the CUMSUM diagnostic. I use the first data observation as value for the constant, i.e. $a = \mathbf{data}(1,:)'$. I tried also other values as the mean value of the data or $1/2 * (\mathbf{data}(T,:) - \mathbf{data}(1,:))$, but the specification with the first observation fits the data better. The variance covariance matrix, R, of the measurement equation is given by the standard deviations of the linear trend.

In a one step approach with a first difference filter, `fd`, the state space is defined as follows

$$
\begin{aligned}
Y_t &= \mathbf{data}(t,:)' - \gamma - \mathbf{data}(t-1,:)' \\
s_t &= \begin{pmatrix} x_{t-1} & z_t \end{pmatrix} \\
F &= \begin{pmatrix} PP & QQ \\ \mathbf{0} & NN \end{pmatrix} \\
G &= \begin{pmatrix} \mathbf{0} & I \end{pmatrix}' \\
H &= \begin{pmatrix} RR & SS \end{pmatrix}
\end{aligned}
$$

The variance covariance matrix, R, of the measurement equation is given by the standard deviations of the unit root.

Finally, in a one step approach with the HP filter, `hp`, the state space is defined as follows

$$
\begin{aligned}
Y_t &= \mathbf{data}(t,:)' \\
s_t &= \begin{pmatrix} y_t^t & \mu_t & x_{t-1} & z_t \end{pmatrix} \\
\omega_{t+1} &= \begin{pmatrix} \zeta_{t+1} & \epsilon_{t+1} \end{pmatrix} \\
F &= \begin{pmatrix} I & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & PP & QQ \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & NN \end{pmatrix} \\
G &= \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix} \\
H &= \begin{pmatrix} I & \mathbf{0} & RR & SS \end{pmatrix}
\end{aligned}
$$

The variance covariance matrix, R, of the measurement equation, (2), is set to zero. The variance matrix of the transition equation, Σ, is a diagonal matrix with entries the standard deviations of the stochastic I(2) trend and standard deviations of the DSGE model structural shock.

`prior.m` computes the (log) value of the prior distribution evaluated at a certain value for the parameters, `theta`, for a particular type of estimation, two step, `2s`, hp-dsge, `hp`, lt-dsge, `lt` and fd-dsge, `fd`. Prior selection is quite standard; I assumed Beta distribution for those parameters that must lie in the 0-1 interval, like $\rho_R, \zeta_p, \rho_\chi, \rho_a$. I choose a prior mean close to 0.5 for the probability of keeping the prices fixed, whereas the autoregressive parameters in the exogenous processes have prior mean close to 0.7. I employ Gamma or Inverse Gamma distributions for the parameters that must be positive, like the elasticity of consumption and leisure (σ_c and σ_n). For the standard deviations, I use Inverse Gamma with mean close to 0.006 and standard deviation of 0.002. The remaining parameters have normal distributions.

`var_rwm.m` sets the variance of the RWM algorithm. The input are the data, `data`, scaling parameter, `k`, which is set in `MAIN.m`, and type of estimation, two step, `2s`, hp-dsge, `hp`, lt-dsge, `lt` and fd-dsge, `fd`.

The folder `POSTERIORS` contains MATLAB files and other folders aimed to produce and collect posterior statistic and distributions. So, one needs first to run the main program and then posterior statistics can be computed. These statistics are computed as if the number of iterations is one million. For smaller number of iterations one should revise the 'burn-in' draws (`burnin`) and the way in which the draws are picked to compute the posterior distributions (`ELLE`) in `posterior.m` .

`main_posterior.m` and `posterior.m` produce graphs, which are stored in the folder `graphs`. For a given setup `posterior.m` first plots the log of the likelihood across draws. The second series of graphs are the CUMSUM plots. After having removed the first 3/10 of the chain, the program computes

$$CUMSUM_j = \sum_{k=1}^{j} \frac{\theta_k - \mu_\theta}{\sigma_\theta}$$

where $j = 1, ..., N_0$ and $N_0 = 0.7N$, and $\mu_\theta = \frac{1}{N_0}\sum_{i=1}^{N_0} \theta_i$ and $\sigma_\theta = \frac{1}{N_0-1}\sum_{i=1}^{N_0}(\theta_i - \mu_\theta)^2$. If the graphs settle after an initial period, then the chain has converged.
With one million draws, we get convergence for all the parameters after 500'000 draws, thus to compute the posterior distributions we use the second half of the chain. We pick randomly one every 1000 draws to remove the correlation among draws induced by the Markov Chain. Posterior distributions are computed using kernel density methods. All the plots are saved in the folder `graphs`. `main_posterior.m` run `posterior.m` for each setting. As mentioned, I consider two types of estimation approaches: two step, where data are first filtered and then DSGE structural parameters are estimated, and one step, where structural and filtering pa-

rameters are jointly estimated. For each approach I consider three filters: linear detrending, HP filter and first difference. Thus, we have 6 setups. Posterior graphs (likelihood, convergence and prior-posterior distribution) are saved in `.\POSTEIRORS\graphs\` according to the econometric approach (one step, `1s`, or two step, `2s`) and to type of filter (linear detrending, `lt`, HP filter, `hp`, and first difference, `fd`); for example, `Convergence_S_1s_hp.eps` means that the Convergence plots for the structural parameters are computed with the estimates of the one step method using the HP filter. Default format for graphs is `.eps`, but this option can be changed in `posterior.m`.

`POdds.m` computes the Posterior Odds of different specifications. The Posterior Odds (PO) is the ratio of predictive density of the data conditional on the DSGE model and a filter specification with respect to the linear detrending one. The predictive density of the data conditional on the DSGE model and on the filter specification is computed by taking an average of the sum of the log likelihood and log prior for all the accepted draws. The program computes the PO and creates a table for LaTeX, which is saved in a .txt file in `.\POSTEIOR\table`.

`tables.m` computes the posterior statistics for each setup. I consider two types of estimation approaches: two step, where data are first filtered and then DSGE structural parameters are estimated, and one step, where structural and filtering parameters are jointly estimated. For each approach I consider three filters: linear detrending, HP filter and first difference. Thus, we have 6 setups. Posterior statistics are saved in a `.mat` file according to the econometric approach (one step, `1s`, or two step, `2s`) and to type of filter (linear detrending, `lt`, HP filter, `hp`, and first difference, `fd`); for example, `stat_1s_hp.mat` means that the posterior statistics are computed with the estimates of the one step method using the HP filter. Output is stored in the folder `.\POSTERIORS\table`. Finally, in `.\POSTERIORS\table` there is a file, `tables_tex.m` that prepares the the tables for LaTeX.

`apiter.m` reduces randomly the dimensionality of `theta`. This program throws the `LL` rows of the matrix `theta`, and then picks randomly one row every `elle` rows. This program basically is aimed to remove the correlation between draws created by the Markov Chain of the RWM algorithm.