

Bayesian Macroeconometrics in R

Keith O'Hara*

July 20, 2015

Version: 0.5.0

*Department of Economics, New York University. My thanks to Alan Fernihough and Karl Whelan for their helpful comments and suggestions.

Contents

1. Introduction	3
1.1. Bayesian Econometrics	4
2. Introduction to Bayesian VARs	6
2.1. Gibbs Samplers	9
2.2. Normal-inverse-Wishart Prior	10
2.3. Minnesota Prior	13
3. Steady State BVAR	15
4. Main BVAR Functions	17
4.1. Normal-inverse-Wishart Prior	17
4.2. Minnesota Prior	19
4.3. Steady State Prior	21
5. Additional Functions	24
5.1. Classical VAR	24
5.2. GACF and GPACF	26
5.3. Time-Series Plot	27
5.4. Stationarity	28
6. BVAR Examples	30
6.1. Monte Carlo Study	30
6.2. Monetary Policy VAR	35
6.3. Steady State Example	47
7. Time-Varying Parameters	49
7.1. BVARTVP Function	53
7.2. BVARTVP Example	57
8. DSGE	67
8.1. Solution Methods	69
8.1.1. Uhlig's Method	69
8.1.2. gensys	74

8.2. Estimation	75
8.2.1. The Kalman Filter	76
8.2.2. Chandrasekhar Recursions	78
8.3. Prior Distributions	79
8.4. Posterior Sampling	84
8.5. Marginal Likelihood	85
8.6. Summary	85
8.7. DSGE Functions	88
8.7.1. Solve DSGE (gensys, uhlig, SDSGE)	88
8.7.2. Simulate DSGE (DSGESim)	90
8.7.3. Estimate DSGE (EDSGE)	91
8.7.4. Prior Specification (prior)	94
9. DSGE Examples	96
9.1. RBC Model	96
9.1.1. Log-Linearising the RBC Model	100
9.1.2. Inputting the Model to BMR	108
9.1.3. Estimation	112
9.2. The Basic New-Keynesian Model	118
9.2.1. Dixit-Stiglitz Aggregators	118
9.2.2. The Household	122
9.2.3. Marginal Cost and Returns to Scale	123
9.2.4. The Firm	125
9.2.5. Inputting the Model to BMR	133
9.2.6. Estimation	138
9.3. A Rather More Complicated Affair	147
9.3.1. Steady State	155
9.3.2. Log-linearized Model	158
10.DSGE-VAR	163
10.1.Details	163
10.2.Estimate DSGE-VAR (DSGEVAR)	166
10.3.Example	170

A Calculating Impulse Response Functions for BVARs	182
B Specifying Full Matrix Priors in BMR	185
B1. Prior Mean of the Coefficients	185
B2. Prior Covariance Matrix	187
C Class-related Functions	188
C1. Forecast	188
C1.1. BVARM, BVARS, BVARW, DSGEVAR	188
C1.2. CVAR	189
C1.3. EDSGE	190
C2. IRF	191
C2.1. BVARM, BVARS, BVARW, CVAR	191
C2.2. BVARTVP	192
C2.3. DSGEVAR	192
C2.4. EDSGE	192
C2.5. SDSGE	193
C3. Mode Check	194
C3.1. DSGEVAR, EDSGE	194
C4. Plot	195
C4.1. BVARM, BVARS, BVARW	195
C4.2. BVARTVP	196
C4.3. DSGEVAR, EDSGE	196
D Example of a Parameter to Matrices Function (partomats)	197

1. Introduction

Bayesian Macroeconomics in **R** ('BMR') is a collection of **R** and C++ routines for estimating Bayesian Vector Autoregressive (BVAR) and Dynamic Stochastic General Equilibrium (DSGE) models in the **R** statistical environment. Designed to be a flexible and self-contained resource for these methods, the BMR package incorporates several popular prior forms for BVAR models, including models with time-varying parameters, and utilizes the Armadillo C++ linear algebra library—linked with **R** via the RCPPARMADILLO interface of Francois et al. (2012)—to run computationally expensive Bayesian simulation algorithms.

The BMR project was inspired by several existing software libraries. First, Koop and Korobilis (2010), who provided extensive Matlab routines for estimating BVAR models; Adjemian et al. (2011)'s DYNARE, a set of Matlab and Octave routines for estimating DSGE models; and YADA, written by the DSGE team at the European Central Bank and maintained by Warne (2012). The BMR package brings these popular macroeconometric tools to **R**, and all graphs produced by BMR use Wickham (2009)'s excellent GGPLOT2 package.

This accompanying document begins with a brief introduction to Bayesian econometrics and Bayesian Vector Autoregression models. We first derive a useful likelihood factorisation of the classical VAR model that will be used throughout this document. We then move to the problem of simulating from the posterior distribution of BVAR parameters, with three different prior forms discussed: the normal-inverse-Wishart prior, the so-called Minnesota prior in the spirit of Doan et al. (1983) and Litterman (1986), and the steady-state prior of Villani (2009). After outlining the input syntax to the BMR BVAR functions, several brief examples are given to illustrate the code-at-work. The final BVAR model we consider is one with time-varying parameters.

The second-half of the document is dedicated to the solution, simulation, and estimation of DSGE models. The solution methods available in BMR include C++ implementations of Uhlig (1999)'s method of undetermined coefficients and Chris Sims' gensys solver. After describing the solvers, we turn to Bayesian estimation using a state-space and filter approach, and posterior simulation using a Markov Chain Monte Carlo algorithm. The document concludes with three working examples of DSGE models, their input into BMR, and, finally, a description of the hybrid DSGE-VAR model.

1.1. Bayesian Econometrics

We begin with a brief introduction to Bayesian econometrics. For those interested in an in-depth textbook treatment, see Geweke (2005); the reader might also be interested in Canova (2007), which provides a shorter, more macroeconomics-inspired treatment with BVAR and DSGE models, and (Dave and DeJong, 2007, Ch. 9), which covers DSGE estimation.¹

Denote an arbitrary model by $M_i \in M \subset \mathcal{M}$, where M is a general class of models and \mathcal{M} being the set of all models. The prior density, given by

$$p(\theta|M_i), \quad (1)$$

reflects the researcher's *a priori* beliefs about the k -dimensional parameter vector $\theta \in \Theta \subseteq \mathbb{R}^k$. The exact form of the joint prior (1) will depend on the model in question; for example, when working with Bayesian VARs, the functional form(s) assigned to $p(\theta|M_i)$ will be based on conditional conjugacy considerations, where the conditional posterior distributions are targeted to be from a family of well-known parametric distributions, and, for DSGE models, the choice of prior is largely based on the assumed support for each parameter.

The density of our observed data, $\mathcal{Y} = \mathcal{Y}^T := \{\mathcal{Y}_t\}_{t=1}^T$, conditional on the parameter vector, is given by

$$p(\mathcal{Y}|\theta, M_i). \quad (2)$$

There is an unknown dependence structure between the observations, but we can factorise the joint data density as

$$p(\mathcal{Y}|\theta, M_i) = p(\mathcal{Y}_1|\theta, M_i) \prod_{t=2}^T p(\mathcal{Y}_t|\mathcal{Y}^{t-1}\theta, M_i)$$

When \mathcal{Y} is Markov, this becomes

$$p(\mathcal{Y}|\theta, M_i) = p(\mathcal{Y}_1|\theta, M_i) \prod_{t=2}^T p(\mathcal{Y}_t|\mathcal{Y}_{t-1}, \theta, M_i)$$

where \mathcal{Y}_{t-1} (with θ and M_i) forms a sufficient statistic for the predictive density of \mathcal{Y}_t .

¹For non-textbook references, and examples of Bayesian methods in macroeconomics, see Fernández-Villaverde et al. (2009) and Del-Negro and Schorfheide (2011).

By the definition of conditional probability, we can factorise the joint density of the parameters and data into two equivalent forms:

$$\begin{aligned} p(\mathcal{Y}, \theta | M_i) &= p(\mathcal{Y} | \theta, M_i)p(\theta | M_i) \\ p(\mathcal{Y}, \theta | M_i) &= p(\theta | \mathcal{Y}, M_i)p(\mathcal{Y} | M_i) \end{aligned}$$

Equating these, we have

$$\begin{aligned} p(\theta | \mathcal{Y}, M_i)p(\mathcal{Y} | M_i) &= p(\mathcal{Y} | \theta, M_i)p(\theta | M_i) \\ \Rightarrow p(\theta | \mathcal{Y}, M_i) &= \frac{p(\mathcal{Y} | \theta, M_i)p(\theta | M_i)}{p(\mathcal{Y} | M_i)} \end{aligned} \quad (3)$$

Equation (3) is commonly referred to as Bayes' Theorem. It states that the posterior distribution of the parameters (θ), given the observed data (\mathcal{Y}) and a model (M_i), is equal to the density of our observed data times the prior density of the parameters, divided by a normalising ‘constant’, $p(\mathcal{Y} | M_i)$. The ‘constant’ term is the marginal density of \mathcal{Y} (also known as the marginal data density, or marginal likelihood), where

$$p(\mathcal{Y} | M_i) = \int_{\Theta} p(\mathcal{Y} | \theta, M_i)p(\theta | M_i)d\theta = \int_{\Theta} p(\mathcal{Y}, \theta | M_i)d\theta. \quad (4)$$

In general, this integral cannot be evaluated analytically, but can be approximated using quadrature or Monte Carlo integration. $p(\mathcal{Y} | M_i)$ is important for model comparison and how we judge model ‘fit’. As $p(\mathcal{Y} | M_i)$ is constant for any θ , we can see that

$$p(\theta | \mathcal{Y}, M_i) \propto p(\mathcal{Y} | \theta, M_i)p(\theta | M_i) \equiv \mathcal{K}(\theta | \mathcal{Y}, M_i)$$

where ‘ \propto ’ denotes proportionality and $\mathcal{K}(\theta | \mathcal{Y}, M_i)$ is the kernel of the posterior distribution—*i.e.*, it describes the posterior distribution up to an unknown constant.

If the density $p(\mathcal{Y} | \theta, M_i)$ can be described (at least up to an unknown constant) by the likelihood function $\mathcal{L}(\theta | \mathcal{Y}, M_i)$, we can give the final form of the posterior kernel as

$$p(\theta | \mathcal{Y}, M_i) \propto \mathcal{L}(\theta | \mathcal{Y}, M_i)p(\theta | M_i). \quad (5)$$

Equation (5) is the familiar ‘likelihood times prior’ of Bayesian econometrics.

2. Introduction to Bayesian VARs

An m variable VAR(p) model with T observations is denoted by

$$Y_t = \Phi + \sum_{i=1}^p Y_{t-i} \beta_i + \varepsilon_t \quad (6)$$

where Y_t are the observations at time $t \in [1, T]$, Φ is a matrix of intercepts, β is a matrix of coefficients, and ε are the disturbance terms. The model can be written in stacked form as

$$Y = Z\beta + \varepsilon \quad (7)$$

with matrices: $Y_{(T \times m)}$, $Z_{(T \times (1_c + m \cdot p))}$, $\beta_{((1_c + m \cdot p) \times m)}$, and $\varepsilon_{(T \times m)}$, where 1_c is equal to 1 if there are intercept (constant) terms, zero otherwise, and β includes Φ in its first row. Canova (2007) and Geweke (2005) note a useful, alternative vectorized form for (7)

$$y = (\mathbb{I}_m \otimes Z)\alpha + \epsilon \quad (8)$$

where $\alpha = \text{vec}(\beta)$ (vec being the column-stacking operator), $y_{((T \times m) \times 1)} = \text{vec}(Y)$ is the stacked matrix of observations, $\epsilon_{((T \times m) \times 1)} \sim (0, \Sigma \otimes \mathbb{I}_T)$ is the stacked vector of disturbance terms, \mathbb{I}_T is an identity matrix of size $T \times T$, and \otimes is the Kronecker product.² Assuming the disturbance term is normally distributed, the likelihood function is then

$$\mathcal{L}(\alpha, \Sigma | y, Z) = (2\pi)^{-T \cdot m / 2} |\Sigma \otimes \mathbb{I}_T|^{-1/2} \exp \left\{ -\frac{1}{2} (y - (\mathbb{I}_m \otimes Z)\alpha)^\top (\Sigma^{-1} \otimes \mathbb{I}_T) (y - (\mathbb{I}_m \otimes Z)\alpha) \right\}. \quad (9)$$

where $|\cdot|$ denotes a matrix determinant. The log-likelihood is then

$$\begin{aligned} \ln \mathcal{L} &= -\frac{T \times m}{2} (2\pi) - \frac{1}{2} \ln |\Sigma \otimes \mathbb{I}_T| - \left\{ \frac{1}{2} (y - (\mathbb{I}_m \otimes Z)\alpha)^\top (\Sigma^{-1} \otimes \mathbb{I}_T) (y - (\mathbb{I}_m \otimes Z)\alpha) \right\} \\ &\propto -\frac{1}{2} \ln |\Sigma \otimes \mathbb{I}_T| - \left\{ \frac{1}{2} (y - (\mathbb{I}_m \otimes Z)\alpha)^\top (\Sigma^{-1} \otimes \mathbb{I}_T) (y - (\mathbb{I}_m \otimes Z)\alpha) \right\} \end{aligned} \quad (10)$$

To factorise this expression further, examine the properties of the bracketed term above. First, note that if Σ is symmetric positive-definite, we can factorise as $\Sigma = \Omega \Omega^\top$.³ Also note

²Recall that the Kronecker product of two matrices, say X and Y , with dimensions $a \times b$ and $c \times d$, is a matrix of size $(a \times c) \times (b \times d)$. Also recall the connection between vec operators and Kronecker products: $\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B)$. Thus, $\text{vec}(Z\beta \mathbb{I}_m) = (\mathbb{I}_m^\top \otimes Z)\text{vec}(\beta)$.

³This follows from an eigen decomposition of $\Sigma = Q \Lambda Q^\top$, where Λ is diagonal with elements $\lambda \in \mathbb{R}_{++}$ and Q is

that, when multiplied out, the brackets will yield a scalar value, so the result is invariant to application of the trace function. Now, expand the brackets to give

$$\begin{aligned}
& (y - (\mathbb{I}_m \otimes Z)\alpha)^\top (\Sigma^{-1} \otimes \mathbb{I}_T)(y - (\mathbb{I}_m \otimes Z)\alpha) \\
&= (y - (\mathbb{I}_m \otimes Z)\alpha)^\top (\Omega^{-1} \otimes \mathbb{I}_T)^\top (\Omega^{-1} \otimes \mathbb{I}_T)(y - (\mathbb{I}_m \otimes Z)\alpha) \\
&= [(\Omega^{-1} \otimes \mathbb{I}_T)y - (\Omega^{-1} \otimes Z)\alpha]^\top [(\Omega^{-1} \otimes \mathbb{I}_T)y - (\Omega^{-1} \otimes Z)\alpha]
\end{aligned} \tag{11}$$

which used the fact that $\Sigma^{-1} = (\Omega\Omega^\top)^{-1} = (\Omega^\top)^{-1}\Omega^{-1} = (\Omega^{-1})^\top\Omega^{-1}$. Define

$$\hat{\alpha} := (\Sigma^{-1} \otimes Z^\top Z)^{-1}(\Sigma^{-1} \otimes Z)^\top y \tag{12}$$

Add and subtract $(\Omega^{-1} \otimes Z)\hat{\alpha}$ to each of the square brackets in (11),

$$\begin{aligned}
& [(\Omega^{-1} \otimes \mathbb{I}_T)y + (\Omega^{-1} \otimes Z)(\hat{\alpha} - \alpha - \hat{\alpha})]^\top [(\Omega^{-1} \otimes \mathbb{I}_T)y + (\Omega^{-1} \otimes Z)(\hat{\alpha} - \alpha - \hat{\alpha})] \\
&= [(\Omega^{-1} \otimes \mathbb{I}_T)y - (\Omega^{-1} \otimes Z)\hat{\alpha}]^\top [(\Omega^{-1} \otimes \mathbb{I}_T)y - (\Omega^{-1} \otimes Z)\hat{\alpha}] \\
&\quad + (\hat{\alpha} - \alpha)^\top (\Sigma^{-1} \otimes Z^\top Z)(\hat{\alpha} - \alpha)
\end{aligned}$$

Thus, we can re-write the log-likelihood as

$$\begin{aligned}
\ln \mathcal{L} &\propto -\frac{T}{2} \ln |\Sigma| - \left\{ \frac{1}{2}(\alpha - \hat{\alpha})^\top (\Sigma^{-1} \otimes Z^\top Z)(\alpha - \hat{\alpha}) \right\} \\
&\quad - \left\{ \frac{1}{2} \text{tr} \left[[(\Omega^{-1} \otimes \mathbb{I}_T)y - (\Omega^{-1} \otimes Z)\hat{\alpha}]^\top [(\Omega^{-1} \otimes \mathbb{I}_T)y - (\Omega^{-1} \otimes Z)\hat{\alpha}] \right] \right\}
\end{aligned}$$

by also noting that $\ln(|\Sigma \otimes \mathbb{I}_T|) = \ln(|\Sigma|^T |\mathbb{I}_T|^m) = T \ln |\Sigma| + m \ln |\mathbb{I}_T| = T \ln |\Sigma|$; recall that the determinant of the identity matrix is 1 (the natural log of which is zero).

For notational convenience, define $\mathcal{Z} := \mathbb{I}_m \otimes Z$. With this change, recall the mixed-product property of Kronecker products (which we also used above),

$$\begin{aligned}
(\mathbb{I}_m \otimes Z)^\top (\Sigma^{-1} \otimes \mathbb{I}_T)(\mathbb{I}_m \otimes Z) &= (\mathbb{I}_m \otimes Z)^\top (\Sigma^{-1} \mathbb{I}_m \otimes \mathbb{I}_T Z) \\
&= (\mathbb{I}_m^\top \Sigma^{-1} \mathbb{I}_m \otimes Z^\top \mathbb{I}_T Z) \\
&= (\Sigma^{-1} \otimes Z^\top Z)
\end{aligned}$$

a unitary matrix, $Q^\top Q = QQ^\top = \mathbb{I}$. $\Sigma = \Sigma^\top = Q\Lambda^{1/2}\Lambda^{1/2}Q^\top = Q\Lambda^{1/2}(Q\Lambda^{1/2})^\top := \Omega\Omega^\top =: \Omega^\top\Omega$.

We will also use the transformations:

$$(\Sigma^{-1} \otimes Z^\top Z) = \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z}$$

and

$$\begin{aligned}\hat{\alpha} &= (\Sigma^{-1} \otimes Z^\top Z)^{-1} (\Sigma^{-1} \otimes Z)^\top y \\ &= (\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z})^{-1} \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y\end{aligned}$$

Going back to the log-likelihood,

$$\begin{aligned}\ln \mathcal{L} &\propto -\frac{T}{2} \ln |\Sigma| - \left\{ \frac{1}{2} (\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) \right\} \\ &\quad - \left\{ \frac{1}{2} \text{tr} \left[[(\Omega^{-1} \otimes \mathbb{I}_T) y - (\Omega^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \hat{\alpha}]^\top [(\Omega^{-1} \otimes \mathbb{I}_T) y - (\Omega^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \hat{\alpha}] \right] \right\} \\ &= -\frac{T}{2} \ln |\Sigma| - \left\{ \frac{1}{2} (\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) \right\} \\ &\quad - \left\{ \frac{1}{2} \text{tr} \left[[(\Omega^{-1} \otimes \mathbb{I}_T) (y - \mathcal{Z} \hat{\alpha})]^\top [(\Omega^{-1} \otimes \mathbb{I}_T) (y - \mathcal{Z} \hat{\alpha})] \right] \right\} \\ &= -\frac{T}{2} \ln |\Sigma| - \left\{ \frac{1}{2} (\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) \right\} \\ &\quad - \left\{ \frac{1}{2} \text{tr} \left[[(y - \mathcal{Z} \hat{\alpha})^\top (\Omega^{-1} \otimes \mathbb{I}_T)^\top (\Omega^{-1} \otimes \mathbb{I}_T) (y - \mathcal{Z} \hat{\alpha})] \right] \right\} \\ &= -\frac{T}{2} \ln |\Sigma| - \left\{ \frac{1}{2} (\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) \right\} \\ &\quad - \left\{ \frac{1}{2} \text{tr} \left[[(y - \mathcal{Z} \hat{\alpha})^\top (\Sigma^{-1} \otimes \mathbb{I}_T) (y - \mathcal{Z} \hat{\alpha})] \right] \right\} \\ &= -\frac{T}{2} \ln |\Sigma| - \left\{ \frac{1}{2} (\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) \right\} \\ &\quad - \left\{ \frac{1}{2} \text{tr} \left[[(y - \mathcal{Z} \hat{\alpha})(y - \mathcal{Z} \hat{\alpha})^\top (\Sigma^{-1} \otimes \mathbb{I}_T)] \right] \right\} \\ &\propto \ln (\mathcal{N}(\alpha | \hat{\alpha}, \Sigma, \mathcal{Z}, y) \mathcal{IW}(\Sigma | \hat{\alpha}, \mathcal{Z}, y)).\end{aligned}\tag{13}$$

Thus, the joint likelihood of a VAR model is proportional to the product of a conditional normal distribution (for α) and a conditional inverse-Wishart distribution (for Σ).⁴

⁴Remember that the trace function is invariant under cyclic permutations, thus allowing for some change in the order of multiplication, and the inner-product of conformable vectors is equal to the trace of their outer-product.

2.1. Gibbs Samplers

Almost all Bayesian VAR estimation procedures use a Gibbs-style sampler, where the hierarchical structure of the posterior distribution means that iterated sampling from the conditional posterior distribution of each block of parameters (e.g., α and Σ) is relatively simple. We take this approach because, in general, the conditional posterior distribution of each block of parameters then relates to a well-known parametric distribution.

Let θ denote a vector of parameters of interest—e.g., $\theta = (\alpha, \Sigma)$. Our goal is to generate $\{\theta^{(h)}\}_{h=1}^H$, the realization of a Markov Chain, and as $H \rightarrow \infty$, the resulting empirical distribution—post a sample burn-in phase—should equal our posterior distribution of interest, $p(\theta|\mathcal{Y}, M_i)$.⁵ This approach, known under the general heading of Markov Chain Monte Carlo (MCMC) posterior sampling algorithms, centers around selecting a transition density kernel with an invariant distribution equal to $p(\theta|\mathcal{Y}, M_i)$.

For BVARs, instead of sampling directly from the joint posterior distribution of θ , we separate the parameters into blocks and sample by each block of parameters, $\{\theta_{(b)}\}_{b=1}^B \sim \{p(\theta_{(b)}|\theta_{-(b)}, \mathcal{Y}, M_i)\}_{b=1}^B$, where $\theta_{-(b)}$ denotes all blocks other than the b th block. Thus, Gibbs sampling builds a Markov Chain

$$\{\theta^{(h)}\}_{h=1}^H \sim \left\{ \left\{ p\left(\theta_{(b)} \mid \left\{\theta_{(i)}^{(h)}\right\}_{i < b}^{b-1}, \left\{\theta_{(i)}^{(h-1)}\right\}_{i=b+1}^B, \mathcal{Y}, M_i\right)\right\}_{b=1}^B \right\}_{h=1}^H$$

the realization of which is equivalent to draws from $p(\theta|\mathcal{Y}, M_i)$.

The choice of parameter blocks is model-dependent, and, in some cases, careful conditioning can induce simple forms to the sampling densities. For the BVAR models considered here, the Gibbs chain is $\{\theta^{(h)}\}_{h=1}^H \sim \{p(\alpha|\Sigma^{(h-1)}, \mathcal{Z}, y), p(\Sigma|\alpha^{(h)}, \mathcal{Z}, y)\}_{h=1}^H$, with a transition density kernel $p(\theta^{(h)}|\theta^{(h-1)}) = p(\alpha^{(h)}|\Sigma^{(h-1)}, \mathcal{Z}, y)p(\Sigma^{(h)}|\alpha^{(h)}, \mathcal{Z}, y)$.

An alternative approach would be to sample directly from the joint density of the parameters, though often the joint distribution doesn't have a well-known functional form with analytic expressions for the moments of interest. As we will see later in this document, the parameters of DSGE models are estimated jointly, requiring a different sampling method, namely the Metropolis-Hastings MCMC algorithm (with Gibbs being a special case of this).

⁵Sample burn-in refers to discarding an initial number/proportion of draws to eliminate the effects of initial conditions, allowing the Markov Chain to converge to its invariant distribution; MCMC draws are, in general, serially correlated.

2.2. Normal-inverse-Wishart Prior

The first BVAR model we consider is the normal-inverse-Wishart model, where the kernel of the joint posterior distribution is of α and Σ is

$$p(\alpha, \Sigma | \mathcal{Z}, y) \propto p(y | \mathcal{Z}, \alpha, \Sigma) p(\alpha, \Sigma)$$

We've already derived the data density $p(y | \mathcal{Z}, \alpha, \Sigma)$ in (13), with the form of the joint prior, $p(\alpha, \Sigma)$, left to the econometrician. As noted in the previous section, we will avoid sampling directly from $p(\alpha, \Sigma | \mathcal{Z}, y)$ as this doesn't have a well-known functional form (unless we impose a conditional structure on the joint prior).

Instead, we will choose prior distributions such that sampling from $p(\alpha | \Sigma, \mathcal{Z}, y)$ and $p(\Sigma | \alpha, \mathcal{Z}, y)$ is straightforward; these being a normal and inverse-Wishart distribution, respectively. What follows is a simple derivation of these conditional posterior distributions.

The parameters in the joint prior $p(\alpha, \Sigma)$ are assumed to be independent, and so can be factorized to $p(\alpha, \Sigma) = p(\alpha)p(\Sigma)$, where

$$\begin{aligned} p(\alpha) &= \mathcal{N}(\bar{\alpha}, \Xi_\alpha) \\ &= (2\pi)^{-(m \times p+1)/2} |\Xi_\alpha|^{-1/2} \exp(-(\alpha - \bar{\alpha})^\top \Xi_\alpha^{-1} (\alpha - \bar{\alpha})/2) \end{aligned} \quad (14)$$

and

$$\begin{aligned} p(\Sigma) &= \mathcal{IW}(\Xi_\Sigma, \gamma) \\ &= 2^{-\gamma m/2} \pi^{-m(m-1)/4} |\Xi_\Sigma|^{\gamma/2} \left\{ \prod_{i=1}^m \Gamma[(\gamma + 1 - i)/2] \right\}^{-1} \\ &\quad \cdot |\Sigma|^{-(\gamma-m-1)/2} \exp\left(-\frac{1}{2} \text{tr}(\Xi_\Sigma \Sigma^{-1})\right) \end{aligned} \quad (15)$$

with $\Gamma(\cdot)$ denoting the gamma function.

The form of the posterior distribution of α and Σ is then proportional to the product of (13), (14), and (15). The kernel of the conditional distribution of α (for some given Σ) is thus

$$p(\alpha | \Sigma, \mathcal{Z}, y) \propto \exp\left(-\frac{1}{2} [(\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) + (\alpha - \bar{\alpha})^\top \Xi_\alpha^{-1} (\alpha - \bar{\alpha})]\right)$$

Multiply the brackets out,

$$\begin{aligned}
& (\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) + (\alpha - \bar{\alpha})^\top \Xi_\alpha^{-1} (\alpha - \bar{\alpha}) \\
&= \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \alpha - \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \hat{\alpha} \\
&\quad - \hat{\alpha}^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \alpha + \hat{\alpha}^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \hat{\alpha} \\
&\quad + \alpha^\top \Xi_\alpha^{-1} \alpha - \alpha^\top \Xi_\alpha^{-1} \bar{\alpha} - \bar{\alpha}^\top \Xi_\alpha^{-1} \alpha + \bar{\alpha}^\top \Xi_\alpha^{-1} \bar{\alpha} \\
&= \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \alpha + \alpha^\top \Xi_\alpha^{-1} \alpha - \alpha^\top \Xi_\alpha^{-1} \bar{\alpha} - \bar{\alpha}^\top \Xi_\alpha^{-1} \alpha + \bar{\alpha}^\top \Xi_\alpha^{-1} \bar{\alpha} \\
&\quad - \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \hat{\alpha} - \hat{\alpha}^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \alpha \\
&\quad + \hat{\alpha}^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \hat{\alpha}
\end{aligned}$$

Remember that $\hat{\alpha} := (\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z})^{-1} \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y$, so

$$\begin{aligned}
& (\alpha - \hat{\alpha})^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\alpha - \hat{\alpha}) + (\alpha - \bar{\alpha})^\top \Xi_\alpha^{-1} (\alpha - \bar{\alpha}) \\
&= \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \alpha + \alpha^\top \Xi_\alpha^{-1} \alpha - \alpha^\top \Xi_\alpha^{-1} \bar{\alpha} - \bar{\alpha}^\top \Xi_\alpha^{-1} \alpha + \bar{\alpha}^\top \Xi_\alpha^{-1} \bar{\alpha} \\
&\quad - \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} (\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z})^{-1} \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y \\
&\quad - [(\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z})^{-1} \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y]^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \alpha \\
&\quad + [(\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z})^{-1} \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y]^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \\
&\quad \times (\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z})^{-1} \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y \\
&= \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \alpha + \alpha^\top \Xi_\alpha^{-1} \alpha - \alpha^\top \Xi_\alpha^{-1} \bar{\alpha} - \bar{\alpha}^\top \Xi_\alpha^{-1} \alpha + \bar{\alpha}^\top \Xi_\alpha^{-1} \bar{\alpha} \\
&\quad - \alpha^\top \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y - [\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y]^\top \alpha + y^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y \\
&= \alpha^\top (\mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} + \Xi_\alpha^{-1}) \alpha \\
&\quad - \alpha^\top (\Xi_\alpha^{-1} \bar{\alpha} + \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y) - [\Xi_\alpha^{-1} \bar{\alpha} + \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y]^\top \alpha \\
&\quad + y^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y + \bar{\alpha}^\top \Xi_\alpha^{-1} \bar{\alpha}
\end{aligned}$$

Finally, if we ‘complete the square,’ we can express the conditional kernel as

$$p(\alpha | \Sigma, \mathcal{Z}, y) \propto \exp\left(-\frac{1}{2} [(\alpha - \tilde{\alpha})^\top \tilde{\Sigma}_\alpha^{-1} (\alpha - \tilde{\alpha}) + C]\right) \quad (16)$$

where

$$\begin{aligned}\tilde{\Sigma}_\alpha^{-1} &= \Xi_\alpha^{-1} + \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \\ \tilde{\alpha} &= \tilde{\Sigma}_\alpha (\Xi_\alpha^{-1} \bar{\alpha} + \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y) \\ C &= y' (\Sigma^{-1} \otimes \mathbb{I}_T) y + \bar{\alpha}^\top \Xi_\alpha^{-1} \bar{\alpha} - \tilde{\alpha}^\top \tilde{\Sigma}_\alpha^{-1} \tilde{\alpha}\end{aligned}$$

The conditional posterior distribution of Σ is a little easier to see. Conditional on an α draw, form the β matrix by recalling that $\alpha = \text{vec}(\beta)$, and note that

$$\begin{aligned}\text{tr} [\mathbb{I}_T (Y - Z\beta) \Sigma^{-1} (Y - Z\beta)^\top] &= \text{vec} \left(((Y - Z\beta)^\top)^\top \right)^\top \left((\Sigma^{-1})^\top \otimes \mathbb{I}_T \right) \text{vec} (Y - Z\beta) \\ &= \text{vec} (Y - Z\beta)^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \text{vec} (Y - Z\beta)\end{aligned}$$

Then, the posterior kernel of Σ is

$$\begin{aligned}p(\Sigma | \beta, Z, Y) &\propto |\Sigma|^{-T/2} |\Sigma|^{-(\gamma-m-1)/2} \\ &\quad \times \exp \left(-\frac{1}{2} \text{tr} [\Xi_\Sigma \Sigma^{-1}] \right) \exp \left(-\frac{1}{2} \text{tr} [\Sigma^{-1} (Y - Z\beta)^\top (Y - Z\beta)] \right) \\ &\propto |\Sigma|^{-(T+\gamma-m-1)/2} \times \exp \left(-\frac{1}{2} \text{tr} [(\Xi_\Sigma + (Y - Z\beta)^\top (Y - Z\beta)) \Sigma^{-1}] \right) \quad (17)\end{aligned}$$

Thus, we have well-known parametric distributions that define the blocks of our conditional posterior distributions of interest, namely the normal and inverse-Wishart distributions,

$$p(\alpha | \Sigma, \mathcal{Z}, y) = \mathcal{N}(\tilde{\alpha}, \tilde{\Sigma}_\alpha) \quad (18)$$

$$p(\Sigma | \beta, Z, Y) = \mathcal{IW}(\Xi_\Sigma + (Y - Z\beta)^\top (Y - Z\beta), T + \gamma) \quad (19)$$

Sampling from these distributions is a simple task.

2.3. Minnesota Prior

The primary distinction between the previous model and the so-called Minnesota (or Litterman) prior is that Σ , the covariance matrix of the disturbance terms, is fixed before posterior sampling begins, and this yields a significant decrease in computational burden; the standard references for this type of model are Doan et al. (1983) and Litterman (1986).

The Minnesota prior sets the Σ matrix to be diagonal, where the diagonal elements come from equation-by-equation estimation of an AR(p) model for each of the m -variables. That is, for each variable in the VAR model, we estimate the model assuming that all coefficients, except their own-lag terms (and a possible constant term), are equal to zero.

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & 0 & 0 & 0 & \sigma_m \end{bmatrix}$$

Let $i, j \in \{1, \dots, m\}$; with the equation being indexed by i , and the variable indexed by j . (Koop and Korobilis, 2010, p. 7) define the prior covariance matrix of β as follows:

$$\Xi_{\beta_{i,j}}(\ell) = \begin{cases} H_1/\ell^2 \\ H_2 \cdot \sigma_i^2 / (\ell^2 \cdot \sigma_j^2) \\ H_3 \cdot \sigma_i^2 \end{cases}$$

which correspond to own lags, cross-variable lags, and exogenous variables (a constant), respectively, where $\ell \in \{1, \dots, p\}$. (Canova, 2007, p. 378) defines a more general version of this procedure as

$$\Xi_{\beta_{i,j}}(\ell) = \begin{cases} H_1/d(\ell) \\ H_1 \cdot H_2 \cdot \sigma_j^2 / (d(\ell) \cdot \sigma_i^2) \\ H_1 \cdot H_3 \end{cases}$$

which, again, correspond to own lags, cross-variable lags, and exogenous variables, respectively, where $d(\ell)$ is the ‘decay’ function; in the previous case, $d(\ell) = \ell^2$. Also, note that, for cross-equation coefficients, the ratio of the variance of each equation has been inverted (*i.e.*, σ_i^2 is the denominator and σ_j^2 is in the numerator).

The user can choose between two functional forms of $d(\ell)$, harmonic and geometric decay,

$$d(\ell) = \begin{cases} \ell^{H_4} \\ H_4^{-\ell+1} \end{cases}$$

respectively, where $H_4 > 0$. (When $H_4 = 1$, we have linear decay $d(\ell) = \ell$.) To select harmonic decay, one enters “H” in the decay field (including quotation marks) and “G” for geometric. To select the Koop and Korobilis (2010) version, enter VType = 1 in the Minnesota BVAR function, and VType = 2 for the Canova (2007) version. (See Section 4.2 for further details.)

Finally, note that the dimensions of Ξ_β will be $(\mathbf{1}_c + m \cdot p) \times m$, which is then vectorised, and this becomes the main diagonal elements to a new matrix of size $((\mathbf{1}_c + m \cdot p) \times m) \times ((\mathbf{1}_c + m \cdot p) \times m)$, where all other entries are set to zero.

With Σ fixed (with elements as described above), the posterior distribution of α is similar to the normal-inverse-Wishart case,

$$p(\alpha | \Sigma, \mathcal{Z}, Y) = \mathcal{N}(\tilde{\alpha}, \tilde{\Sigma}_\alpha), \quad (20)$$

where

$$\begin{aligned} \tilde{\Sigma}_\alpha^{-1} &= \Xi_\alpha^{-1} + \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) \mathcal{Z} \\ \tilde{\alpha} &= \tilde{\Sigma}_\alpha (\Xi_\alpha^{-1} \bar{\alpha} + \mathcal{Z}^\top (\Sigma^{-1} \otimes \mathbb{I}_T) y) \end{aligned}$$

From the description above, we see the rather convenient computational aspect of the Minnesota prior. The posterior covariance matrix of the coefficients, $\tilde{\Sigma}_\alpha$, which will be of size $(\mathbf{1}_c + m \cdot p) \cdot m \times (\mathbf{1}_c + m \cdot p) \cdot m$, need only be calculated—and, more importantly, inverted—once, as Σ is fixed for all sample draws of α ; the prior and data won’t change. In BMR, this implies that a model with a Minnesota prior will be estimated several-hundred times faster than with a normal-inverse-Wishart prior.

3. Steady State BVAR

Rather than placing a prior on the intercept vector Φ in (7), Villani (2009) opts to place a prior directly on the unconditional mean of each series by restating the problem as

$$(Y_t - d_t \Psi) \beta(L) = \varepsilon_t, \quad (21)$$

where $d_t \Psi$ is the unconditional mean— d_t is a $1 \times q$ matrix of exogenous variables (at time t) with coefficients $\Psi_{q \times m}$ —and $\beta(L)$ is the matrix lag-polynomial⁶

$$\beta(L) = \mathbb{I}_m - \sum_{i=1}^p L^i \beta_i$$

We can then rewrite (21) in more familiar form as

$$Y_t = d_t \Psi + \sum_{i=1}^p (Y_{t-i} - d_{t-i} \Psi) \beta_i + \varepsilon_t$$

In BMR, $d_t = d \ \forall t$, so the only exogenous variable permitted is a column-vector of ones.⁷ Let $Z = [Y_{t-1} \ Y_{t-2} \ \dots \ Y_{t-p}]$, and $\psi = \text{vec}(\Psi)$; we can rewrite the model in full matrix form as

$$Y = \iota_T \psi^\top + Z \beta - (\iota_T \cdot \iota_p^\top) (\mathbb{I}_p \otimes \Psi) \beta + \varepsilon \quad (22)$$

with matrices: $Y_{(T \times m)}$, $Z_{(T \times (m \cdot p))}$, $\psi_{(m \times 1)}$, $\beta_{((m \cdot p) \times m)}$, and $\varepsilon_{(T \times m)}$.⁸ Imposing the condition that the eigenvalues of $\beta(L)$ lie inside the unit circle, we can see how Ψ is a non-linear function of the standard BVAR setup with Φ and β , $\Psi_{1 \times m} = \Phi_{1 \times m} (\mathbb{I}_m - \sum_{i=1}^p \beta_i)^{-1}$.

The estimation procedure in BMR follows that of Warne (2012). The priors for each block of parameters are

$$p(\psi) = \mathcal{N}(\bar{\psi}, \Xi_\psi) \quad (23)$$

$$p(\beta) = \mathcal{N}(\bar{\beta}, \Xi_\beta) \quad (24)$$

$$p(\Sigma) = \mathcal{IW}(\Xi_\Sigma, \gamma) \quad (25)$$

⁶ L is the backshift operator; i.e., $y_t L = y_{t-1}$, $y_t L^2 = y_{t-2}$, ..., $y_t L^i = y_{t-i}$, $i \in \mathbb{Z}$.

⁷The reason for this is a technical one, and I direct those looking for a more general implementation to Warne (2012). (In future implementations, I intend to relax this restriction.)

⁸The symbol ι denotes a column-vector of ones, the length of which is indicated by a subscript.

Two restrictions apply to this. First, Ξ_β is defined by a simplified version of harmonic decay, shown in the Minnesota prior section,

$$\Xi_\beta(\ell) = (H_1/\ell^{H_4}) \cdot \mathbb{I}_m$$

Second, Ξ_Σ is set to the maximum likelihood estimate of Σ .

Let Y_d and Z_d be demeaned series, given by

$$\begin{aligned} Y_d &= Y - \iota_T \Psi \\ Z_d &= Z - (\iota_T \cdot \iota_p^\top) (\mathbb{I}_p \otimes \Psi) \end{aligned}$$

respectively, and let

$$\xi = Y - Z\beta$$

$$\xi_d = Y_d - Z_d \beta$$

The conditional posterior distributions of our parameters are

$$p(\psi|\beta, \Sigma, Z, Y) = \mathcal{N}(\tilde{\psi}, \tilde{\Sigma}_\psi) \quad (26)$$

$$p(\beta|\psi, \Sigma, Z, Y) = \mathcal{N}(\tilde{\beta}, \tilde{\Sigma}_\beta \otimes \Sigma) \quad (27)$$

$$p(\Sigma|\psi, \beta, Z, Y) = \mathcal{IW}\left(\Xi_\Sigma + \xi_d^\top \xi_d + (\tilde{\beta} - \bar{\beta})^\top \Xi_\beta^{-1} (\tilde{\beta} - \bar{\beta}), T + m \cdot p + \gamma\right) \quad (28)$$

where

$$\begin{aligned} \tilde{\psi} &= \tilde{\Sigma}_\psi (U^\top \text{vec}(\Sigma^{-1} \xi^\top D) + \Xi_\psi^{-1} \bar{\psi}) \\ \tilde{\Sigma}_\psi &= \left(\Xi_\psi^{-1} + U^\top (D^\top D \otimes \Sigma^{-1}) U \right)^{-1} \\ \tilde{\beta} &= \tilde{\Sigma}_\beta (Z_d^\top Y_d + \Xi_\beta^{-1} \bar{\beta}) \\ \tilde{\Sigma}_\beta &= (\Xi_\beta^{-1} + Z_d^\top Z_d)^{-1} \end{aligned}$$

with $U_{((m+m \cdot p) \times m)} = [\mathbb{I}_m \ \beta_1^\top \ \beta_2^\top \ \cdots \ \beta_p^\top]^\top$ and $D_{(T \times (p+1))} = [\iota_T \ -(\iota_T \cdot \iota_p^\top)]$.

4. Main BVAR Functions

4.1. Normal-inverse-Wishart Prior

```
BVARW(mydata,cores=1,coefprior=NULL,p=4,constant=TRUE,
      irf.periods=20,keep=10000,burnin=1000,
      XiBeta=1,XiSigma=1,gamma=NULL)
```

- **mydata**

A matrix or data frame containing the data series used for estimation; this should be of size $T \times m$.

- **cores**

A positive integer value indicating the number of CPU cores that should be used for the sampling run. **Do not allocate more cores than your computer can safely handle!** If in doubt, set cores = 1, which is the default.

- **coefprior**

A numeric vector of length m , matrix of size $(m \cdot p + 1_c) \times m$, or a value of ‘NULL’, that contains the prior mean-value of each coefficient. Providing a numeric vector of length m will set a zero prior on all coefficients except the own first-lags, which are set according to the elements in ‘coefprior’. Setting this input to ‘NULL’ will give a random-walk-in-levels prior.

- **p**

The number of lags to include of each variable, where $p \in \mathbb{Z}_{++}$. The default value is 4.

- **constant**

A logical statement on whether to include a constant vector (intercept) in the model. The default is ‘TRUE’, and the alternative is ‘FALSE’.

- **irf.periods**

An integer value for the horizon of the impulse response calculations, which must be greater than zero. The default value is 20.

- **keep**

The number of Gibbs sampling replications to keep from the sampling run.

- **burnin**

The sample burn-in length for the Gibbs sampler.

- **XiBeta**

A numeric vector of length 1 or matrix of size $(m \cdot p + 1_c) \cdot m \times (m \cdot p + 1_c) \cdot m$ comprising the prior covariance of each coefficient. If the user supplies a scalar value, then the prior becomes $\mathbb{I}_{(m \cdot p + 1_c) \cdot m} \cdot \Xi_\beta$. The structure of Ξ_β corresponds to $\text{vec}(\beta)$.

- **XiSigma**

A numeric vector of length 1 or matrix of size $m \times m$ that contains the location matrix of the inverse-Wishart prior. If this is a scalar then Ξ_Σ is $\mathbb{I}_{(m \cdot p + 1_c)m} \cdot \Xi_\Sigma$.

- **gamma**

A numeric vector of length 1 corresponding to the prior degrees of freedom of the error covariance matrix. The minimum value is $m + 1$, and this is the default value.

The function returns an object of class ‘BVARW’, which contains:

- **Beta**

A matrix of size $(m \cdot p + 1_c) \times m$ containing the posterior mean of the coefficient matrix (β).

- **BDraws**

An array of size $(m \cdot p + 1_c) \times m \times \text{keep}$ which contains the post burn-in draws of β .

- **Sigma**

A matrix of size $m \times m$ containing the posterior mean estimate of the residual covariance matrix (Σ).

- **SDraws**

An array of size $m \times m \times \text{keep}$ which contains the post burn-in draws of Σ .

- **IRFs**

A four-dimensional object of size $\text{irf.periods} \times m \times \text{keep} \times m$ containing the impulse response function calculations; the first m refers to responses to the last m shock.

- **data**

The data used for estimation.

- **constant**

A logical value, TRUE or FALSE, indicating whether the user chose to include a vector of constants in the model.

4.2. Minnesota Prior

```
BVARM(mydata,coefprior=NULL,p=4,constant=TRUE,
      irf.periods=20,keep=10000,burnin=1000,
      VType=1,decay="H",HP1=0.5,HP2=0.5,HP3=1,HP4=2)
```

- **mydata**

A matrix or data frame containing the data series used for estimation; this should be of size $T \times m$.

- **coefprior**

A numeric vector of length m , matrix of size $(m \cdot p + 1_c) \times m$, or a value of ‘NULL’, that contains the prior mean-value of each coefficient. Providing a numeric vector of length m will set a zero prior on all coefficients except the own first-lags, which are set according to the elements in ‘coefprior’. Setting this input to ‘NULL’ will give a random-walk-in-levels prior.

- **p**

The number of lags to include of each variable, where $p \in \mathbb{Z}_{++}$. The default value is 4.

- **constant**

A logical statement on whether to include a constant vector (intercept) in the model. The default is ‘TRUE’, and the alternative is ‘FALSE’.

- **irf.periods**

An integer value for the horizon of the impulse response calculations, which must be greater than zero. The default value is 20.

- **keep**

The number of Gibbs sampling replications to keep from the sampling run.

- **burnin**

The sample burn-in length for the Gibbs sampler.

- **VType**

Whether to use a variance type of Koop and Korobilis (2010) ('VType=1') or Canova (2007) ('VType=2'). The default is 1.

- **decay**

Whether to use harmonic or geometric decay for the VType=2 case.

- **HP1, HP2, HP3, HP4**

These correspond to H_1, H_2, H_3 , and H_4 , respectively, from section 2.3. Recall that the prior covariance matrix of β , Ξ_β , is given by

$$\Xi_{\beta_{i,j}}(\ell) = \begin{cases} H_1/\ell^2 \\ H_2 \cdot \sigma_i^2 / (\ell^2 \cdot \sigma_j^2) \\ H_3 \cdot \sigma_i^2 \end{cases}$$

in the case of VType=1, and

$$\Xi_{\beta_{i,j}}(\ell) = \begin{cases} H_1/d(\ell) \\ H_1 \cdot H_2 \cdot \sigma_j^2 / (d(\ell) \cdot \sigma_i^2) \\ H_1 \cdot H_3 \end{cases}$$

in the case of VType=2, where

$$d(\ell) = \begin{cases} \ell^{H_4} \\ H_4^{-\ell+1} \end{cases}$$

is the decay function.

The function returns an object of class 'BVARM', which contains:

- **Beta**

A matrix of size $(m \cdot p + \mathbf{1}_c) \times m$ containing the posterior mean of the coefficient matrix

(β) .

- **BDraws**

An array of size $(m \cdot p + \mathbf{1}_c) \times m \times \text{keep}$ which contains the post burn-in draws of β .

- **BetaVPr**

An matrix of size $(m \cdot p + \mathbf{1}_c) \cdot m \times (m \cdot p + \mathbf{1}_c) \cdot m$ containing the prior covariance matrix of $\text{vec}(\beta)$.

- **Sigma**

A matrix of size $m \times m$ containing the fixed residual covariance matrix (Σ).

- **IRFs**

A four-dimensional object of size $\text{irf.periods} \times m \times \text{keep} \times m$ containing the impulse response function calculations; the first m refers to responses to the last m shock.

- **data**

The data used for estimation.

- **constant**

A logical value, TRUE or FALSE, indicating whether the user chose to include a vector of constants in the model.

4.3. Steady State Prior

```
BVARS(mydata,psiprior=NULL,coefprior=NULL,p=4,
      irf.periods=20,keep=10000,burnin=1000,
      XiPsi=1,HP1=0.5,HP4=2,gamma=NULL)
```

- **mydata**

A matrix or data frame containing the data series used for estimation; this should be of size $T \times m$.

- **psiprior**

A numeric vector of length m that contains the prior mean of each series found in ‘my-data’. The user MUST specify this prior, or the function will return an error.

- **coefprior**

A numeric vector of length m , matrix of size $(m \cdot p) \times m$, or a value of ‘NULL’, that contains the prior mean-value of each coefficient. Providing a numeric vector of length m will set a zero prior on all coefficients except the own first-lags, which are set according to the elements in ‘coefprior’. Setting this input to ‘NULL’ will give a random-walk-in-levels prior.

- **p**

The number of lags to include of each variable, where $p \in \mathbb{Z}_{++}$. The default value is 4.

- **irf.periods**

An integer value for the horizon of the impulse response calculations;, which must be greater than zero. The default value is 20.

- **keep**

The number of Gibbs sampling replications to keep from the sampling run.

- **burnin**

The sample burn-in length for the Gibbs sampler.

- **XiPsi**

A numeric vector of length 1 or matrix of size $m \times m$ that defines the prior location matrix of Ψ .

- **HP1**

H_1 from section 3.

- **HP4**

H_4 from section 3.

- **gamma**

A numeric vector of length 1 corresponding to the prior degrees of freedom of the error covariance matrix. The minimum value is $m + 1$, and this is the default value.

The function returns an object of class ‘BVARS’, which contains:

- **Beta**

A matrix of size $(m \cdot p) \times m$ containing the posterior mean of the coefficient matrix (β).

- **BDraws**

An array of size $(m \cdot p) \times m \times \text{keep}$ which contains the post burn-in draws of β .

- **Psi**

A matrix of size $1 \times m$ containing the posterior mean estimate of the unconditional mean matrix (Ψ).

- **PDraws**

An array of size $1 \times m \times \text{keep}$ which contains the post burn-in draws of Ψ .

- **Sigma**

A matrix of size $m \times m$ containing the posterior mean estimate of the residual covariance matrix (Σ).

- **SDraws**

An array of size $m \times m \times \text{keep}$ which contains the post burn-in draws of Σ .

- **IRFs**

A four-dimensional object of size `irf.periods` $\times m \times \text{keep} \times m$ containing the impulse response function calculations; the first m refers to responses to the last m shock.

- **data**

The data used for estimation.

5. Additional Functions

5.1. Classical VAR

```
CVAR(mydata,p=4,constant=TRUE,irf.periods=20,boot=10000)
```

- **mydata**

A matrix or data frame containing the data series used for estimation; this should be of size $T \times m$.

- **p**

The number of lags to include of each variable, where $p \in \mathbb{Z}_{++}$. The default value is 4.

- **constant**

A logical statement on whether to include a constant vector (intercept) in the model. The default is ‘TRUE’, and the alternative is ‘FALSE’.

- **irf.periods**

An integer value for the horizon of the impulse response calculations, which must be greater than zero. The default value is 20.

- **boot**

The number of replications to run for the bootstrapped IRFs. The default is 10,000.

For the VAR model,

$$Y = Z\beta + \varepsilon$$

the algorithm to obtain bootstrapped samples in BMR is as follows. First, obtain the OLS/MLE estimates of the coefficients, $\hat{\beta}$, and the estimated disturbance terms $\hat{\varepsilon} = y - Z\hat{\beta}$. Then, sample with replacement from $\hat{\varepsilon}$ for a user-specified (H) number of times (for example, 10,000 times, the default value of ‘boot’ above), and build H new series of Y by the following method. For the purpose of illustration, let $p = 2$ and let there be a constant.

1. Let $h \in [1, H]$, $t \in [1, T]$, $Z = [\iota_T \ Y_{t-1} \ Y_{t-2} \ \cdots \ Y_{t-p}]$, and fix $Y_0, Y_{-1}, \dots, Y_{-p+1}$. Using $\hat{\beta}$, calculate

$$Y_1^{(h)} = \hat{\Phi} + Y_0\hat{\beta}_1 + Y_{-1}\hat{\beta}_2 + \cdots + Y_{-p+1}\hat{\beta}_p + \hat{\varepsilon}_1^{(h)}$$

2. Then, using $Y_1^{(h)}$,

$$Y_2^{(h)} = \hat{\Phi} + Y_1^{(h)}\hat{\beta}_1 + Y_0\hat{\beta}_2 + \cdots + Y_{-p+2}\hat{\beta}_p + \hat{\varepsilon}_2^{(h)}$$

3. Continuing in this fashion:

$$\begin{aligned} Y_3^{(h)} &= \hat{\Phi} + Y_2^{(h)}\hat{\beta}_1 + Y_1^{(h)}\hat{\beta}_2 + \cdots + Y_{-p+3}\hat{\beta}_p + \hat{\varepsilon}_3^{(h)} \\ &\vdots \quad = \vdots \quad \vdots \quad \vdots \quad \vdots \\ Y_T^{(h)} &= \hat{\Phi} + Y_{T-1}^{(h)}\hat{\beta}_1 + Y_{T-2}^{(h)}\hat{\beta}_2 + \cdots + Y_{T-p}\hat{\beta}_p + \hat{\varepsilon}_T^{(h)} \end{aligned}$$

- 4. Then, with the new $Y^{(h)}$ series, estimate $\hat{\beta}^{(h)}$ and $\hat{\Sigma}^{(h)}$ and store them.
- 5. Go back to part 1 and do it all again for a new $h = h + 1$.
- 6. After repeating the process above H times, and storing H number of β and Σ matrices, estimate the IRFs.

The function returns an object of class ‘CVAR’, which contains:

- **Beta**

A matrix of size $(m \cdot p + \mathbf{1}_c) \times m$ containing the OLS estimate of the coefficient matrix (β) .

- **BDraws**

An array of size $(m \cdot p + \mathbf{1}_c) \times m \times \text{keep}$ which contains the bootstrapped β draws.

- **Sigma**

A matrix of size $m \times m$ containing the OLS estimates of the residual covariance matrix (Σ) .

- **SDraws**

An array of size $m \times m \times \text{keep}$ which contains bootstrapped Σ draws.

- **IRFs**

A four-dimensional object of size $\text{irf.periods} \times m \times \text{boot} \times m$ containing the impulse response function calculations. The first m refers to the responses to the last m shock.

- **data**

The data used for estimation.

- **constant**

A logical value, TRUE or FALSE, indicating whether the user chose to include a vector of constants in the model.

5.2. GACF and GPACF

While the autocorrelation and partial autocorrelation functions ('acf' and 'pacf') in the base package of R 'work', this author dislikes the graphs they produce, especially the rather unnecessary zero-lag on the ACF. BMR includes two functions to produce ACFs and PACFs with GGPLOT2, and modifies the confidence intervals to instead use Bartlett's formula.

```
gacf(y, lags=10, ci=.95, plot=T, barcolor="purple",
      names=FALSE, save=FALSE, height=12, width=12)
gpacf(y, lags=10, ci=.95, plot=T, barcolor="darkred",
      names=FALSE, save=FALSE, height=12, width=12)
```

- **y**

A matrix or data frame of size $T \times m$ containing the relevant series.

- **lags**

The number of lags to plot.

- **ci**

A numeric value between 0 and 1 specifying the confidence interval to use; the default value is 0.95.

- **barcolor**

The color of the bars.

- **names**

Whether to plot the names of the series.

- **save**

Whether to save the plots. The default is 'FALSE'.

- **height**

If save = TRUE, use this to set the height of the plot.

- **width**

If save = TRUE, use this to set the width of the plot.

5.3. Time-Series Plot

```
gtsplot(X,dates=NULL,rowdates=FALSE,dates.format="%Y-%m-%d",
        save=FALSE,height=13,width=11)
```

- **X**

A matrix or data frame of size $T \times m$ containing the relevant time-series data, where m is the number of series.

- **dates**

A $T \times 1$ date or character vector containing the relevant date stamps for the data.

- **rowdates**

A TRUE or FALSE statement indicating whether the row names of the X matrix contain the date stamps for the data.

- **dates.format**

If ‘dates’ is not set to NULL, then indicate what format the dates are in, such as Year-Month-Day.

- **save**

Whether to save the plot(s).

- **height**

The height of the saved plot(s).

- **width**

The width of the saved plot(s).

5.4. Stationarity

BMR extends a Bayesian hand of friendship to the Bayesian-Classical agnostic by including a basic function to run Augmented Dickey-Fuller (ADF) and KPSS tests.

```
stationarity(y, KPSSp=4, ADFp=8, print=TRUE)
```

- **y**

A matrix or data frame containing the series to be used in testing, and should be of size $T \times m$.

- **KPSSp**

The number of lags to include for the test of Kwiatkowski et al. (1992), the ‘KPSS test’, based on a model

$$y_t = \delta t + \xi_t + \varepsilon_{Y,t},$$

$$\xi_t = \xi_{t-1} + \varepsilon_{\xi,t}$$

or

$$y_t = \delta t + \sum_{s=0}^t \varepsilon_{\xi,s} + \varepsilon_{Y,t}$$

Let $\hat{\varepsilon}$ be the estimated residuals from regressing y on a constant and time trend; the test statistic is

$$K = \frac{1}{T^2 \hat{\sigma}_\varepsilon^2} \hat{\varepsilon}^\top \hat{\varepsilon},$$

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{T} \left(\sum_{t=1}^T \hat{\varepsilon}_t + 2 \sum_{j=1}^p \left(1 - \frac{j}{1+p} \right) \hat{\varepsilon}_{j+1:T}^\top \hat{\varepsilon}_{1:T-j} \right)$$

The test has a null hypothesis of $\hat{\sigma}_{\varepsilon_\xi}^2 = 0$, stationarity. The function will return test statistics for both $\delta \neq 0$ and $\delta = 0$, that is, with a time trend and without a time trend. The critical values for the KPSS test are from (Kwiatkowski et al., 1992, Table 1).

- **ADFp**

The maximum number of (first-differenced) lags to include in the Augmented Dickey-Fuller (ADF) tests, $p \in \mathbb{Z}_{++}$. The default value is 8, and the optimal number of lags

is based on minimising the Bayesian information criterion. Three different functional forms are estimated for the ADF tests; in the order that they're returned: with drift and a time trend

$$\Delta y_t = \delta_0 + \delta_1 t + \beta y_{t-1} + \sum_{i=1}^p \alpha_i \Delta y_{t-i} + \varepsilon_t,$$

with drift, but without a time trend,

$$\Delta y_t = \delta_0 + \beta y_{t-1} + \sum_{i=1}^p \alpha_i \Delta y_{t-i} + \varepsilon_t,$$

and, finally, without drift or a time trend,

$$\Delta y_t = \beta y_{t-1} + \sum_{i=1}^p \alpha_i \Delta y_{t-i} + \varepsilon_t,$$

where Δ is the first-difference operator. The ADF test is based on a null-hypothesis that $\beta = 0$ and alternative hypothesis $\beta < 0$. See any good time-series textbook for further information. The critical values given by BMR are those found in (Hamilton, 1994, Appendix 5).

- **print**

A logical statement on whether the test results should be printed in the output screen. the default is 'TRUE'.

6. BVAR Examples

This section illustrates the use the BVAR functions in BMR. First, we estimate a VAR(2) model with Minnesota prior, then a VAR(4) model with normal-inverse-Wishart prior, and conclude with a steady-state example.

6.1. Monte Carlo Study

For the purpose of illustration, the following two variable VAR(2) model was used to generate 100 artificial data points,

$$\mathbf{Y} = \begin{bmatrix} Y_{1,t} & Y_{2,t} \end{bmatrix}, \quad \Phi = \begin{bmatrix} 7 & 3 \end{bmatrix}$$

$$\beta = \begin{bmatrix} 0.5 & 0.2 \\ 0.28 & 0.7 \\ -0.39 & -0.1 \\ 0.1 & 0.05 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

With this data, we estimate a BVAR with Minnesota prior using the ‘BVARM’ function. As input to the function, we include the data (labelled: ‘bvarMCdata’), set the prior on the first own-lag coefficients to 0.5 for both variables, and everything else to zero; set the number of lags to 2; include a vector of constants in the model; set the impulse response horizon to 20; set the number of runs of the Gibbs sampler to 15000, 5000 of which being sample burn-in; set the variance type (as described in section 3) to ‘1’; and set the hyper-parameters H_1 , H_2 , and H_3 to values of 0.5, 0.5, and 10, respectively.

```
data(BMRMCData)
testbvarm <- BVARM(bvarMCdata,c(0.5,0.5),p=2,constant=TRUE,irf.periods=20,
                     keep=10000,burnin=5000,VType=1,
                     HP1=0.5,HP2=0.5,HP3=10)
Starting Gibbs C++, Tue Jan 20 10:33:17 2015.
C++ reps finished, Tue Jan 20 10:33:17 2015. Now generating IRFs.
```

We can check the mean coefficient estimate by accessing \$Beta:

```
testbvarm$Beta
```

	Var1	Var2
Constant	6.3523726	2.90617697
Eq 1, lag 1	0.5585694	0.30112058
Eq 2, lag 1	0.2656987	0.57107739
Eq 1, lag 2	-0.3699877	-0.06192194
Eq 2, lag 2	0.1750157	0.05618627

and, for those who prefer to visualise the parameter densities, we can plot the posterior distributions of each coefficient with

```
plot(testbvarm,type=1,save=T)
```

the results of which are shown in figures 1, 2, and 3.

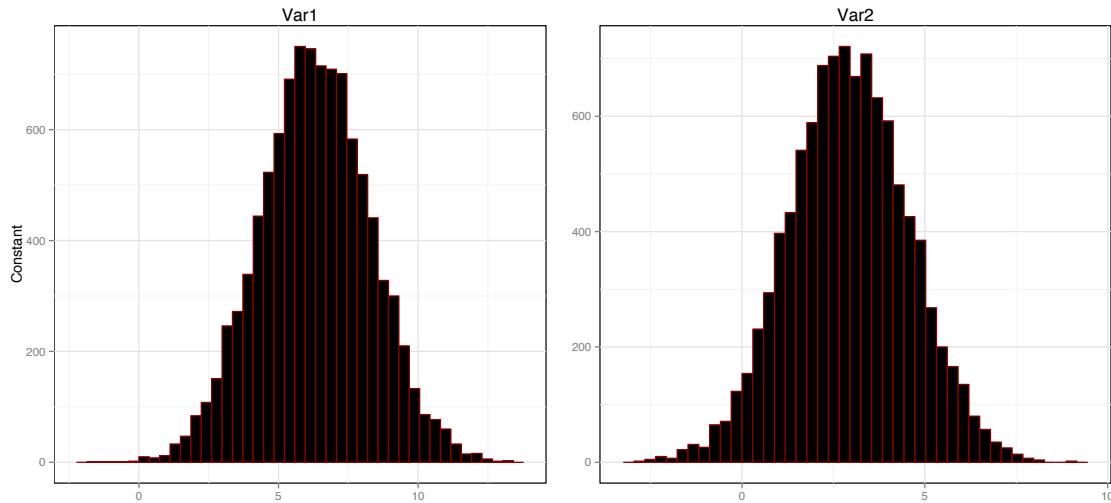


Figure 1: Posterior Distributions of Constant Terms.

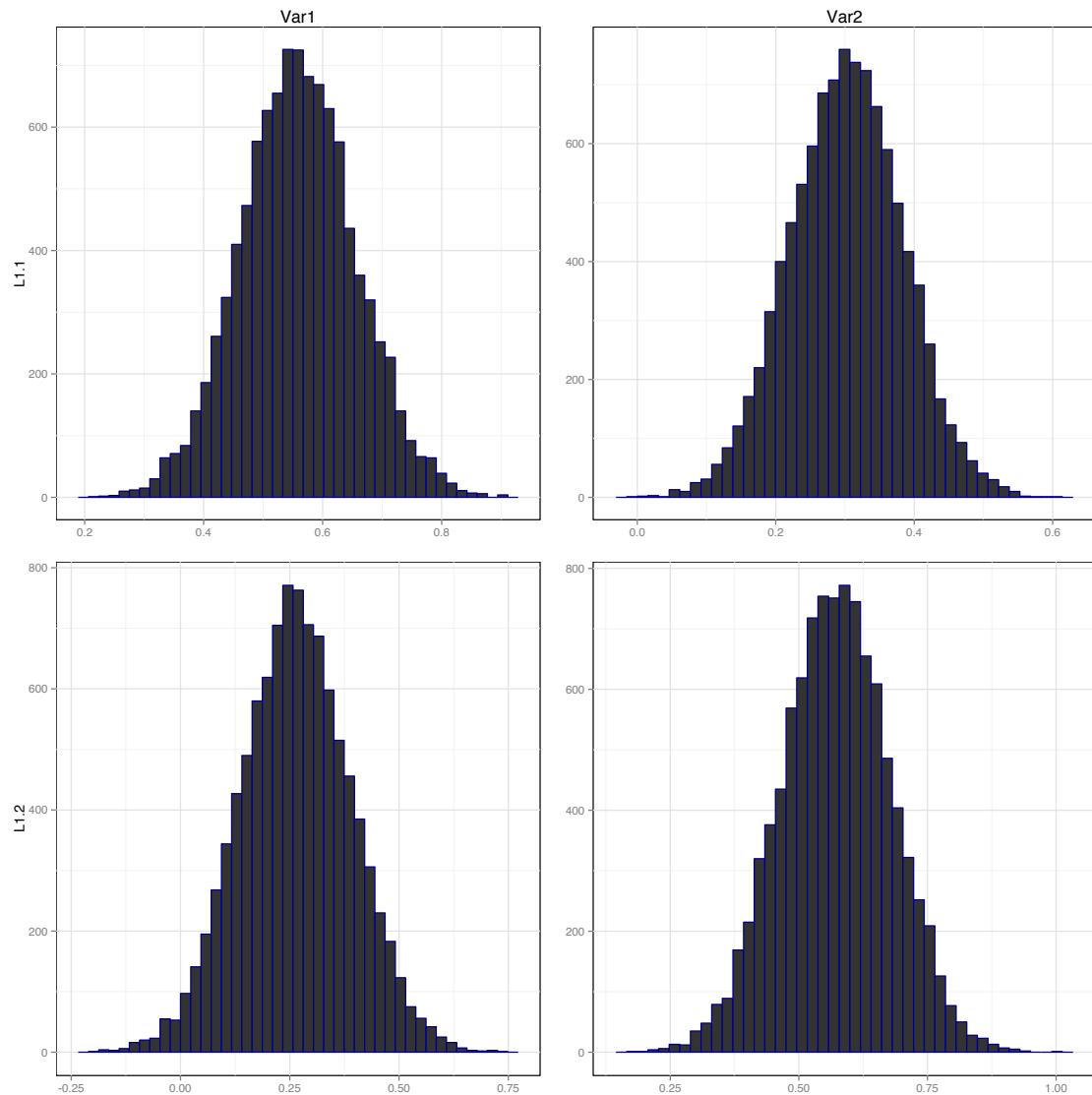


Figure 2: Posterior Distributions of First Lags.

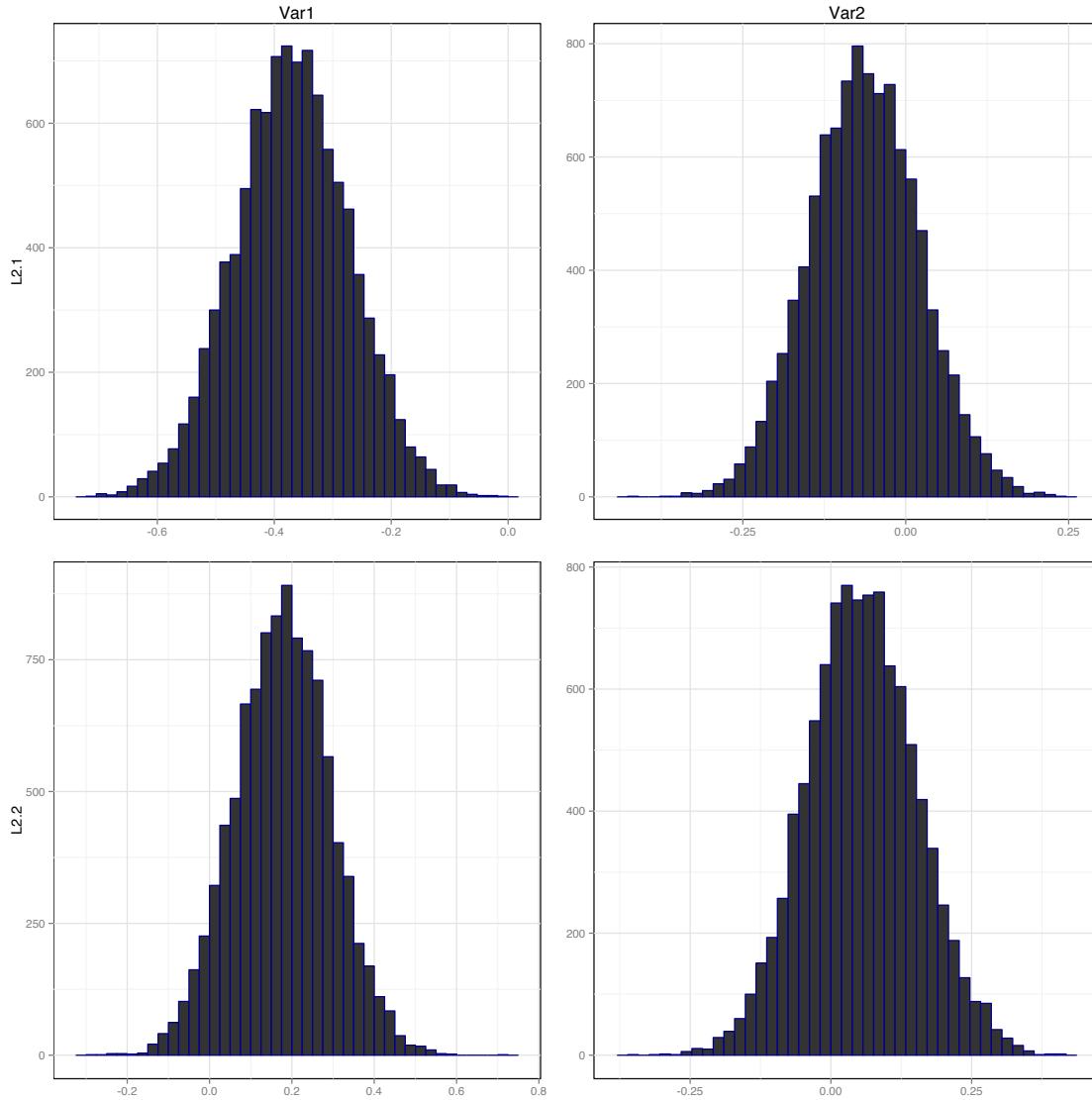


Figure 3: Posterior Distributions of Second Lags.

Finally, we can plot the IRFs, using the median percentile as the the central tendency, along with the 5th and 95th percentile bands, using

```
> IRF(testbvarm,percentiles=c(0.05,0.5,0.95),save=T)
```

This will save 'IRFs.eps' to your working directory, and the result is illustrated below.

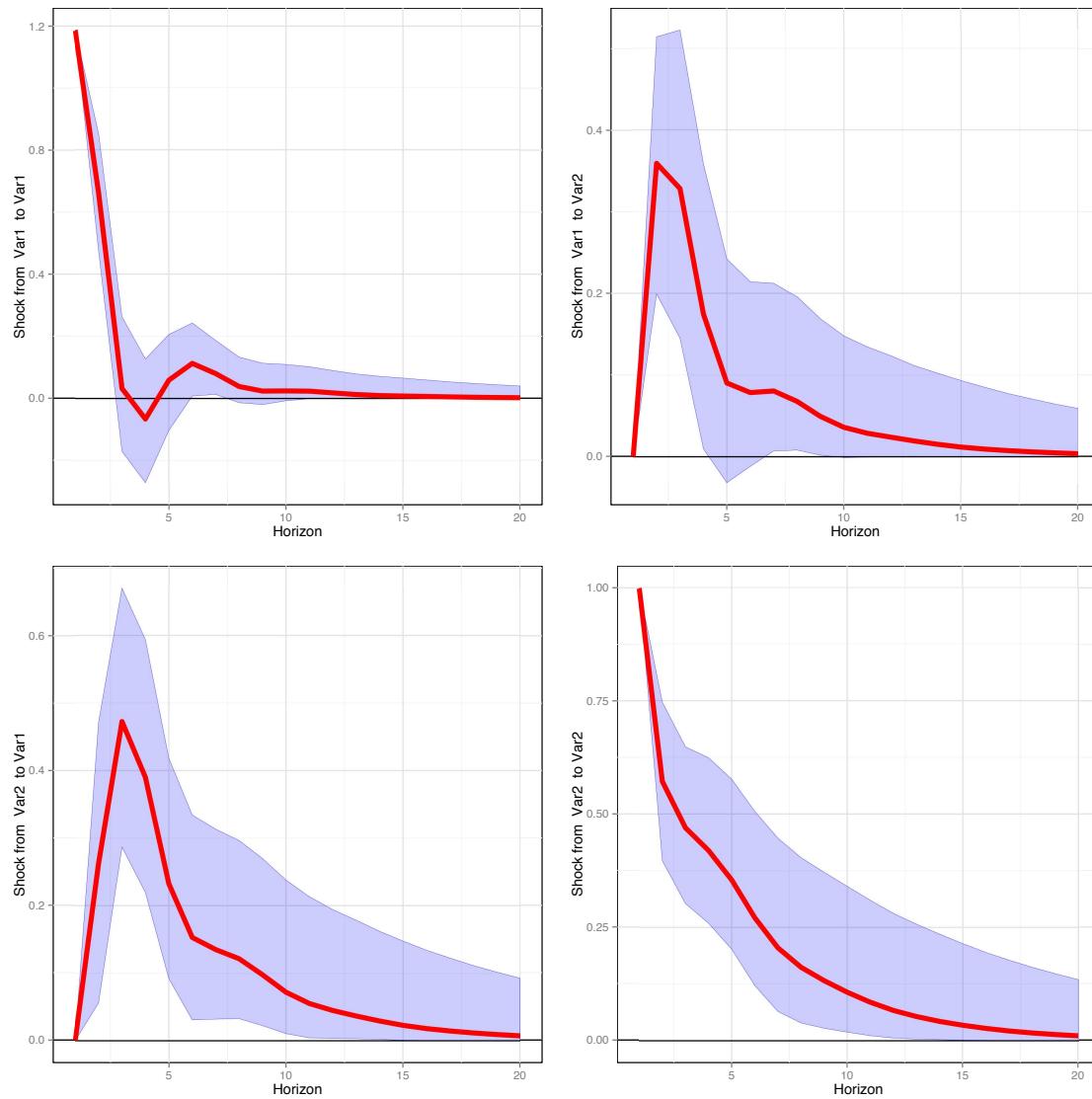


Figure 4: IRFs.

6.2. Monetary Policy VAR

Now to a more ‘real’ example. Perhaps the most common use of VAR-type models in macroeconomics is that of the monetary policy VAR, a nice example of which is Stock and Watson (2001). There, the authors estimate a three-variable VAR(4) model using quarterly inflation, unemployment and the federal funds rate. I will use the same data source as in their paper, but extend the dataset to the present day; this dataset comes packaged with BMR, and the user can the appropriate sample period for estimation, based on their own preferred lag to allow for major data revisions.

The data run from the second quarter of 1954 to the last quarter of 2011, and are constructed as follows. The quarterly inflation rate is based on the chain-weighted price index of GDP, and annualised inflation is defined as $\pi_t = 400 \ln(P_t/P_{t-1})$, where P_t is the price index at time t . Quarterly values of the unemployment and federal funds rates are based on simple averages of the monthly values for that quarter. The full dataset is illustrated in figure 6.

To begin, we examine the properties of each series with the ‘stationarity’ function:

```
data(BMRVARData); stationarity(USMacroData[,2:4],4,8)
KPSS Tests: 4 lags
      INFLATION    UNRATE   FEDFUND 1 Pct 2.5 Pct 5 Pct 10 Pct
Time Trend: 0.6288789 0.2577763 0.8038222 0.216    0.176 0.146 0.119
No Trend:   0.8332977 0.5007845 0.8471684 0.739    0.574 0.463 0.347
```

Augmented Dickey-Fuller Tests:

```
      INFLATION    UNRATE   FEDFUND 1 Pct 2.5 Pct 5 Pct 10 Pct
Time Trend: -3.054973 -3.8956878 -2.600558 -3.99    -3.69 -3.43 -3.13
Constant:   -2.878536 -3.7868331 -2.508190 -3.46    -3.14 -2.88 -2.57
Neither:    -1.473235 -0.6515485 -1.366868 -2.58    -2.23 -1.95 -1.62
```

Number of Diff Lags for ADF Tests:

	Trend	Model	Drift	Model	None
INFLATION	1		1		1
UNRATE	1		1		1
FEDFUND	3		3		3

To this end, we can also use the ‘gacf’ and ‘gpacf’ functions as follows:

```
> gacf(USMacroData[,2:4],lags=12,ci=0.95,plot=T,barcolor="purple",
      names=T,save=T,height=6,width=12)
> gpacf(USMacroData[,2:4],lags=12,ci=0.95,plot=T,barcolor="darkred",
      names=F,save=T,height=6,width=12)
```

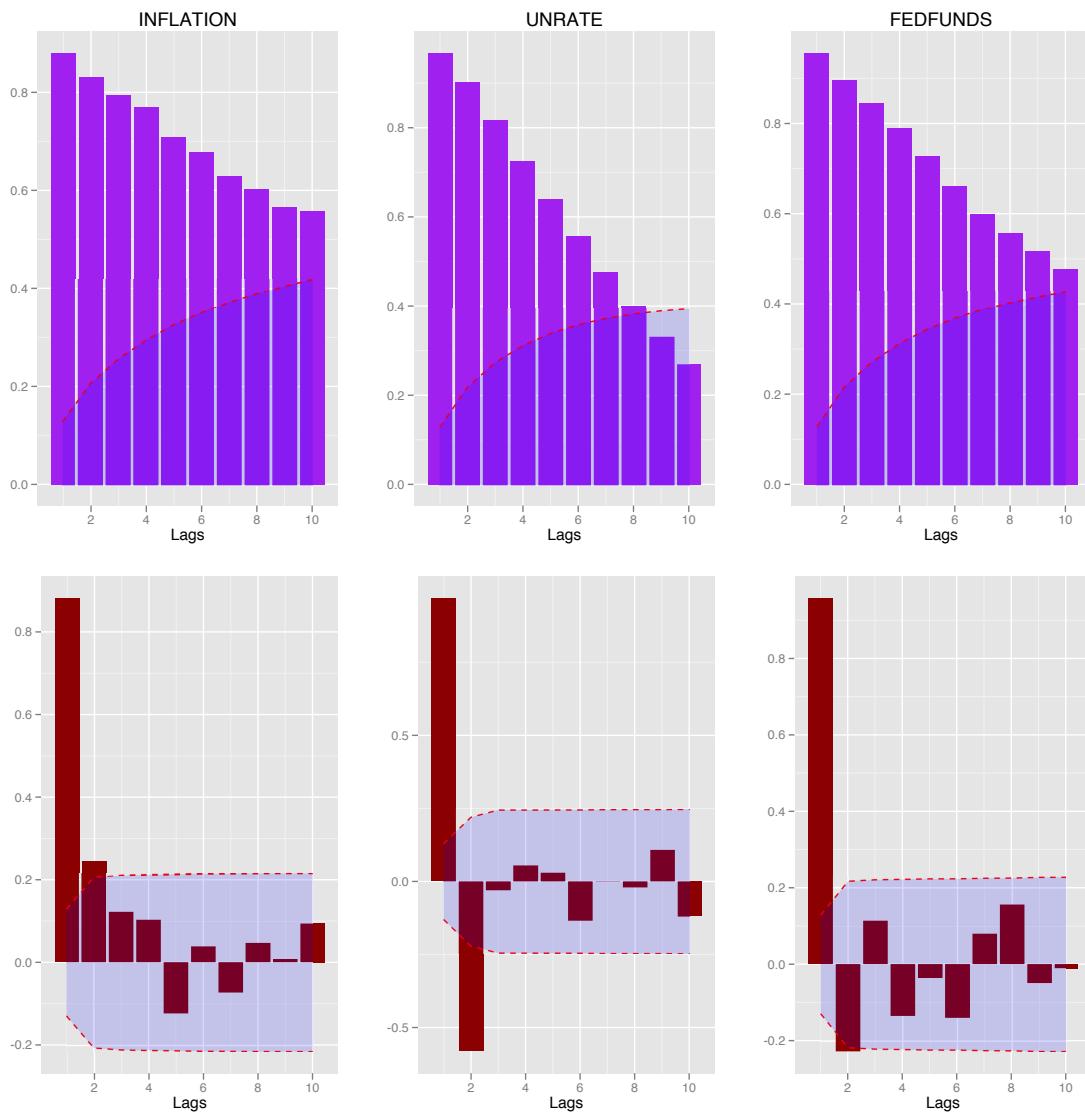


Figure 5: ACF and PACFs.

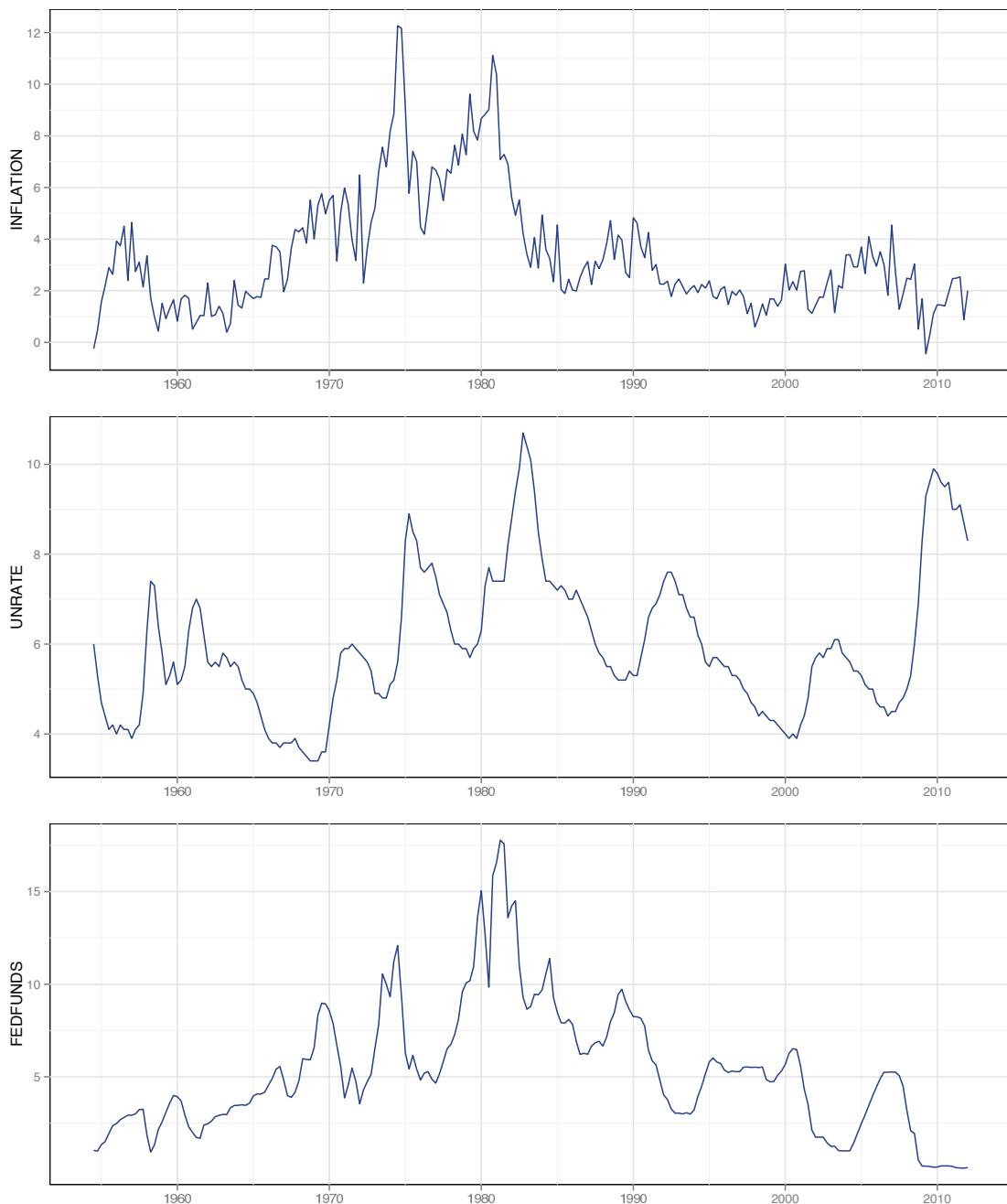


Figure 6: VAR Data.

We now estimate a BVAR with normal-inverse-Wishart prior using the ‘BVARW’ function. As it’s a small model, we will not need more than 1 CPU core, so ‘cores’ is set to 1. First, we set the prior on the first own-lag coefficients to be 0.9, 0.95, and 0.95 for inflation, unemployment and the federal funds rate, respectively; the number of lags to 4; include a constant vector by setting $\text{constant}=T$; set the IRF horizon to 20 quarters; set the number of Gibbs sampling replications to 15000, 5000 being burn-in; set the variance of each prior coefficient to 4, with $\Xi_\beta = 4$; and for the residual covariance matrix Σ , we set the location matrix Ξ_Σ to be the identity matrix (by setting $\Xi_\Sigma = 1$) with 4 degrees of freedom, $\gamma = 4$.

```
testbvarw <- BVARW(USMacroData[,2:4],cores=1,c(0.9,0.95,0.95),
                     p=4,constant=T,irf.periods=20,
                     keep=10000,burnin=5000,
                     XiBeta=4,XiSigma=1,gamma=NULL)

Starting Gibbs C++, Tue Jan 20 10:42:46 2015.
C++ reps finished, Tue Jan 20 10:43:08 2015. Now generating IRFs.
```

As before, the mean of each coefficient is found by accessing \$Beta

```
> testbvarw$Beta
      INFLATION      UNRATE      FEDFUNDS
Constant    0.70225539  0.157218843  0.15365707
Eq 1, lag 1  0.52975134  0.024468380  0.07054064
Eq 2, lag 1 -0.72468233  1.590139740 -0.96468634
Eq 3, lag 1  0.14581331 -0.010235817  1.08208554
Eq 1, lag 2  0.18893757 -0.009580225  0.15781691
Eq 2, lag 2  0.82796252 -0.607117584  1.13306691
Eq 3, lag 2 -0.10223745  0.061954813 -0.43528377
Eq 1, lag 3  0.06878538  0.008277082 -0.06316018
Eq 2, lag 3 -0.07583659 -0.039813286 -0.38637307
Eq 3, lag 3  0.06795957 -0.047048908  0.37002400
Eq 1, lag 4  0.16834972 -0.018506048 -0.04772803
Eq 2, lag 4 -0.11408413  0.015970999  0.19661854
Eq 3, lag 4 -0.11635178  0.009430811 -0.09468271
```

Plot the IRFs, with 5th and 95th percentiles, by using the 'IRF' function.

```
> IRF(testbvarw,percentiles=c(0.05,0.5,0.95),save=T)
```

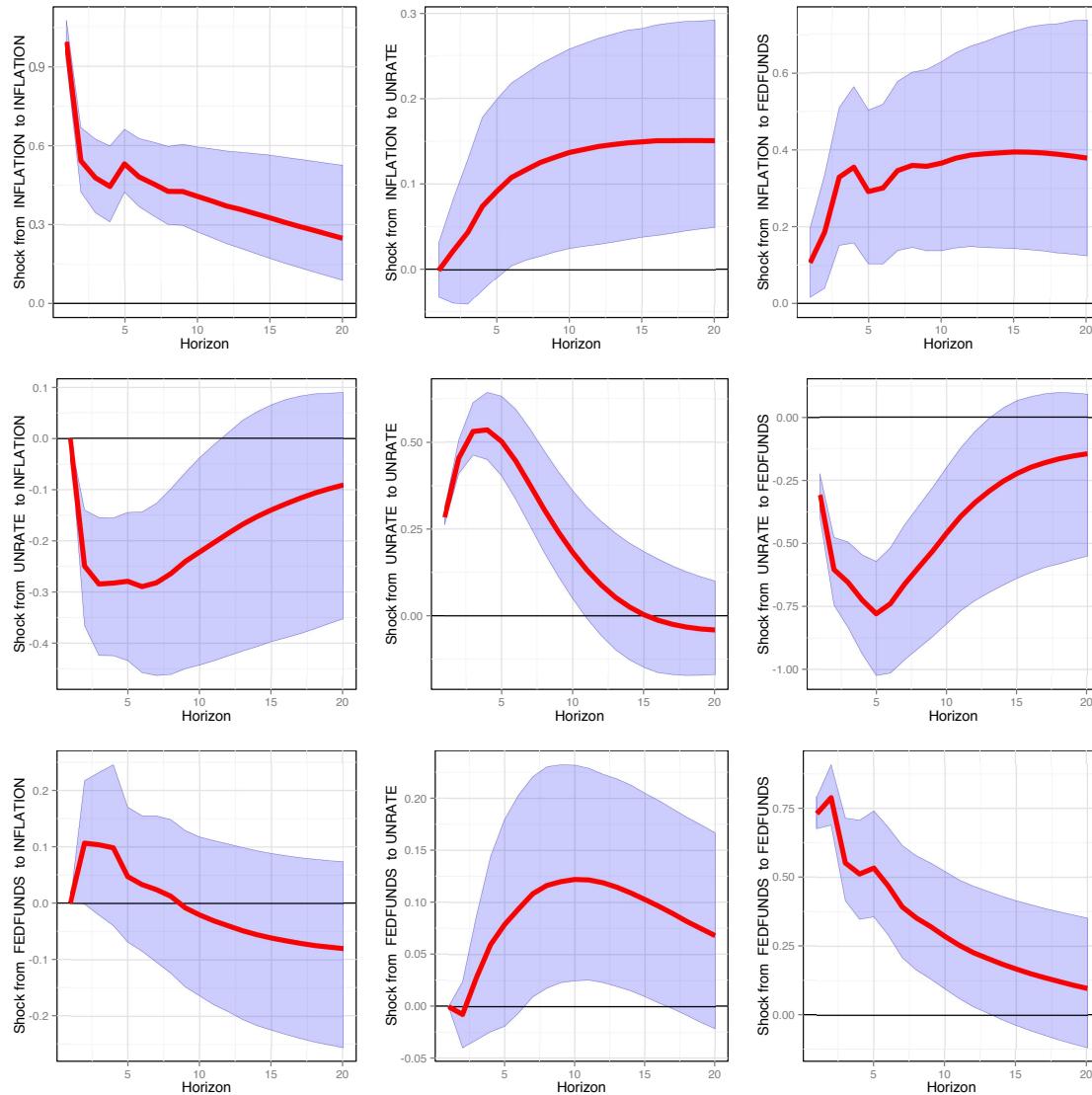


Figure 7: IRFs.

We can plot the posterior distribution of each coefficient, along with the elements of the residual covariance matrix, by using a plot function, where the input is

```
> plot(testbvarw, save=T, height=13, width=13)
```

The coefficient plots are given in figures 8, 9, 10, and 11, and the distribution of the elements of Σ is given in figure 12.

Finally, with the estimated model, we can project forward in time with the ‘forecast’ function. Reestimate the model using data up to the last quarter of 2005,

```
> USMacroData <- USMacroData[1:203,2:4]
> testbvarw <- BVARW(USMacroData,1,c(0.9,0.95,0.95),p=4,constant=T,
  irf.periods=20,keep=10000,burnin=5000,
  XiBeta=4,XiSigma=1,gamma=4)
Starting Gibbs C++, Tue Jun 17 15:59:11 2014.
C++ reps finished, Tue Jun 17 15:59:26 2014. Now generating IRFs.
```

In addition to parameter uncertainty, the ‘shocks=T’ option incorporates uncertainty about future shocks when calculating the percentile bands. ‘backdata=10’ includes the 10 previous ‘real’ data points in the plot, and the dashed line marks where our forecast begins.

```
> forecast(testbvarw,periods=10,shocks=T,plot=T,
  percentiles=c(.05,.50,.95),backdata=10,save=T)
```

and this is shown in figure 13.

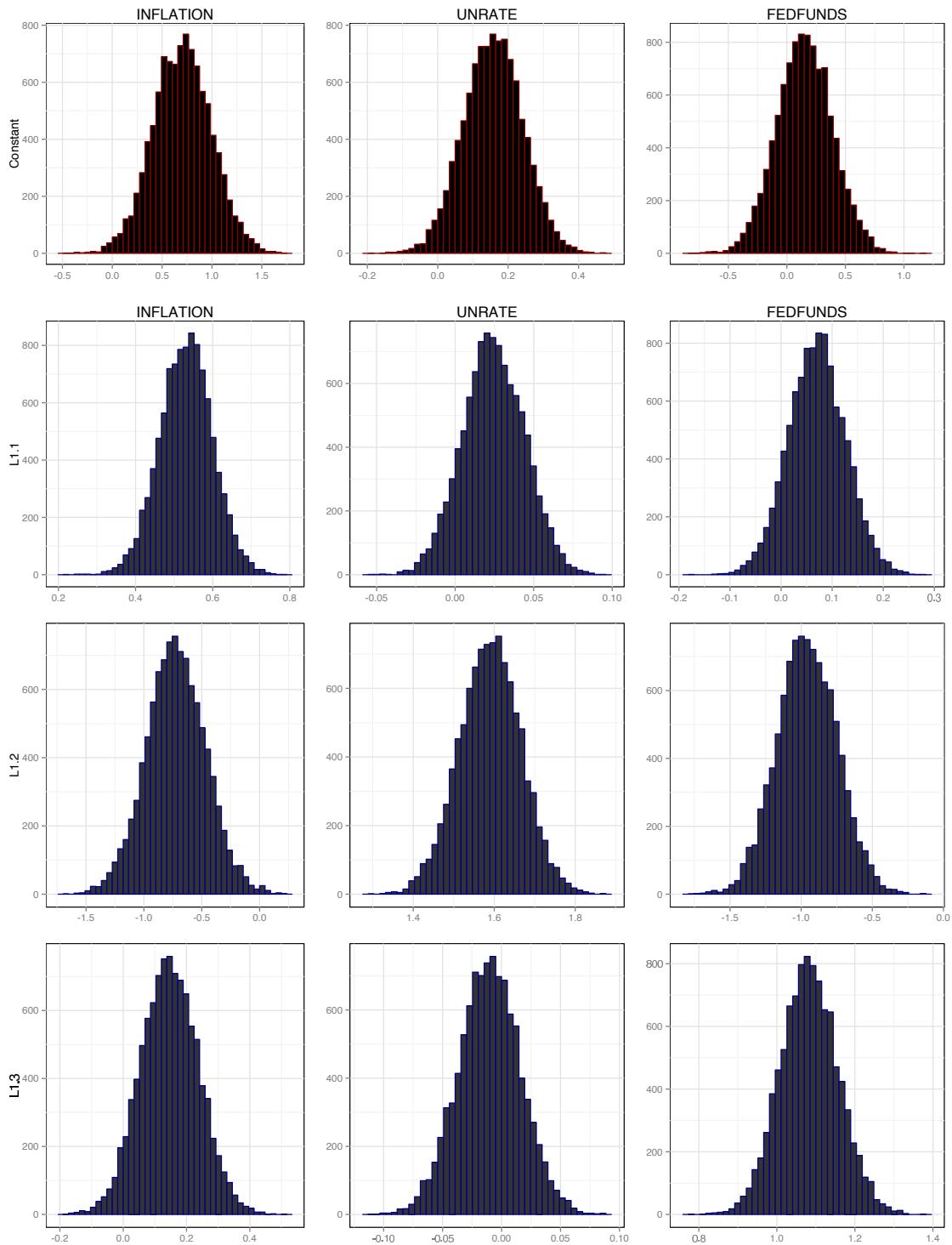


Figure 8: Posterior Distributions of Constants and First Lags.

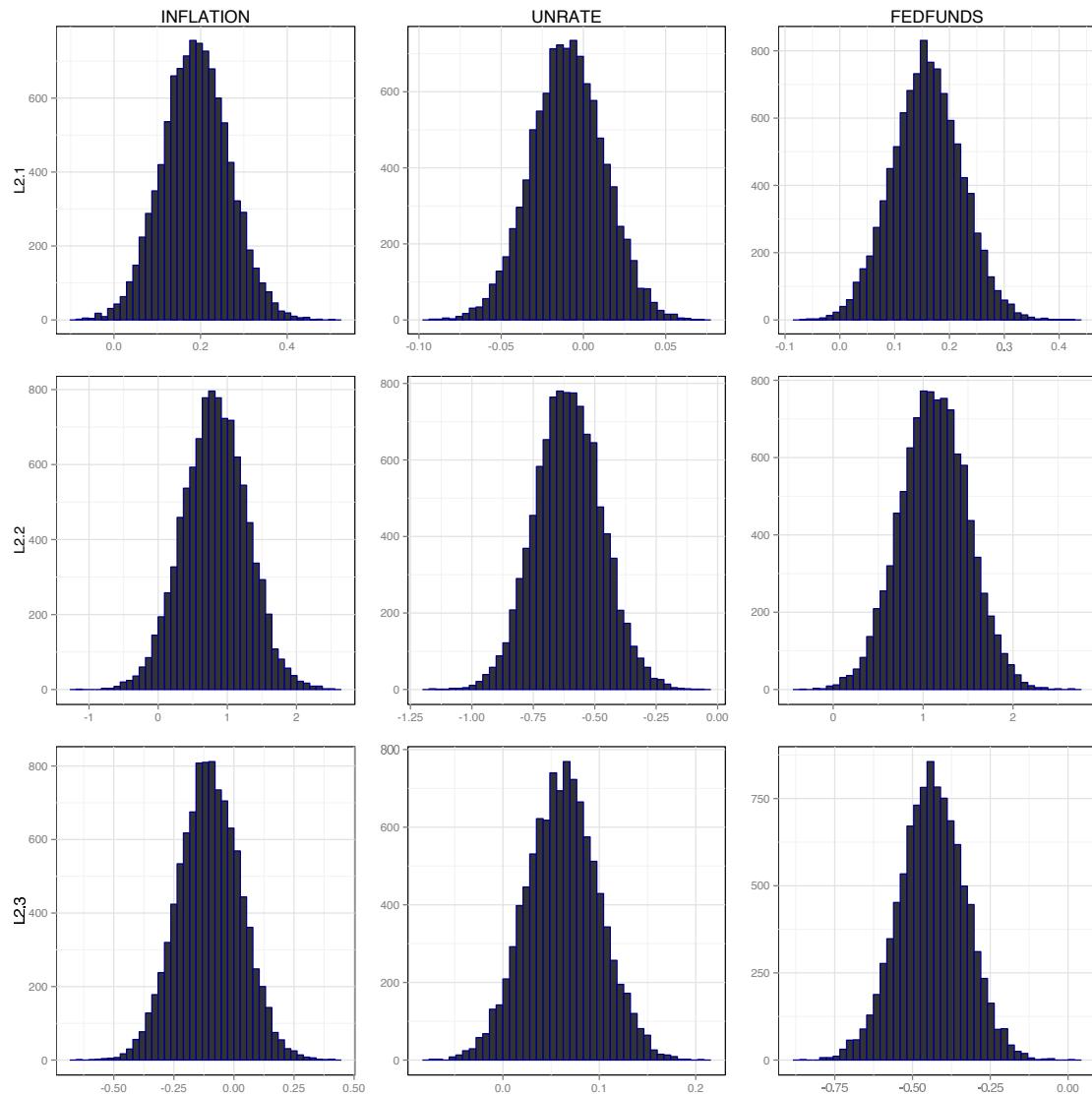


Figure 9: Posterior Distributions of Second Lags.

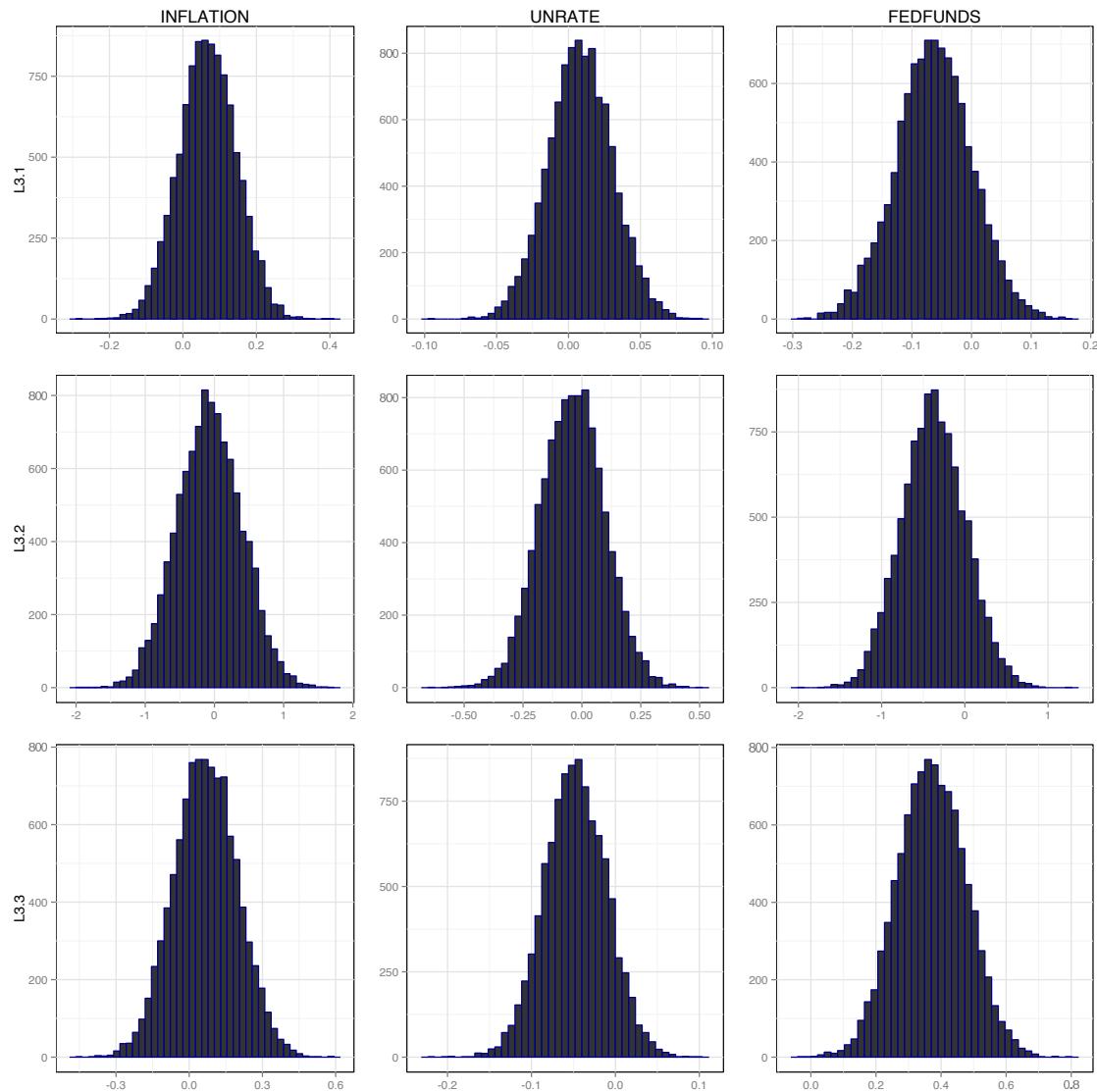


Figure 10: Posterior Distributions of Third Lags.

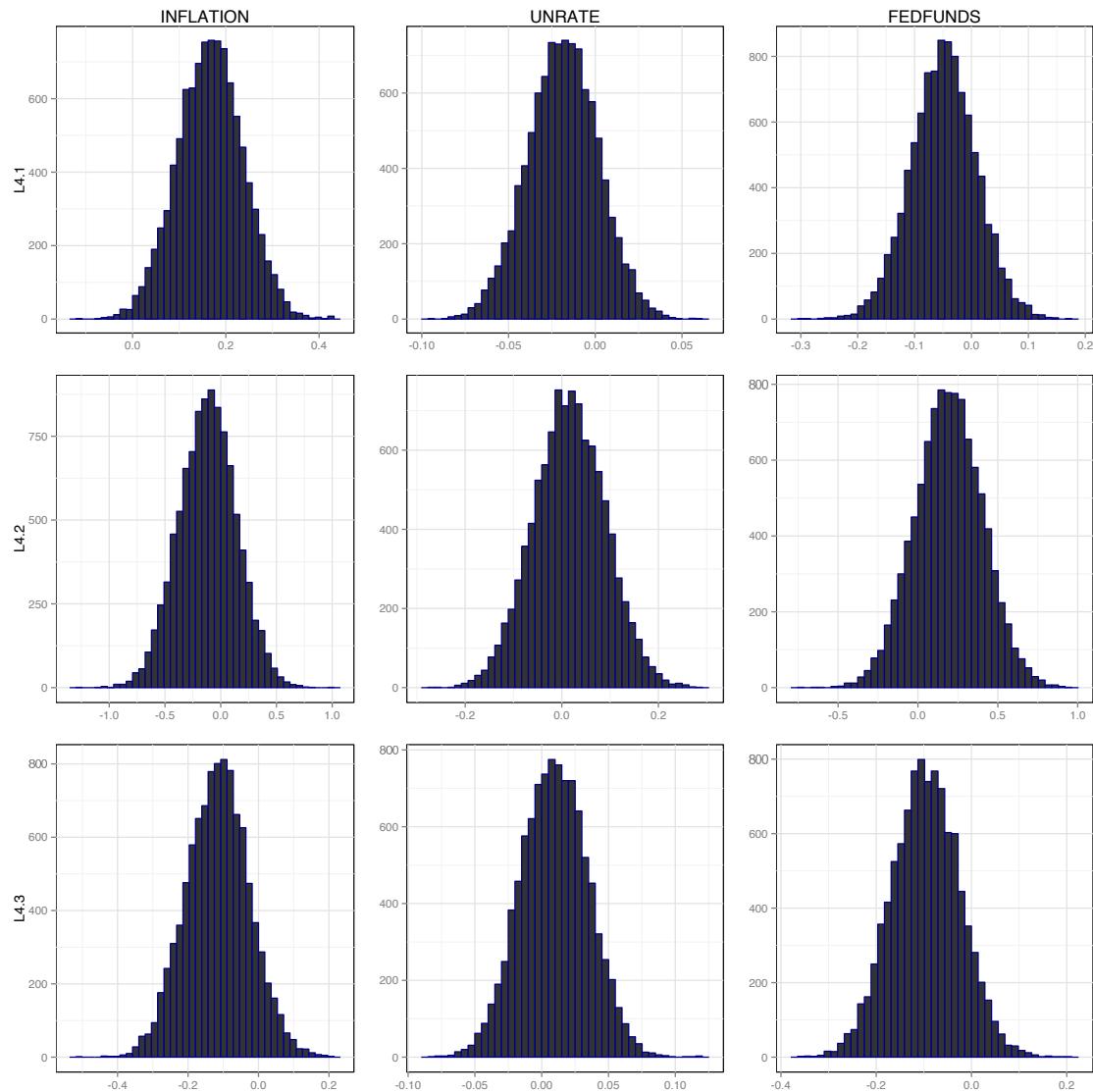


Figure 11: Posterior Distributions of Fourth Lags.

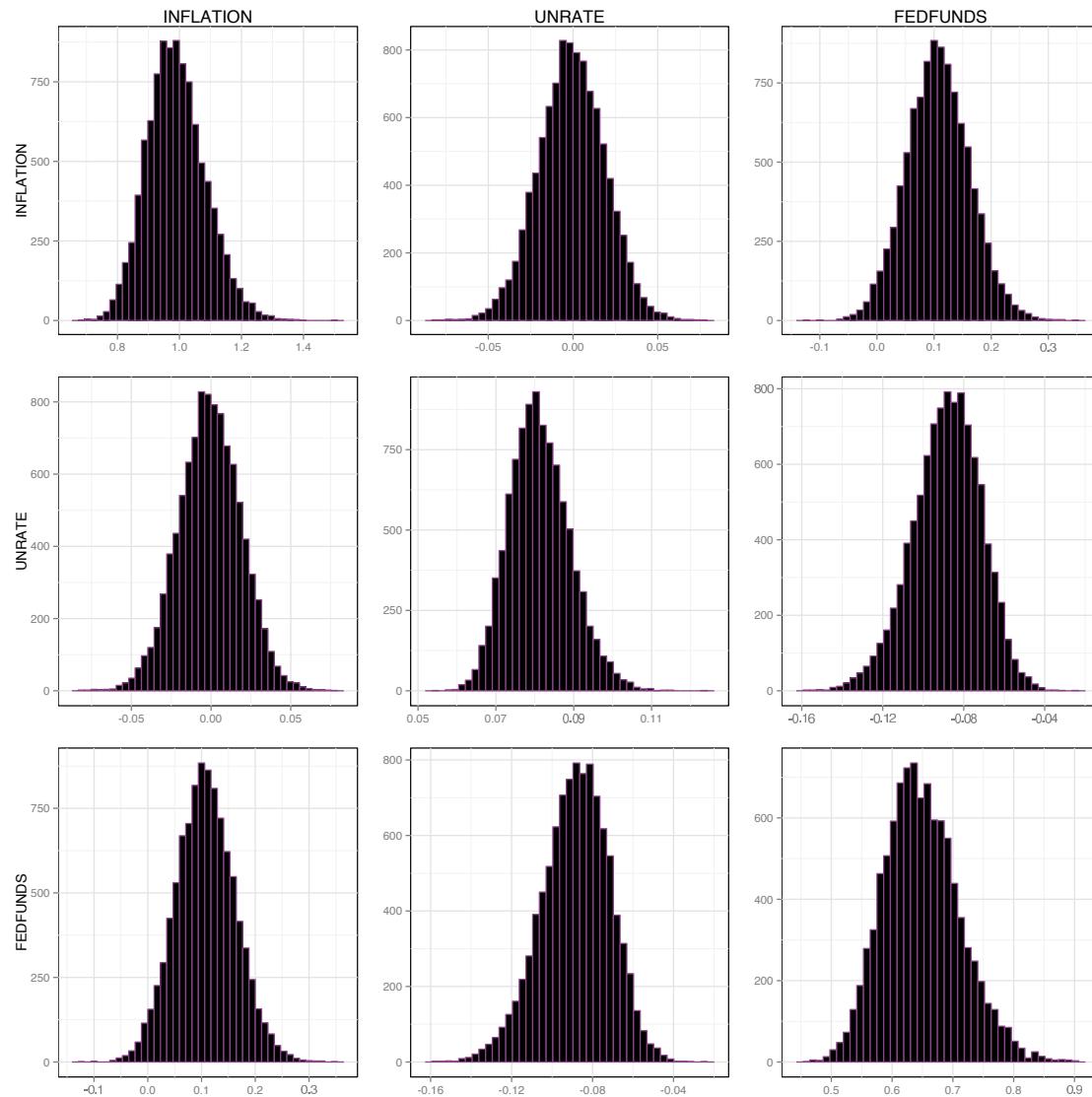


Figure 12: Posterior Distribution of Σ .

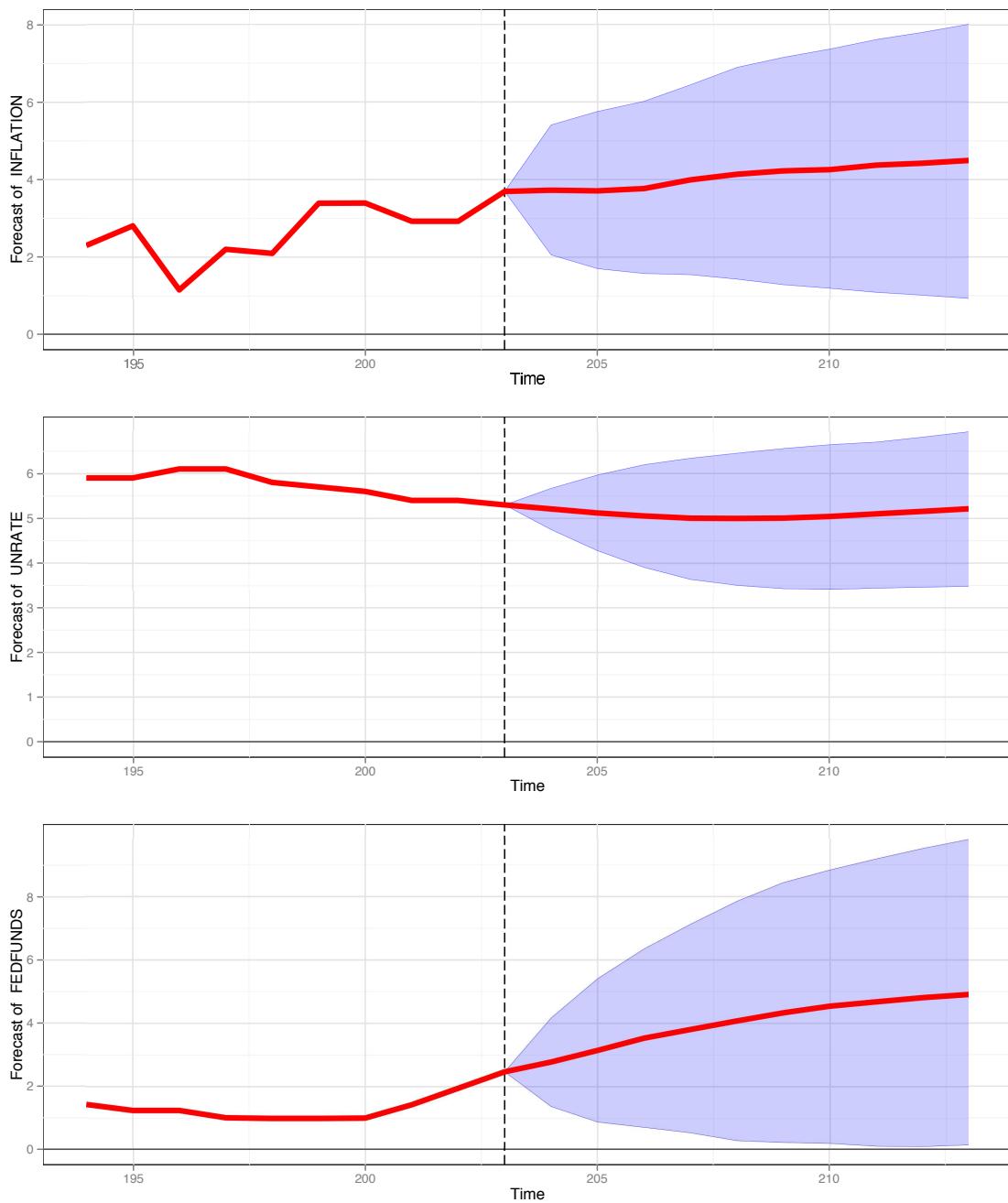


Figure 13: Forecasts.

6.3. Steady State Example

Using the Monte Carlo model from section 6.1, let us briefly illustrate the use of the ‘BVARS’ function. While it is possible to combine our priors of the constant and coefficient matrices to form a prior on the mean, $\Phi_{1 \times m} (\mathbb{I}_m - \sum_{i=1}^2 \beta_i)^{-1} = \Psi_{1 \times m}$, we can instead use the steady-state prior of Villani (2009).

Begin by setting a prior of 0.5 on both first own lag coefficients, and zero on all others, which we’ll call ‘mycfp’. Then set a prior of 18 and 19 on the mean of each series, which we’ll call ‘mypsi’. Then, with a prior variance of 3 on each of the mean values, and hyper parameter values of $H_1 = 0.5$ and $H_4 = 2$, with $m + 1$ prior degrees of freedom on the error covariance matrix (recall that setting $\gamma = \text{NULL}$ will give the default value of $m + 1$), we call the ‘BVARS’ function:

```
mycfp <- c(0.5,0.5)
mypsi <- c(18,19)
testbvars <- BVARS(bvarMCdata,mypsi,mycfp,p=2,irf.periods=20,
                     keep=20000,burnin=5000,
                     XiPsi=3,HP1=0.5,HP4=2,gamma=NULL)
```

We can then check the mean value of Ψ by typing

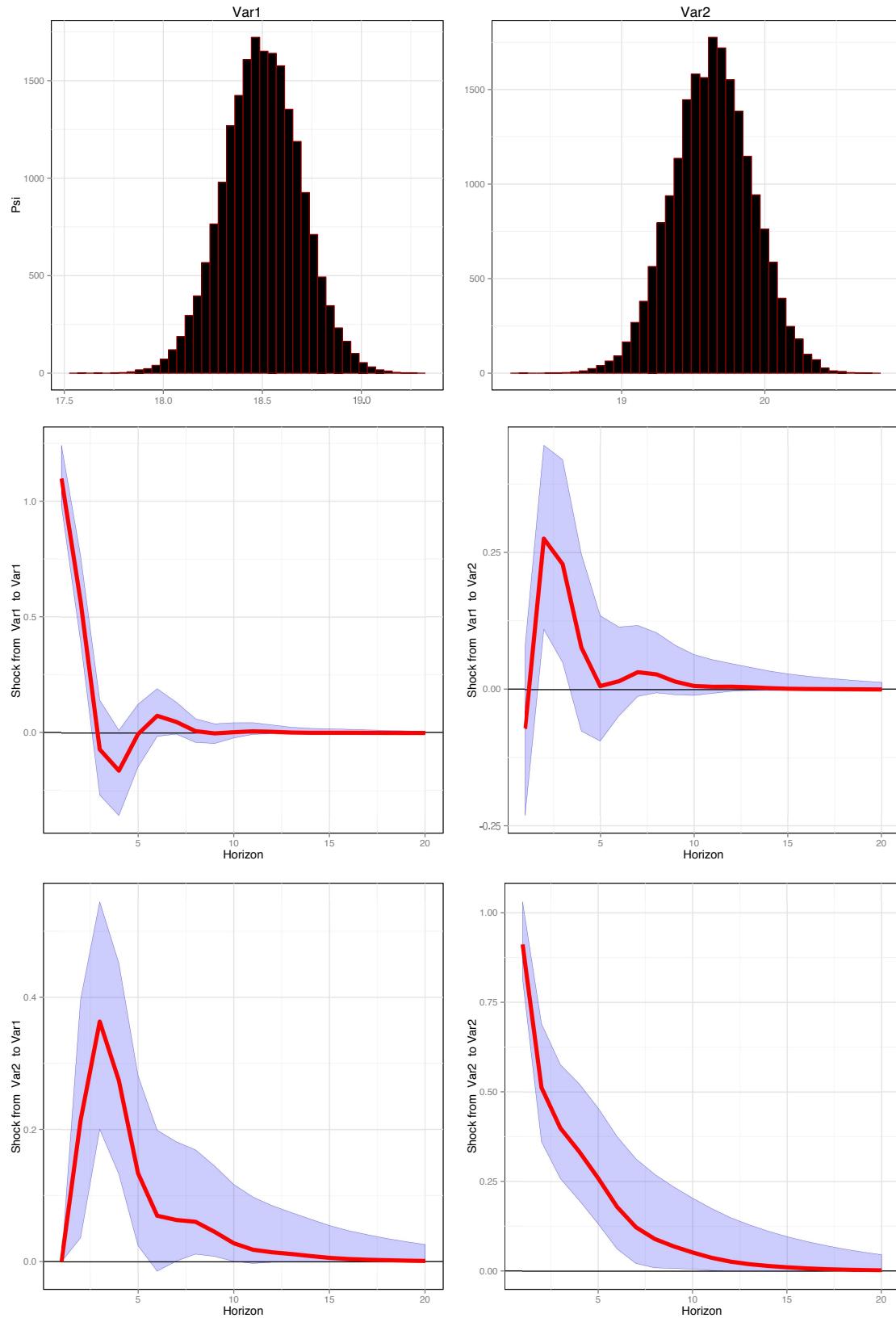
```
testbvars$Psi
      Var1      Var2
Psi 18.49575 19.66013
```

and the mean values for β

```
testbvars$Beta
      Var1      Var2
Eq 1, lag 1 0.5326724 0.28927310
Eq 2, lag 1 0.2345287 0.56842585
Eq 1, lag 2 -0.3978970 -0.07739211
Eq 2, lag 2 0.1433033 0.04712463
```

We can also plot the distribution of Ψ and IRFs, which are qualitatively similar to before.

```
plot(testbvars,save=T)
IRF(testbvars,save=T)
```

Figure 14: Distribution of Ψ and IRFs.

7. Time-Varying Parameters

A less restrictive (though more complicated) approach to Bayesian VAR modelling allows the parameters of the coefficient matrix β (and, in some cases, the parameters of the residual covariance matrix) to vary over time. One macroeconomic motivation for doing so would be that the structure of the economy we're trying to model, and thus the relationships between our variables, may change over time, including key economic and political institutions that form and implement economic policy such as a central bank.

In the context of monetary policy VARs, since 1971 there have been five different Chairmen of the Federal Reserve System, leading to an obvious question of how the mechanism by which monetary policy interacts with the 'real' economy, and the so-called hawkishness of each Chairman, have changed over time, leading to different responses of the economy to monetary policy 'shocks.' Stated another way, have the parameters of the monetary policy reaction function drifted over time? This is a question we can address with BVAR-TVP models.

The model is comprised of two series, the first describing the dynamics of our observable series in a familiar VAR-type fashion, with the second defining the evolution of the parameter vector over time, which is assumed to be an unobserved random walk. We denote this by

$$Y_t = Z_t \beta_t + \varepsilon_{Y,t} \quad (29)$$

$$\beta_t = \beta_{t-1} + \varepsilon_{\beta,t} \quad (30)$$

where $\varepsilon_{Y,t} \sim \mathcal{N}(0, \Sigma)$ and $\varepsilon_{\beta,t} \sim \mathcal{N}(0, Q)$, and, for a given t , matrices: $Y_{m \times 1}$, $Z_{m \times (1+m \cdot p) \cdot m}$, $\beta_{(1+m \cdot p) \cdot m \times 1}$. The joint posterior kernel is

$$p(\beta_{1:T}, Q, \Sigma | Z, Y) \propto p(Y | Z, \beta_{1:T}, Q, \Sigma) p(\beta_0, Q, \Sigma)$$

where the prior densities for Q and Σ are standard, though note that we are now placing a prior on the initial value of β . The prior densities are assumed to be independent, $p(\beta_0, Q, \Sigma) = p(\beta_0)p(\Sigma)p(Q)$, and are given by

$$p(\beta_0) = \mathcal{N}(\bar{\beta}, \Xi_\beta) \quad (31)$$

$$p(\Sigma) = \mathcal{IW}(\Xi_\Sigma, \gamma_\Sigma) \quad (32)$$

$$p(Q) = \mathcal{IW}(\Xi_Q, \gamma_Q) \quad (33)$$

BMR provides two options for specifying $p(\beta_0)$. The first is a user-specified prior exactly the same as in the normal-inverse-Wishart model. The other option is to use a subset of Y , $Y_{1:\tau}$, as a training sample to initialise estimation of $\beta_{\tau+1:T}$. In the latter case, $\bar{\beta}$ becomes the OLS estimate of β ,

$$\hat{\beta} = \left(\sum_{t=1}^{\tau} Z_t^\top Z_t \right)^{-1} \left(\sum_{t=1}^{\tau} Z_t^\top Y_t \right),$$

then Ξ_β serves to scale the variance of the OLS estimate,

$$V(\hat{\beta}) = \frac{1}{\tau - (1 + m \cdot p)} \left(\sum_{t=1}^{\tau} Z_t^\top \left(\sum_{s=1}^{\tau} \varepsilon_{Y,s} \varepsilon_{Y,s}^\top \right)^{-1} Z_t \right)^{-1},$$

and the location matrix of Q becomes $\Xi_Q \cdot \tau$ times the variance of the OLS estimate. Thus, when $\tau > 0$, the three prior densities become:

$$p(\beta_0) = \mathcal{N}(\hat{\beta}, \Xi_\beta \cdot V(\hat{\beta})) \quad (34)$$

$$p(\Sigma) = \mathcal{IW}(\Xi_\Sigma, \gamma_\Sigma) \quad (35)$$

$$p(Q) = \mathcal{IW}(\Xi_Q \cdot \tau \cdot V(\hat{\beta}), \gamma_Q) \quad (36)$$

The standard factorisation of the posterior density kernel still applies, so we can implement a Gibbs-style sampler. The conditional posterior distributions of Σ and Q are

$$p(\Sigma | \beta_{1:T}, Q, Z, y) = \mathcal{IW}\left(\Xi_\Sigma + \sum_{t=1}^T (y_t - Z_t \beta_t)(y_t - Z_t \beta_t)^\top, T + \gamma_\Sigma\right) \quad (37)$$

$$p(Q | \beta_{0:T}, \Sigma, Z, y) = \mathcal{IW}\left(\Xi_Q + \sum_{t=1}^T (\Delta \beta_t)(\Delta \beta_t)^\top, T + \gamma_Q\right) \quad (38)$$

where Δ is the first-difference operator. The conditional posterior distribution of the coefficient matrix $\beta_{1:T}$, $p(\beta_{1:T} | \Sigma, Q, \mathcal{Y}^T)$, is a little more complicated, however.

Let $\mathcal{Y} = \mathcal{Y}^T = \{\mathcal{Y}_t\}_{t=1}^T$ be the set of all observations, where a superscript (\mathcal{Y}^t) denotes all information up to and including time t , whereas a subscript implies a specific observation at time t . We can factorise the joint conditional posterior density of $\beta_{1:T}$ to

$$p(\beta_{1:T} | \Sigma, Q, \mathcal{Y}^T) = p(\beta_T | \Sigma, Q, \mathcal{Y}^T) \prod_{t=1}^{T-1} p(\beta_t | \beta_{t+1}, \Sigma, Q, \mathcal{Y}^t) \quad (39)$$

where

$$p(\beta_T | \Sigma, Q, \mathcal{Y}^T) = \mathcal{N}(\beta_T, P_T) \quad (40)$$

$$p(\beta_t | \beta_{t+1}, \Sigma, Q, \mathcal{Y}^t) = \mathcal{N}(\beta_{t|t+1}, P_{t|t+1}) \quad (41)$$

are the conditional predictive densities, with

$$\beta_{t|t-1} = \mathbb{E}(\beta_t | \mathcal{Y}^{t-1}, \Sigma, Q)$$

$$P_{t|t-1} = V(\beta_t | \mathcal{Y}^{t-1}, \Sigma, Q)$$

(See, for example, Carter and Kohn (1994) for a technical discussion, and Cogley and Sargent (2005) and Primiceri (2005) for applications and very detailed appendices.)

For each sampling iteration, we evaluate $p(\beta_{1:T} | \Sigma, Q, \mathcal{Y}^T)$ by applying a Kalman filter from $t = 0$ to T , then a backwards recursion to obtain a smoothed estimate. (See the section on DSGE estimation for an overview of the steps involved in the Kalman filter, equations (67) through (73).) Following Koop and Korobilis (2010), BMR utilises the Durbin and Koopman (2002) simulation smoother to do this. The process is divided into three sections. We begin with a Kalman filter recursion to find $\hat{\beta}_{1:T}$.

Given $\beta_{t+1|t} = \beta_{t|t}$ and $P_{t+1|t} = P_{t|t} + Q$, and some initial values, β_0 and P_0 , find the residual, its covariance matrix, and Kalman gain matrix, given by

$$\begin{aligned} \epsilon_{t+1} &= \mathcal{Y}_{t+1} - Z_{t+1}\beta_{t+1|t} \\ \Sigma_{\epsilon,t+1} &= Z_{t+1}P_{t+1|t}Z_{t+1}^\top + \Sigma \\ K_{t+1} &= P_{t+1|t}Z_{t+1}^\top \Sigma_{\epsilon,t+1}^{-1} \end{aligned}$$

respectively. Update the predicted β_{t+1} and P_{t+1} series with $t + 1$ information,

$$\beta_{t+1|t+1} = \beta_{t+1|t} + K_{t+1}\epsilon_{t+1}$$

$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1}Z_{t+1}P_{t+1|t}$$

Then, using the filtered series, obtain smoothed estimates of the disturbances

$$\begin{aligned}\hat{\epsilon}_{Y,t} &= \Sigma \Sigma_{\epsilon,t}^{-1} \epsilon_t - \Sigma K_t^\top r_t \\ \hat{\epsilon}_{\beta,t} &= Q r_t\end{aligned}$$

where r_t is given by

$$r_{t-1} = Z_t \Sigma_{\epsilon,t}^{-1} \epsilon_t + r_t - Z_t^\top K_t^\top r_t$$

such that $r_T = \mathbf{0}$. A smoothed estimate of β_{t+1} is then given by iterating

$$\hat{\beta}_t = \hat{\beta}_{t-1} + Q r_t$$

forwards $\forall t$.

The simulation smoother requires three series of β to give a final estimate. Save $\hat{\beta}$ from an initial Kalman filter and smoothing run. Then sample new disturbance series, $\epsilon_{Y,t}$ and $\epsilon_{\beta,t}$, using the fact that both are normally distributed with covariance matrices Σ and Q , respectively, and denote these new series with a plus ('+')'. Using $\epsilon_{Y,t}^{(+)}$ and $\epsilon_{\beta,t}^{(+)}$, generate $Y^{(+)}$ and $\beta^{(+)}$ by the methods above, and store $\beta^{(+)}$. With $Y^{(+)}$, find $\hat{\beta}^{(+)}$ by the same filtering and smoothing run. The h th β draw is then given by

$$\beta^{(h)} = \hat{\beta} - (\hat{\beta}^{(+)} - \beta^{(+)})$$

However, as noted in (Durbin and Koopman, 2002, p. 607), we can simplify this process by generating the series $Y^{(*)} = Y - Y^{(+)}$, which requires only one set of smoothed estimates of β , $\hat{\beta}^{(*)}$, and this leads to a substantial improvement in computationally efficiency for estimating the BVAR-TVP model. Therefore, for the h th β draw, BMR constructs

$$\beta^{(h)} = \hat{\beta}^{(*)} + \beta^{(+)}$$
 (42)

and proceeding to draw $Q^{(h)}$ and $\Sigma^{(h)}$ is standard.

7.1. BVARTVP Function

```
BVARTVP(mydata, timelab=NULL, coefprior=NULL, tau=NULL, p=4,
         irf.periods=20, irf.points=NULL,
         keep=10000, burnin=5000,
         XiBeta=4, XiQ=0.01, gammaQ=NULL,
         XiSigma=1, gammaS=NULL)
```

- **mydata**

A matrix or data frame containing the data series used for estimation; this should be of size $T \times m$.

- **timelab**

This is a numeric vector of length T that provides labels for the observations. This is useful for selecting which IRFs to compute.

- **coefprior**

A numeric vector of length m , matrix of size $(m \cdot p + 1) \times m$, or a value ‘NULL’, that contains the prior mean-value of each coefficient. Providing a numeric vector of length m will set a zero prior on all coefficients except the own first-lags, which are set according to the elements in ‘coefprior’. Setting this input to ‘NULL’ will give a random-walk-in-levels prior.

Note that, when τ is set to ‘NULL’, this input becomes the initial draw for the sampling algorithm, and starting with an explosive draw might be a bad idea.

- **tau**

τ is the length of the training-sample prior. If this is set other than ‘NULL’ it will replace ‘coefprior’ above with the coefficients from a pre-sampling estimation run. Selecting this option also affects the ‘XiBeta’ choice below.

- **p**

The number of lags to include of each variable, where $p \in \mathbb{Z}_{++}$. The default value is 4.

- **irf.periods**

An integer value for the horizon of the impulse response calculations, which must be great than zero. The default value is 20.

- **irf.points**

A numeric vector of length $(0, T]$. If the user supplied a ‘timelab’ list above, then this vector should contain points corresponding to that list. The default of ‘NULL’ will mean that all IRFs, for $T - \tau$, will be computed. The IRFs are stored in a 5 dimensional array of size

$$\text{irf.periods} \times m \times m \times \text{length(irf.points)} \times \text{keep}$$

If the number of variables, replications, and/or observations is quite large, then calculating all IRFs will take up a lot of memory. For example, with an IRF horizon of 20, 3 variables, 200 observations, training sample size of 50, and 50000 post-burn-in replications, we have 1,350,000,000 elements to store.

- **keep**

The number of Gibbs sampling replications to keep from the sampling run.

- **burnin**

The sample burn-in length for the Gibbs sampler.

- **XiBeta**

A numeric vector of length 1 or matrix of size $(m \cdot p + 1) \cdot m \times (m \cdot p + 1) \cdot m$ that contains the prior covariance of each coefficient for β_0 . If the user supplies a scalar value, then Ξ_β is $\mathbb{I}_{(m \cdot p + 1)m} \cdot \Xi_\beta$. The structure of Ξ_β corresponds to $\text{vec}(\beta)$.

Note that if $\tau \neq \text{NULL}$, ‘XiBeta’ should be a numeric vector of length 1 that scales the OLS estimate the covariance matrix of $\hat{\beta}$.

- **XiQ**

A numeric vector of length 1 or matrix of size $(m \cdot p + 1) \cdot m \times (m \cdot p + 1) \cdot m$ that contains the location matrix of the inverse-Wishart prior on Q . If the user provides a scalar value, then Ξ_Q is $\mathbb{I}_{(m \cdot p + 1)m} \cdot \Xi_Q$.

- **gammaQ**

A numeric vector of length 1 corresponding to the prior degrees of freedom of the Q matrix. The minimum value is $(m \cdot p + 1)m + 1$, and this is the default value, unless $\tau \neq \text{NULL}$, in which case $\gamma_S = \tau$.

- **XiSigma**

A numeric vector of length 1 or matrix of size $m \times m$ that contains the location matrix of the inverse-Wishart prior on Σ . If the user provides a scalar value, then Ξ_Σ is $\mathbb{I}_{(m \cdot p + 1)m} \cdot \Xi_\Sigma$.

- **gammaS**

A numeric vector of length 1 corresponding to the prior degrees of freedom of the error covariance matrix. The minimum value is $m + 1$, and this is the default value.

The function returns an object of class ‘BVARTVP’, which contains:

- **Beta**

A matrix of size $(m \cdot p + 1) \cdot m \times (T - \tau)$ containing the posterior mean of the coefficient matrix, β , in vectorised form, for $(\tau + 1) : T$.

- **BDraws**

An array of size $(m \cdot p + 1) \times m \times \text{keep} \times (T - \tau)$ which contains the post burn-in draws of β .

- **Q**

A matrix of size $(m \cdot p + 1) \cdot m \times (m \cdot p + 1) \cdot m$ containing the mean posterior estimates of the covariance matrix of ε_β (Q).

- **QDraws**

An array of size $(m \cdot p + 1) \cdot m \times (m \cdot p + 1) \cdot m \times \text{keep}$ which contains the post burn-in draws of Q .

- **Sigma**

A matrix of size $m \times m$ containing the mean posterior estimates of the residual covariance matrix (Σ).

- **SDraws**

An array of size $m \times m \times \text{keep}$ which contains the post burn-in draws of Σ .

- **IRFs**

Let $\ell = \text{number of ‘irf.points’}$ the user selected. ‘IRFs’ is then a five-dimensional object of

size $\text{irf.periods} \times m \times m \times \ell \times \text{keep}$ containing the impulse response function calculations; the first m refers to the responses to the last m shock.

- `data`

The data used for estimation.

- `irf.points`

The points in the sample where the user elected to produce IRFs.

- `tau`

The length of the training sample.

7.2. BVARTVP Example

Continuing with our updated Stock and Watson (2001) dataset, we will estimate a BVARTVP model. Let's pick 1979 as the first year to plot IRFs for, coinciding with Paul Volcker's appointment as Chairman of the Fed, then pick a year every 9 years going forward (that is, 1988, 1997, and 2006).

```
> irf.points<-c(1979,1988,1997,2006)
> yearlab<-seq(1955.00,2010.75,0.25)
> USMacroData<-USMacroData[3:226,2:4]
```

For estimation, we will use a training sample to initialise posterior sampling, using the first 80 quarters of our data to do so. As in the normal-inverse-Wishart case, select 4 lags, and 20 quarters as the IRF horizon. Run the sampler for 70,000 draws, 40,000 of which is sample burn-in. Finally, the parameters of the prior densities are: $\Xi_\beta = 4$, $\Xi_Q = 0.005$, and $\Xi_\Sigma = 1$, with $\gamma_Q = 81$ and $\gamma_\Sigma = 4$.

```
> bvartvptest <- BVARTVP(USMacroData,timelab=yearlab,
                           coefprior=NULL,tau=80,p=4,
                           irf.periods=20,irf.points=irf.points,
                           keep=30000,burnin=40000,
                           XiBeta=4,XiQ=0.005,gammaQ=NULL,
                           XiSigma=1,gammaS=NULL)
```

Finished Prior

Starting Gibbs C++, Tue Jun 17 17:28:34 2014.

C++ reps finished, Tue Jun 17 17:33:49 2014. Now generating IRFs.

Post-estimation, we can plot the posterior distribution of $\beta_{\tau:T}$ using

```
> plot(bvartvptest,save=T)
```

The four IRFs, with a median-based comparison of each, are given by the familiar IRF function,

```
> IRF(bvartvptest,save=T)
```

These are illustrated below.

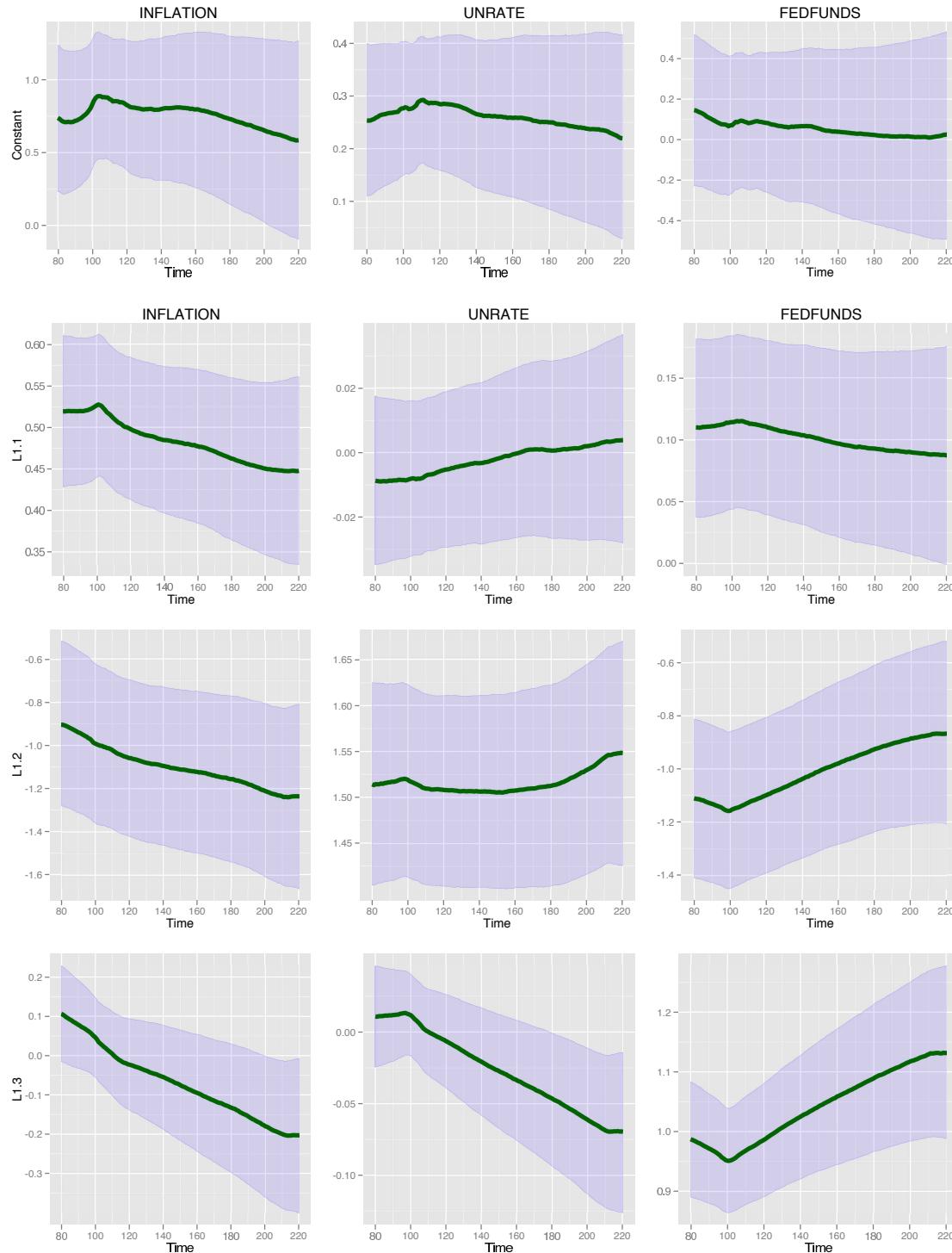


Figure 15: Posterior Distributions of Constants (Top) and First Lags Over Time.

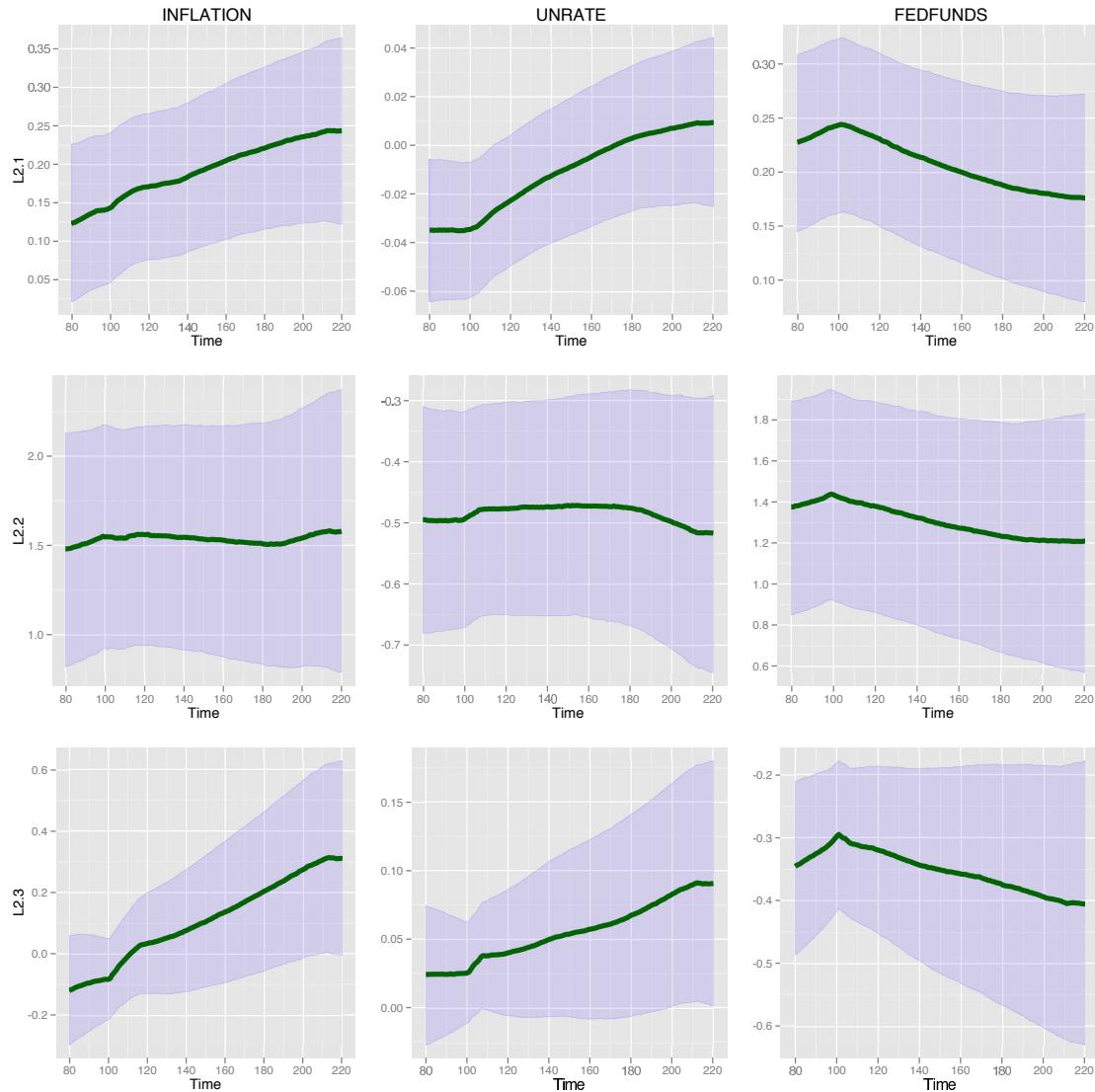


Figure 16: Posterior Distributions of Second Lags Over Time.

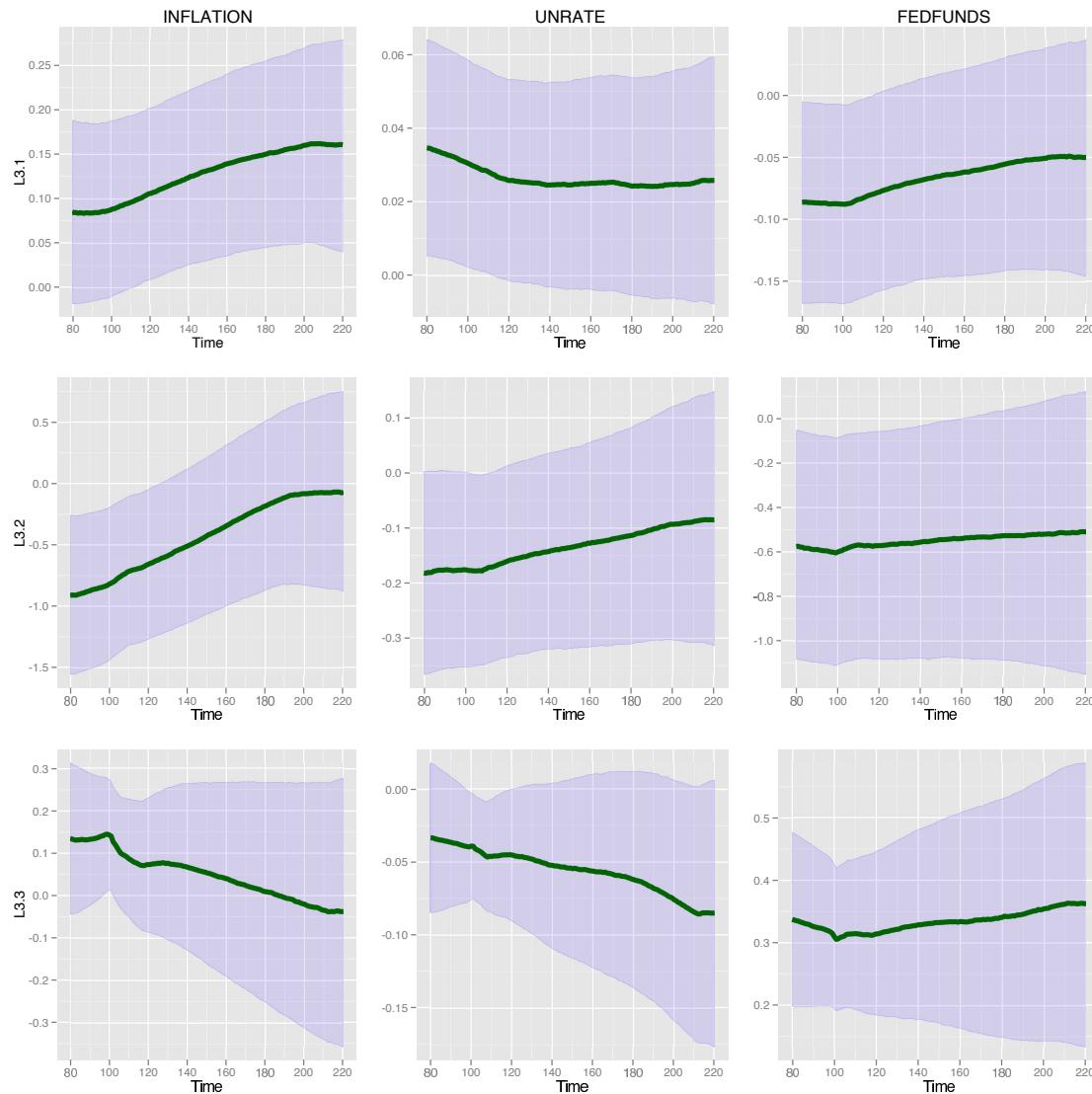


Figure 17: Posterior Distributions of Third Lags Over Time.

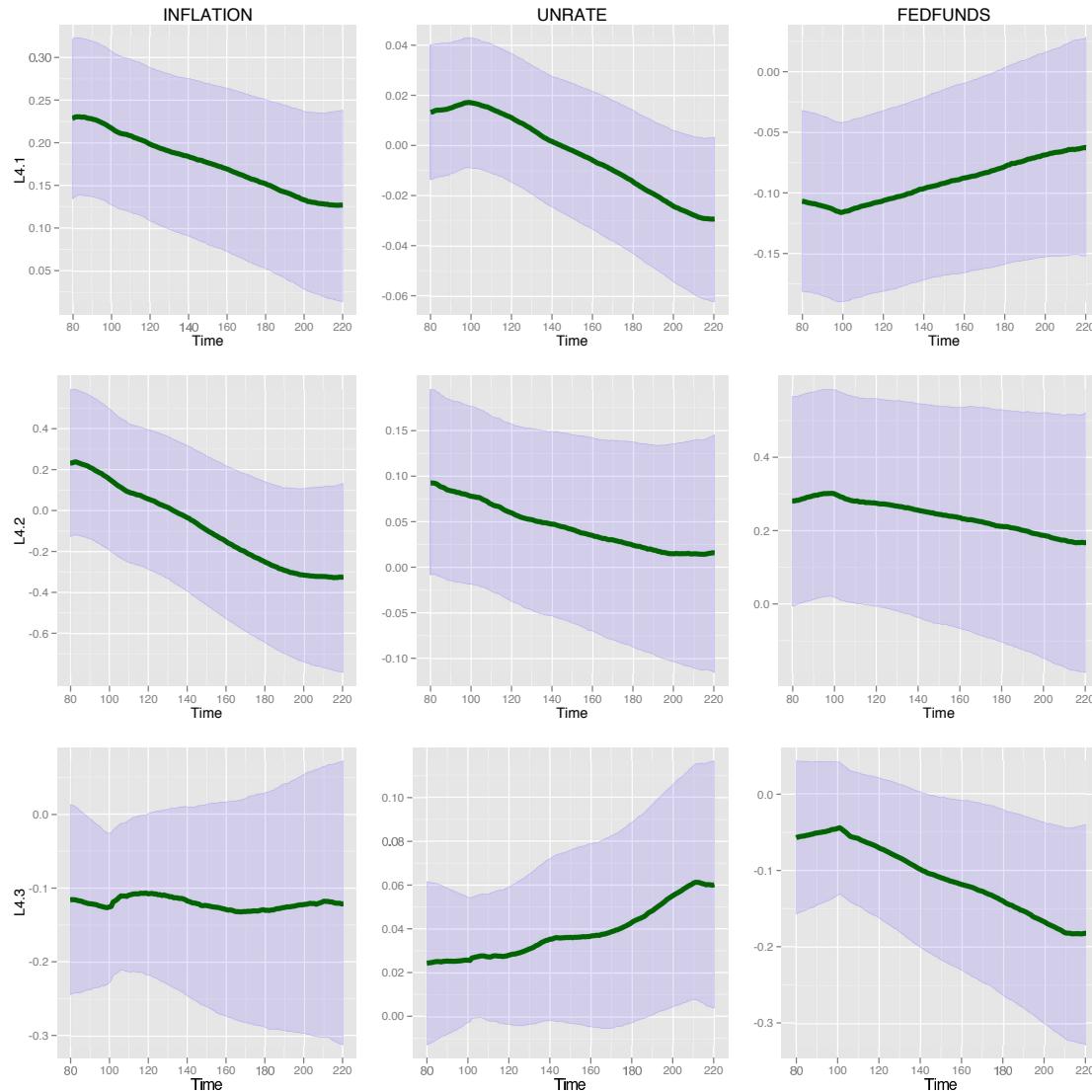


Figure 18: Posterior Distributions of Fourth Lags Over Time.

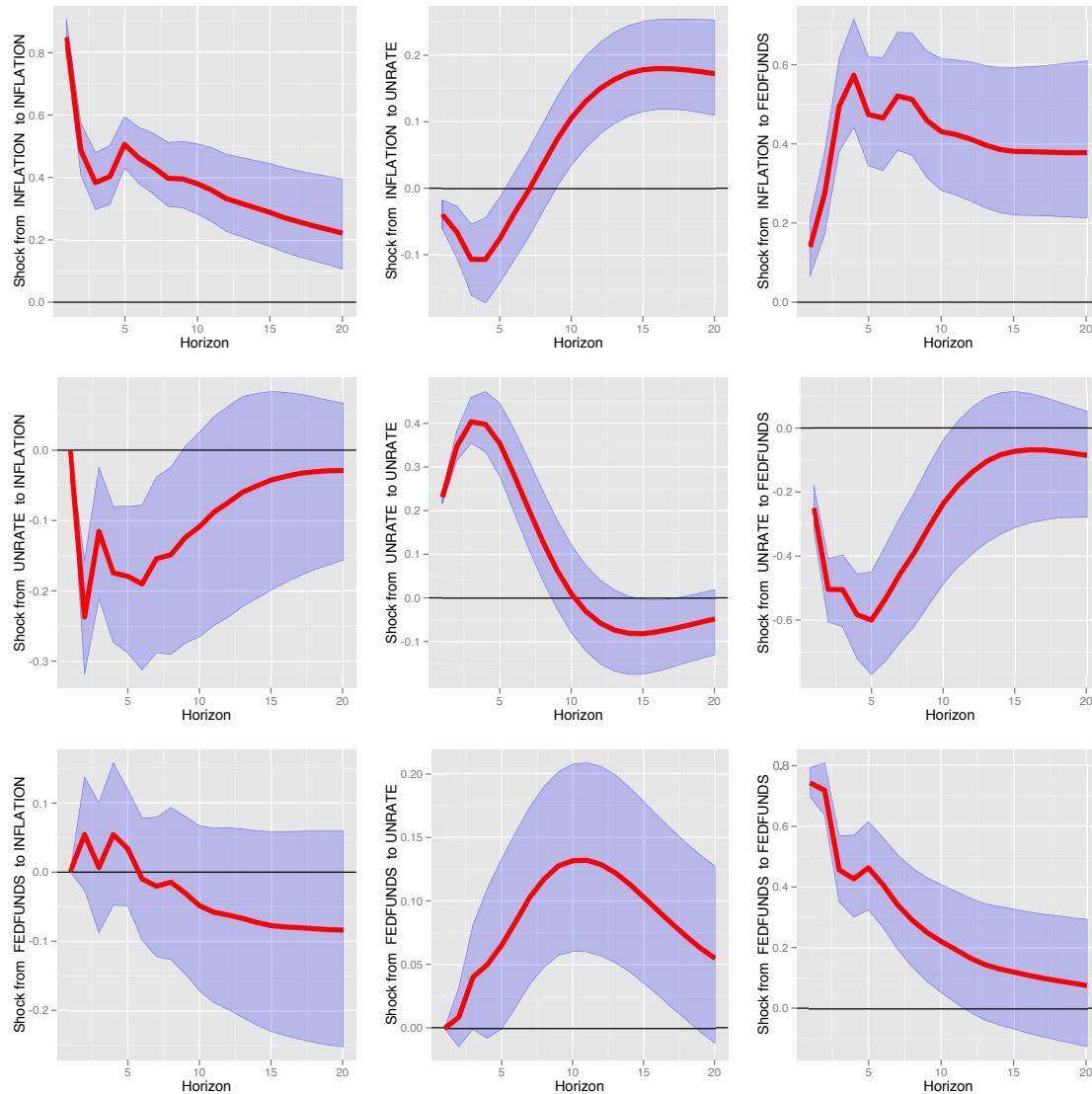


Figure 19: IRFs at 1979.

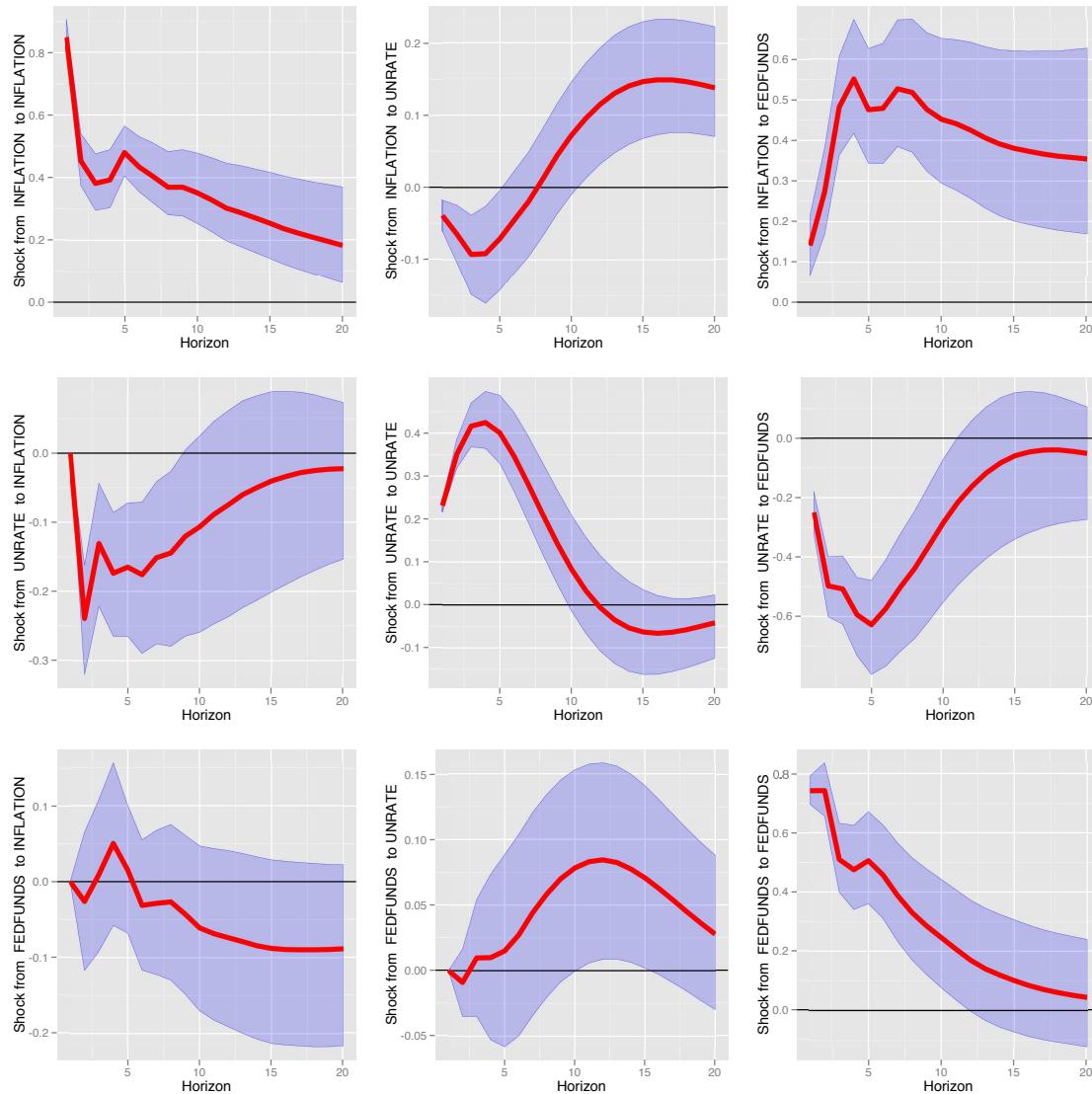


Figure 20: IRFs at 1988.

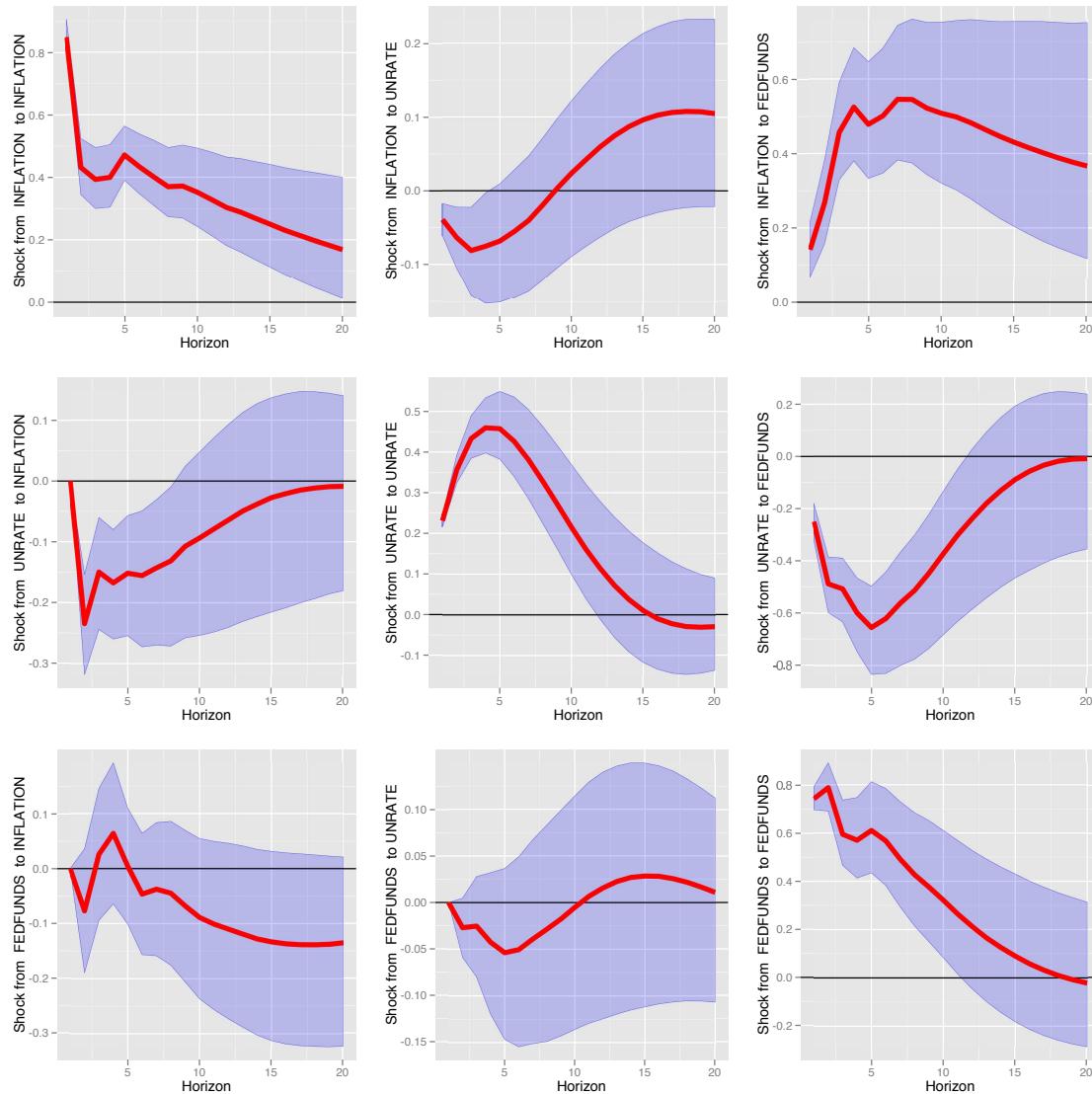


Figure 21: IRFs at 1997.

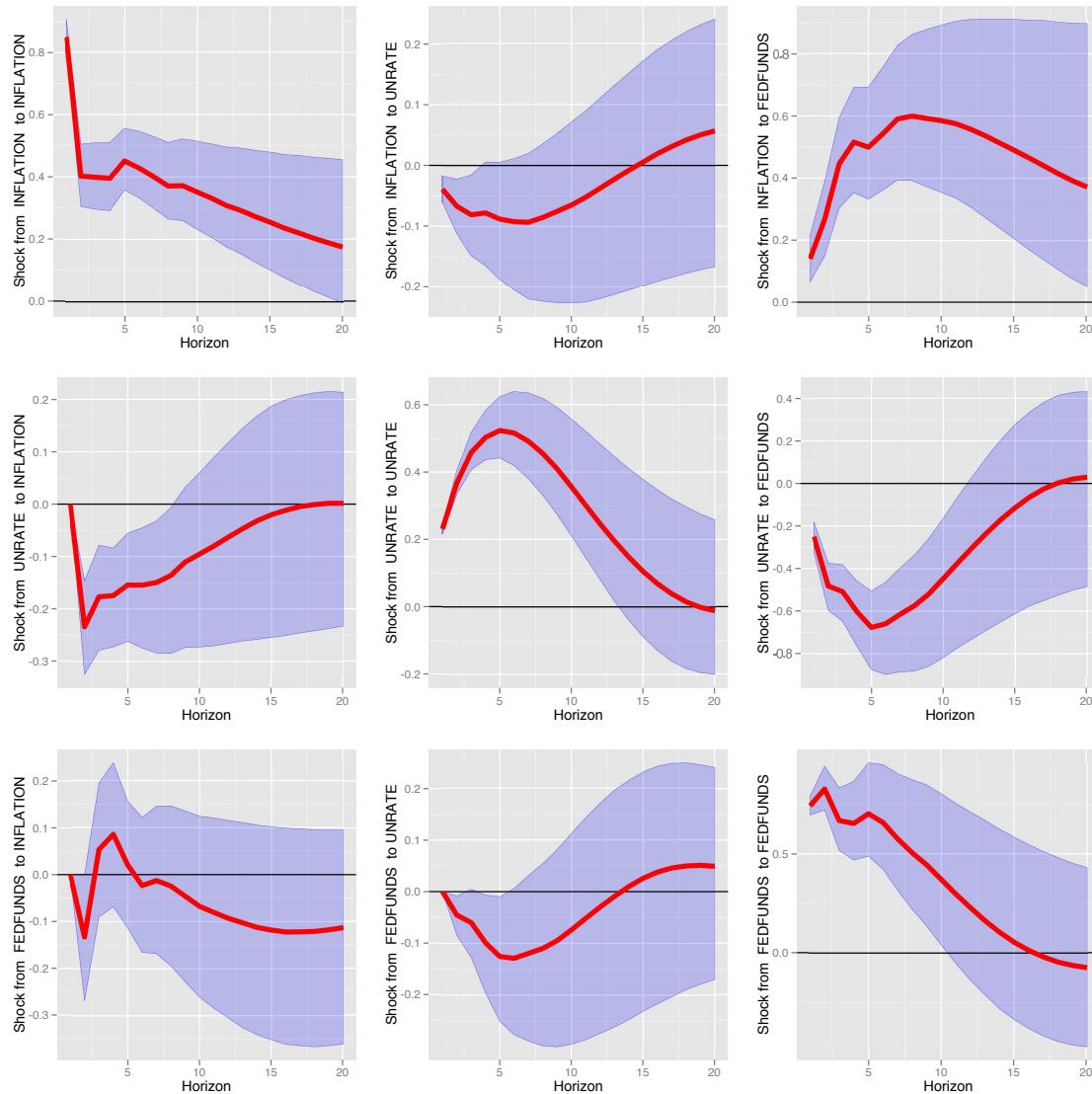


Figure 22: IRFs at 2006.

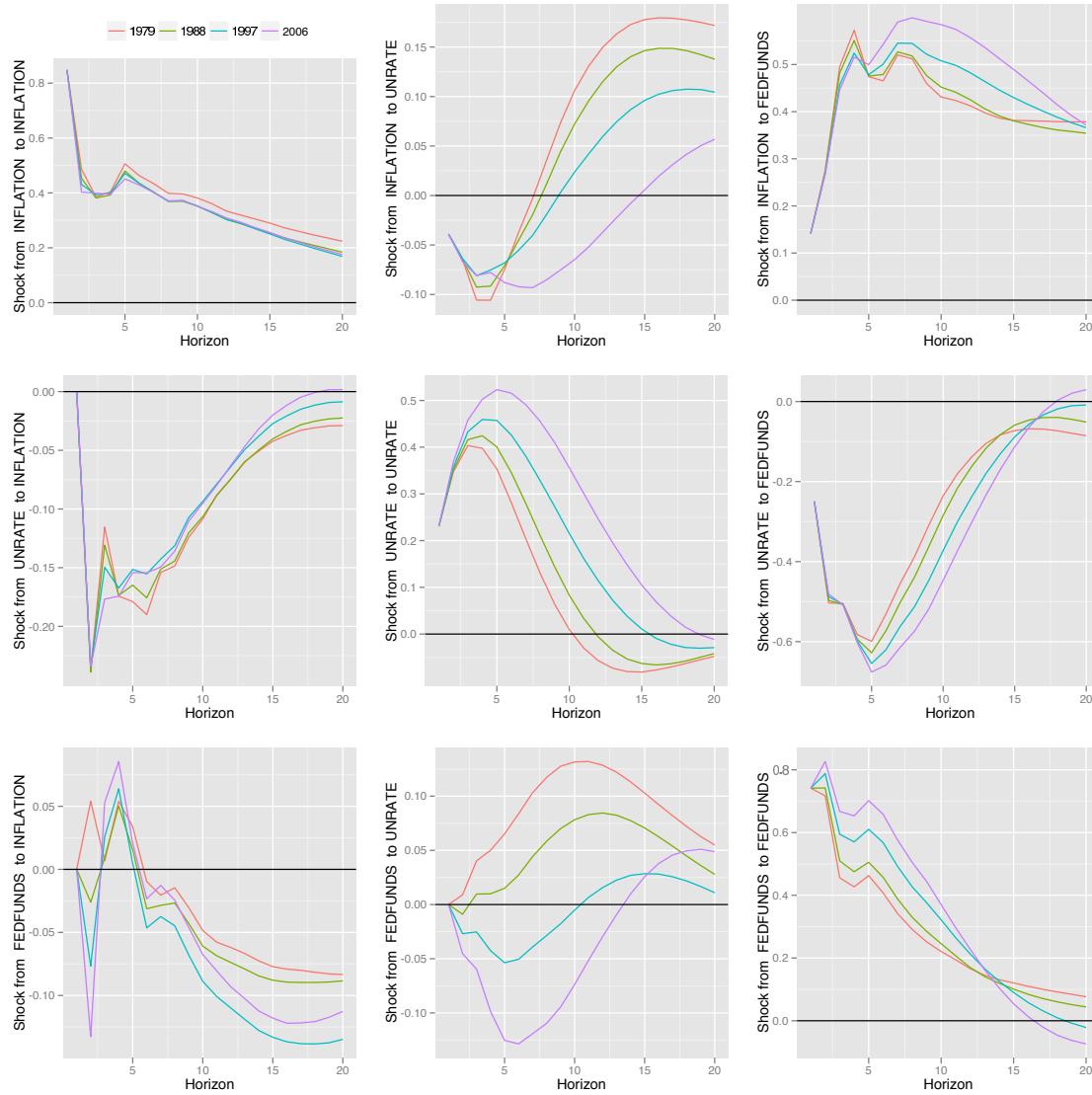


Figure 23: Comparison of Median IRFs.

8. DSGE

DSGE models are summarized by the first-order conditions of dynamic optimization problems faced by forward-looking agents who exhibit rational expectations (*i.e.*, agents know the overall structure of the economy, use all publicly available information efficiently, and do not make systematic errors when forming their expectations about the future). BMR contains numerous methods to solve, simulate, and estimate such models.

The DSGE-related literature has grown enormously since Bayesian methods were first applied to estimate DSGE model parameters, a topic neatly summarized by An and Schorfheide (2007). An early example of this, Smets and Wouters (2002) is a medium-scale estimated DSGE model, built for forecasting and counter-factual policy analysis. In subsequent work, Smets and Wouters (2007), the authors provide an estimated DSGE model of the U.S. economy, a model noted for its strong forecasting performance when benchmarked against a Bayesian VAR, which is often considered to possess the best all-round forecasting properties in the class of linear multivariate models.

The forecasting performance of DSGE models is a topic of many other papers. Christoffel et al. (2010), using a model described in Christoffel et al. (2008), provide a broad forecasting comparison to a DSGE model, using simple univariate models, classical VARs, and small and large BVARs. In addition to these, Edge et al. (2010) compare the forecasts of a DSGE model against the Federal Reserve Board's 'Greenbook' projections.

Bayesian methods are not the only possible approach to estimating DSGE models, however. Christiano et al. (2005), using the archetype New-Keynesian model, chose a minimum distance estimator to match the impulse response functions (IRFs) of a DSGE model to the IRFs produced by a monetary policy VAR, an approach similar in spirit to Rotemberg and Woodford (1997). Another alternative is given by Ireland (2004), who estimates a simple New-Keynesian model using pure maximum likelihood.

A relatively recent innovation is that of the DSGE-VAR by Del-Negro and Schorfheide (2004). The idea is to use the implied moments of a DSGE model as the prior to a BVAR, a neat example of which is found in Lubik and Schorfheide (2007). Using a simpler version of a model presented in Galí and Monacelli (2005), Lees et al. (2011) use the DSGE-VAR model to forecast five key macroeconomic variables of the New Zealand economy, with the DSGE-VAR (and the DSGE model itself) outperforming Bayesian and classical VARs forecasts.

However, Kolasa et al. (2009) find that the DSGE-VAR generally underperforms relative to a DSGE model, but still performs well against a BVAR.

In reduced-form, DSGE models are essentially vector autoregression models of order one, but with very tight cross-equation restrictions. BMR solves DSGE models using either Uhlig (1999)'s method of undetermined coefficients or Chris Sims' gensys algorithm. Other methods include the complex generalised Schur decomposition of Klein (2000). Anderson (2008) surveys six different solution methods for a first-order approximation, providing a computational comparison of each; Aruoba et al. (2006) complements this by covering both perturbation- and projection-based solution methods.

DSGE models, being inherently non-linear objects, often require some simplifications before solving them with a computer. One popular approach is to log-linearize the equilibrium conditions of a model around stationary steady-state values using a first-order Taylor approximation. Consider a general model made up of four types of variables: x^+ , a lead variable; x , a present-state variable; x^- a lag variable; and ε , an exogenous white noise process. We can write the non-linear model in expectational form

$$\mathbb{E}f(\mathbf{x}^+, \mathbf{x}, \mathbf{x}^-, \varepsilon) = 0 \quad (43)$$

and the rational expectations solution is given by ‘policy functions’ of the form

$$\mathbf{x} = \ell(\mathbf{x}^-, \varepsilon). \quad (44)$$

Plugging (44) into (43) yields the functional equation

$$\mathbb{E}f\left(\ell(\ell(\mathbf{x}^-, \varepsilon), \varepsilon^+), \ell(\mathbf{x}^-, \varepsilon), \mathbf{x}^-, \varepsilon\right) = 0$$

Let \bar{x} denote a steady-state value; a first-order approximation of the policy functions gives:

$$\hat{\ell}(\mathbf{x}^-, \varepsilon) = \bar{x} + \frac{\partial \ell}{\partial \mathbf{x}^-} \cdot (\mathbf{x}^- - \bar{x}) + \frac{\partial \ell}{\partial \varepsilon} \cdot (\varepsilon) \quad (45)$$

Numerically, our goal is to find the matrices that form the partial derivatives in (45).

8.1. Solution Methods

This section describes the two solution methods available in BMR. First, we describe Uhlig's method of undetermined coefficients in detail, then briefly mention Chris Sims' gensys solver.

8.1.1. Uhlig's Method

Uhlig (1999)'s method of undetermined coefficients can be used in two different ways. The first is referred to as the 'brute-force' method, which doesn't distinguish between control and so-called 'jump' variables, while the second method, known as the 'sensitive' approach, separates expectational and predetermined equations (those without any expectation operators). With some models, we may not have any natural candidates for control variables, particularly with the simple New-Keynesian model, which we will see later in an example.

Let $\zeta_{1,t}$ denote an $m \times 1$ vector of control variables, $\zeta_{2,t}$ denote an $n \times 1$ vector of jump variables, and z_t denote a $k \times 1$ vector of exogenous processes. With the sensitive approach, we begin with the system

$$0 = A\zeta_{1,t} + B\zeta_{1,t-1} + C\zeta_{2,t} + Dz_t \quad (46)$$

$$0 = \mathbb{E}_t \{ F\zeta_{1,t+1} + G\zeta_{1,t} + H\zeta_{1,t-1} + J\zeta_{2,t+1} + K\zeta_{2,t} + Lz_{t+1} + Mz_t \} \quad (47)$$

$$z_t = Nz_{t-1} + \varepsilon_t \quad (48)$$

and Uhlig's solution method produces the policy functions

$$\zeta_{1,t} = P\zeta_{1,t-1} + Qz_t \quad (49)$$

$$\zeta_{2,t} = R\zeta_{1,t-1} + Sz_t \quad (50)$$

$$z_t = Nz_{t-1} + \varepsilon_t$$

The matrix C is of size $l \times n$, where Uhlig allows for $l \geq n$, however BMR requires that $l = n$, which avoids the need to calculate the null matrix of C from a singular value decomposition. To achieve this, simply set the number of control variables ζ_1 equal to the number of expectational equations (assuming, of course, that one has the same number of equations as 'unknowns'). Even if pseudo control variables are included, they should have zero elements in the eventual solution matrix (P).

The brute-force approach doesn't distinguish between $\zeta_{1,t}$ and $\zeta_{2,t}$, instead let $\zeta_t = [\zeta_{1,t} \ \zeta_{2,t}]^\top$.

Uhlig's method then takes the system

$$0 = \mathbb{E}_t \{ F\zeta_{t+1} + G\zeta_t + H\zeta_{t-1} + Lz_{t+1} + Mz_t \} \quad (51)$$

$$z_t = Nz_{t-1} + \varepsilon_t \quad (52)$$

and produces the solution

$$\zeta_t = P\zeta_{t-1} + Qz_t \quad (53)$$

$$z_t = Nz_{t-1} + \varepsilon_t \quad (54)$$

The solution method works as follows. Using our assumed solution for the sensitive approach, replace $\zeta_{1,t}$ with $P\zeta_{1,t-1} + Qz_t$ in the deterministic block to give

$$\begin{aligned} 0 &= A(P\zeta_{1,t-1} + Qz_t) + B\zeta_{1,t-1} + C(R\zeta_{1,t-1} + Sz_t) + Dz_t \\ &= (AP + CR + B)\zeta_{1,t-1} + (AQ + CS + D)z_t \end{aligned}$$

and, in the expectational block,

$$\begin{aligned} 0 &= ((FQ + JS + L)N + (FP + JR + G)Q + KS + M)z_t \\ &\quad + ((FP + JR + G)P + KR + H)\zeta_{1,t-1} \end{aligned}$$

For these conditions to hold, we must have

$$0 = AP + CR + B$$

$$0 = (FP + JR + G)P + KR + H$$

From the first condition, $R = -C^{-1}(AP + B)$, which we can use on the last equation

$$\begin{aligned} 0 &= (FP + J(-C^{-1}(AP + B)) + G)P + K(-C^{-1}(AP + B)) + H \\ &= FPP - JC^{-1}APP - JC^{-1}BP + GP - KC^{-1}AP - KC^{-1}B + H \\ &= (F - JC^{-1}A)PP - (JC^{-1}B - G + KC^{-1}A)P - KC^{-1}B + H \end{aligned}$$

The values of P must satisfy this equation.

Define three matrices:

$$\begin{aligned}\Psi &:= \begin{bmatrix} F - JC^{-1}A \\ JC^{-1}B - G + KC^{-1}A \end{bmatrix} \\ \Gamma &:= \begin{bmatrix} KC^{-1}B - H \end{bmatrix} \\ \Theta &:= \begin{bmatrix} \Gamma & \Theta \\ \mathbb{I}_m & \mathbf{0}_{m \times m} \end{bmatrix}\end{aligned}$$

(For the brute-force method, set $\Psi = F$, $\Gamma = -G$, and $\Theta = -H$.) Written differently, P should satisfy the matrix ‘quadratic’ equation

$$\Psi P^2 - \Gamma P - \Theta = 0$$

Stack (Ψ, Γ, Θ) as follows

$$\begin{aligned}\Xi &= \begin{bmatrix} \Gamma & \Theta \\ \mathbb{I}_m & \mathbf{0}_{m \times m} \end{bmatrix} \\ \Delta &= \begin{bmatrix} \Psi & \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} & \mathbb{I}_m \end{bmatrix}\end{aligned}$$

With Ξ and Δ , we solve the generalised eigenvalue problem,

$$\Xi \mathbf{v} = \Delta \mathbf{v} \lambda, \quad (55)$$

where \mathbf{v} are the eigenvectors and λ is a diagonal matrix with the eigenvalues on the main diagonal, sorted in ascending order. Fully written, the problem is

$$\begin{bmatrix} \Gamma & \Theta \\ \mathbb{I}_m & \mathbf{0}_{m \times m} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{11} & \mathbf{v}_{12} \\ \mathbf{v}_{21} & \mathbf{v}_{22} \end{bmatrix} = \begin{bmatrix} \Psi & \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} & \mathbb{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{v}_{11} & \mathbf{v}_{12} \\ \mathbf{v}_{21} & \mathbf{v}_{22} \end{bmatrix} \begin{bmatrix} \lambda_{11} & \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} & \lambda_{22} \end{bmatrix} \quad (56)$$

R doesn’t natively provide the same user-friendly approach to solving a generalised eigenvalue problem as that found in other software (such as Matlab). Of course, if Δ^{-1} exists,

then we could solve the standard eigenvalue problem of $A\mathbf{v} = \mathbf{v}\lambda$, where $A = \Delta^{-1}\Xi$, using the ‘eigen’ function. Most linear algebra libraries will solve the problem using decompositions of Δ and Ξ first; for example, if Δ is a Hermitian, positive-definite matrix, one could use the Cholesky decomposition of $\Delta = LL^\top$, where L is lower-triangular, $L^{-1}\Xi(L^\top)^{-1}L^\top\mathbf{v} = L^\top\mathbf{v}\lambda$, computationally simpler than inverting Δ directly.

However, typically Δ^{-1} will not exist—and this is almost always true with large models. Another approach would be to use a singular value decomposition of $\Delta = U\Sigma V'$, where U and V are unitary matrices, $UU' = VV' = \mathbb{I}^*$ (the prime ‘ denotes the conjugate transpose), and, for some tolerance $\epsilon > 0$, set $\Sigma_{i,i}^{-1} = 0$ if $\Sigma_{i,i} < \epsilon$, and denoting the adjusted matrix as $\tilde{\Sigma}$, we have $\Delta^{-1} \approx V\tilde{\Sigma}^{-1}U^\top$, otherwise known as a pseudo-inverse of Δ .

Instead, the generalised eigenvalue problem can be solved by means of a generalised Schur decomposition, commonly referred to as the ‘QZ’ decomposition, a method used by Klein (2000) and in the current version of Dynare. For $A, B \in \mathbb{C}^{d \times e}$, $Ax = Bx\lambda$, we perform a QZ decomposition of the matrix pencil (A, B)

$$\begin{cases} \mathcal{S} = Q'AZ \\ \mathcal{T} = Q'BZ \end{cases}$$

where $QQ' = ZZ' = \mathbb{I}^*$ are unitary. The generalised eigenvalues are then given by $\lambda_i = \mathcal{S}_{ii}/\mathcal{T}_{ii}$. The QZ algorithm in the Armadillo linear algebra library is used to find the generalized eigenvalues and eigenvectors that are needed for Uhlig’s method.

Going back to our problem, we can multiply out (56) to get

$$\begin{bmatrix} \Gamma\mathbf{v}_{11} + \Theta\mathbf{v}_{21} & \Gamma\mathbf{v}_{12} + \Theta\mathbf{v}_{22} \\ \mathbf{v}_{11} & \mathbf{v}_{12} \end{bmatrix} = \begin{bmatrix} \Psi\mathbf{v}_{11}\lambda_{11} & \Psi\mathbf{v}_{12}\lambda_{22} \\ \mathbf{v}_{21}\lambda_{11} & \mathbf{v}_{22}\lambda_{22} \end{bmatrix} \quad (57)$$

Focus on the stable eigenvalues, λ_{11} , and we have a system of two equations

$$\begin{aligned} \Gamma\mathbf{v}_{11} + \Theta\mathbf{v}_{21} &= \Psi\mathbf{v}_{11}\lambda_{11} \\ \mathbf{v}_{11} &= \mathbf{v}_{21}\lambda_{11} \end{aligned}$$

Substitute for \mathbf{v}_{11} into the first to give

$$\Gamma \mathbf{v}_{21} \lambda_{11} + \Theta \mathbf{v}_{21} = \Psi \mathbf{v}_{21} \lambda_{11} \lambda_{11}$$

Post multiply everything by \mathbf{v}_{21}^{-1}

$$\Gamma \mathbf{v}_{21} \lambda_{11} \mathbf{v}_{21}^{-1} + \Theta = \Psi \mathbf{v}_{21} \lambda_{11} \lambda_{11} \mathbf{v}_{21}^{-1}$$

Recall our original problem was

$$\Psi P^2 - \Gamma P - \Theta = 0$$

which looks a lot like

$$\Psi \mathbf{v}_{21} \lambda_{11} \lambda_{11} \mathbf{v}_{21}^{-1} - \Gamma \mathbf{v}_{21} \lambda_{11} \mathbf{v}_{21}^{-1} - \Theta = 0$$

by noticing that $(\mathbf{v}_{21} \lambda_{11} \mathbf{v}_{21}^{-1})(\mathbf{v}_{21} \lambda_{11} \mathbf{v}_{21}^{-1}) = \mathbf{v}_{21} \lambda_{11} \lambda_{11} \mathbf{v}_{21}^{-1}$. Then,

$$P = \mathbf{v}_{21} \lambda_{11} \mathbf{v}_{21}^{-1} \quad (58)$$

solves the matrix quadratic equation.

With the P matrix, we can easily solve for Q and S with

$$\begin{bmatrix} \text{vec}(Q) \\ \text{vec}(S) \end{bmatrix} = - \begin{bmatrix} \mathbb{I}_k \otimes A & \mathbb{I}_k \otimes C \\ N^\top \otimes F + \mathbb{I}_k \otimes (FP + JR + G) & N^\top \otimes J + \mathbb{I}_k \otimes K \end{bmatrix}^{-1} \begin{bmatrix} \text{vec}(D) \\ \text{vec}(LN + M) \end{bmatrix} \quad (59)$$

and, finally,

$$R = -C^{-1}(AP + B) \quad (60)$$

and we have our 4 required matrices. The brute-force method would mean that both R and S are empty, and $\text{vec}(Q) = -[N^\top \otimes F + \mathbb{I}_k \otimes (FP + G)]^{-1} \text{vec}(LN + M)$.

Regardless of whether we use the brute-force approach or the ‘sensitive’ approach, we need to stack the matrices containing the coefficients—which are non-linear functions of the ‘deep’ parameter values of the DSGE model—to form a simpler representation for estimation and simulation. With the solution, we can build a state-space structure,

$$\xi_t = \mathcal{F} \xi_{t-1} + \mathcal{G} \varepsilon_t \quad (61)$$

where $\xi_t = [\zeta_t \ z_t]^\top$, as follows:

$$\mathcal{F} := \begin{bmatrix} P & \mathbf{0}_{m \times n} & QN \\ R & \mathbf{0}_{n \times n} & SN \\ \mathbf{0}_{k \times m} & \mathbf{0}_{k \times n} & N \end{bmatrix}_{(m+n+k) \times (m+n+k)}$$

$$\mathcal{G} := \begin{bmatrix} Q \\ S \\ \mathbb{I}_k \end{bmatrix}_{(m+n+k) \times k}$$

With this structure, we can easily produce impulse response functions (IRFs) and other descriptive features of the model dynamics.

8.1.2. gensys

Chris Sims' ‘gensys’ solver is based on the system

$$\Gamma_0(\theta)\xi_t = C(\theta) + \Gamma_1(\theta)\xi_{t-1} + \Psi(\theta)\epsilon_t + \Pi(\theta)\eta_t \quad (62)$$

and produces a solution

$$\xi_t = \tilde{C}(\theta) + F(\theta)\xi_{t-1} + G(\theta)\epsilon_t \quad (63)$$

As this solution method is rather well-known, we omit the details here.

8.2. Estimation

To estimate the parameter values of a DSGE model, we need a method to evaluate the likelihood function $p(\mathcal{Y}^T | \theta, M_i)$. The problem we face is that, while we can observe some of the variables in the model (such as output), we cannot directly observe others (such as technology), or we assume that these series are unobservable due to the dubious nature of their empirical proxies.

A solution to this problem is to use a state space approach and build the likelihood via a filter; see Arulampalam et al. (2002) for a general discussion of linear and non-linear filtering problems, and Fernández-Villaverde (2010) for a discussion with a particular focus on DSGE models. As in the BVARTVP section, for notational convenience, a superscript (\mathcal{Y}^t) denotes all information up to and including time t , whereas a subscript implies a specific observation at time t .

The state space setup involves two equations which connect the observable and unobservable series of our model. The first, which we call the measurement equation, maps the state ξ_t plus some random noise $\varepsilon_{\mathcal{Y}}$ to our observable series \mathcal{Y}_t ,

$$\mathcal{Y}_t = f(\xi_t, \varepsilon_{\mathcal{Y},t} | \theta) \quad (64)$$

and this relationship defines a conditional density $p(\mathcal{Y}_t | \xi_t, \theta)$. The second equation, which we call the state transition equation, maps previous-period values of the state plus some random noise, ε_{ξ} , to the present state,

$$\xi_t = g(\xi_{t-1}, \varepsilon_{\xi,t} | \theta) \quad (65)$$

and (65) implies a conditional density $p(\xi_t | \xi_{t-1}, \theta)$. Note that these densities are conditional on some values for the ‘deep’ parameters of the DSGE model (θ), as the coefficient matrices in ξ are functions of θ .

The likelihood of a DSGE model, $p(\mathcal{Y}^T | \theta) = p(\mathcal{Y}_1 | \theta) \prod_{t=2}^T p(\mathcal{Y}_t | \mathcal{Y}^{t-1}, \theta)$, can be written as

$$\begin{aligned} p(\mathcal{Y}^T | \theta) &= p(\mathcal{Y}_1 | \theta) \prod_{t=2}^T \int_{\mathbb{R}^{m+n+k}} p(\mathcal{Y}_t | \xi_t, \mathcal{Y}^{t-1}, \theta) p(\xi_t | \mathcal{Y}^{t-1}, \theta) d\xi_t \\ &= p(\mathcal{Y}_1 | \theta) \prod_{t=2}^T \int_{\mathbb{R}^{m+n+k}} p(\mathcal{Y}_t | \xi_t, \theta) p(\xi_t | \mathcal{Y}^{t-1}, \theta) d\xi_t \end{aligned} \quad (66)$$

where $p(\mathcal{Y}_1|\theta) = \int p(\mathcal{Y}_1|\xi_1, \theta)p(\xi_1)d\xi_1$, and $p(\xi_1)$ is the initial distribution of the state. With $p(\xi_1)$ we build a sequence of densities $p(\xi_t|\mathcal{Y}^{t-1}, \theta) \forall t \in [1, T]$ in two steps: prediction and correction (or ‘updating’).

First, given $t - 1$ information, we predict the position (distribution) of the state at time t using

$$\begin{aligned} p(\xi_t|\mathcal{Y}^{t-1}, \theta) &= \int_{\mathbb{R}^{m+n+k}} p(\xi_t|\xi_{t-1}, \mathcal{Y}^{t-1}, \theta)p(\xi_{t-1}|\mathcal{Y}^{t-1}, \theta)d\xi_{t-1} \\ &= \int_{\mathbb{R}^{m+n+k}} p(\xi_t|\xi_{t-1}, \theta)p(\xi_{t-1}|\mathcal{Y}^{t-1}, \theta)d\xi_{t-1} \end{aligned}$$

and this prediction introduces a new conditional density, $p(\xi_t|\mathcal{Y}^t, \theta)$, to our problem, *i.e.*, an update of the position of the state with new t information. We update our $t - 1$ estimate of the state given new t information using Bayes’ rule,

$$p(\xi_t|\mathcal{Y}^t, \theta) = \frac{p(\mathcal{Y}_t|\xi_t, \theta)p(\xi_t|\mathcal{Y}^{t-1}, \theta)}{\int p(\mathcal{Y}_t|\xi_t, \theta)p(\xi_t|\mathcal{Y}^{t-1}, \theta)d\xi_t}$$

Thus, we have four predictive densities in the filtering problem, knowledge of which would allow us to integrate out the state in the likelihood equation. The first two densities relate to our observable series, $p(\mathcal{Y}_t|\xi_t, \theta)$ and $p(\mathcal{Y}_t|\mathcal{Y}^{t-1}, \theta)$, the former being defined by the measurement equation. The other two densities relate to the state, $p(\xi_t|\mathcal{Y}^{t-1}, \theta)$ and $p(\xi_t|\mathcal{Y}^t, \theta)$.

8.2.1. The Kalman Filter

If the system, given by (64) and (65), is linear, and the innovations ($\varepsilon_{\mathcal{Y},t}$ and $\varepsilon_{\xi,t}$) Gaussian, then the likelihood of our DSGE model, $p(\mathcal{Y}^T|\theta, M_i)$, can be evaluated through a Kalman filter, which assumes that the predictive densities $p(\xi_t|\mathcal{Y}^{t-1}, \theta)$, $p(\xi_t|\mathcal{Y}^t, \theta)$, and $p(\mathcal{Y}_t|\mathcal{Y}^{t-1}, \theta)$ are all conditionally Gaussian in form. Thus, all we will need to track are the first and second moments of the problem, as these form sufficient statistics for the Gaussian distribution.

The measurement equation of $\mathcal{Y}_{(T \times j)}$ is now

$$\mathcal{Y}_t = \mathcal{C} + \mathcal{H}^\top \xi_t + \varepsilon_{\mathcal{Y},t}$$

where the matrix $\mathcal{H}_{(n+m+k) \times j}$ relates the state to our observable series, \mathcal{C} is a $1 \times j$ vector of

intercept terms, and $\varepsilon_{\mathcal{Y},t} \sim \mathcal{N}(0, \mathcal{R})$. The state transition equation is

$$\xi_t = \mathcal{F}\xi_{t-1} + \mathcal{G}\varepsilon_{\xi,t}$$

where \mathcal{F} is the state transition matrix. Note that the state transition equation is simply the solution to our DSGE model (61), where we assume that $\varepsilon_{\xi,t} \sim \mathcal{N}(0, \mathcal{Q})$.

The Kalman filter sequentially builds a likelihood function from forecasts of \mathcal{Y}_{t+1} given t information, and error variances of this forecast. This is done in three short steps, repeated $\forall t \in [1, T]$. First, predict the state at time $t + 1$ given information available at time t ; second, update the estimated position of the state with new \mathcal{Y}_{t+1} information; and, finally, calculate the likelihood at $t + 1$ based on forecast errors of \mathcal{Y}_{t+1} and the covariance matrix of these forecasts.

To begin, initialise the state ξ at time $t = 0$, ξ_0 , and the covariance matrix of the state variables, P_0 . We then predict the state at time $t + 1$ given information at time t with

$$\xi_{t+1|t} := \mathbb{E}_t[\xi_{t+1}] = \mathcal{F}\xi_{t|t}, \quad (67)$$

and calculate the covariance matrix of the state

$$P_{t+1|t} = \mathcal{F}P_{t|t}\mathcal{F}^\top + \mathcal{G}\mathcal{Q}\mathcal{G}^\top. \quad (68)$$

With these predictions, we then bring in time $t + 1$ information and update our forecasts of the state accordingly. We first get the prediction residual,

$$\epsilon_{t+1} = \mathcal{Y}_{t+1} - \mathcal{C} - \mathcal{H}^\top\xi_{t+1|t}, \quad (69)$$

and the covariance matrix of the residual,

$$\Sigma_{t+1} = \mathcal{H}^\top P_{t+1|t} \mathcal{H} + \mathcal{R}. \quad (70)$$

Then we calculate the Kalman gain,

$$K_{t+1} = P_{t+1|t} \mathcal{H} \Sigma_{t+1}^{-1} \quad (71)$$

which we use to update our forecast of the state,

$$\xi_{t+1|t+1} = \xi_{t+1|t} + K_{t+1}\epsilon_{t+1} \quad (72)$$

and the covariance matrix

$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1}\mathcal{H}^\top P_{t+1|t}. \quad (73)$$

This completes the updating of the state with $t + 1$ information. As we've assumed normality of $\varepsilon_{\mathcal{Y},t}$ and $\varepsilon_{\xi,t}$, the conditional likelihood function is proportional to that of a normal density

$$\ln(\mathcal{L}_{t+1}) \propto -\frac{1}{2} (\ln(|\Sigma_{t+1}|) + \epsilon_{t+1}^\top \Sigma_{t+1}^{-1} \epsilon_{t+1}) \quad (74)$$

We would then go back to equations (67) and (68), move the time indices forward by one, and, using our $\xi_{t+1|t+1}$ and $P_{t+1|t+1}$ from equations (72) and (73), we estimate the state and covariance of the state for $t + 2$. The process continues, using equations (67) through (73), until time T , and the log likelihood function is the sum

$$\ln(\mathcal{L}^T) \propto -\frac{1}{2} \sum_{t=1}^T \{\ln(|\Sigma_t|) + \epsilon_t^\top \Sigma_t^{-1} \epsilon_t\}$$

given initial values of the state and the covariance matrix of the state.

The initial values of the state (ξ_0) are assumed to be zero, and the initial value of the covariance matrix of the state is set to the steady-state covariance matrix of ξ , which we denote by Ω_{ss} ,

$$\Omega_{ss} = \mathcal{F}\Omega_{ss}\mathcal{F}^\top + \mathcal{G}\mathcal{Q}\mathcal{G}^\top$$

the existence of which follows from the stationarity of the model. Ω is found via a doubling algorithm.

8.2.2. Chandrasekhar Recursions

When estimating DSGE models with a lot of state variables, the term $\mathcal{F}P_{t|t}\mathcal{F}^\top$ in (68) involves large-scale matrix multiplication. Recently, Herbst (2014) showed that, when the number of states is much larger than the number of observables, there are substantial computation gains from recasting the filtering problem in terms of the first difference of the state covariance

matrix $P_{t+1|t}$,

$$\Delta P_{t+1|t} := P_{t+1|t} - P_{t|t-1}$$

Using the decomposition $\Delta P_{t+1|t} = W_t M_t W_t^\top$, where W_t is $j \times \text{rank}(\Delta P_{t+1|t})$, we can reduce the dimension of matrix multiplication by noting that $\text{rank}(\Delta P_{t+1|t}) \leq \min\{n + m + k, j\}$. See Herbst (2014) for details.

Using the notation from the previous section, the recursions for M_t and W_t are

$$\begin{aligned} W_t &= (\mathcal{F} - K_t \Sigma_t^{-1} \mathcal{H}^\top) W_{t-1} \\ M_t &= M_{t-1} + M_{t-1} W_{t-1}^\top \mathcal{H} \Sigma_{t-1}^{-1} \mathcal{H}^\top W_{t-1} M_{t-1} \end{aligned}$$

The forecast residual covariance matrix and Kalman gain recursions then become:

$$\begin{aligned} \Sigma_t &= \Sigma_{t-1} + \mathcal{H}^\top W_{t-1} M_{t-1} W_{t-1}^\top \mathcal{H} \\ K_t &= K_{t-1} + \mathcal{F} W_{t-1} M_{t-1} W_{t-1}^\top \mathcal{H} \end{aligned}$$

8.3. Prior Distributions

This section describes the five prior distributions allowed in BMR; for further details on these distributions, the reader is directed to Papoulis and Pillai (2002). The choice of prior distribution will depend on the assumed support of each parameter; for \mathbb{R} , a Gaussian distribution seems appropriate; for \mathbb{R}_+ , Gamma or inverse-Gamma distributions; or with support $(a, b) \subset \mathbb{R}$, we may select a Beta or Uniform distribution.

The Normal (or Gaussian) distribution is described by two parameters, its mean and variance, denoted μ and σ^2 , respectively, with a probability density function (pdf) given by

$$p(\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\theta - \mu)^2\right) \quad (75)$$

which is symmetric around μ , and $\sqrt{2\pi\sigma^2}$ serves as the normalizing constant (*i.e.*, that the area under $p(\theta)$ integrates to one). If θ is normally distributed, we denote this by $\theta \sim \mathcal{N}(\mu, \sigma^2)$.

For $\theta \in \mathbb{R}_+$, the Gamma distribution is described by the parameters α and β , $\alpha, \beta > 0$,

commonly referred to as the shape and scaling parameters, respectively, and the pdf is

$$p(\theta) = \frac{\theta^{\alpha-1}}{\Gamma(\alpha)\beta^\alpha} \exp\left(-\frac{\theta}{\beta}\right) \quad (76)$$

where $\Gamma(\alpha)$ is the usual Gamma function, defined as

$$\Gamma(\alpha) = \int_0^\infty \theta^{\alpha-1} \exp(-\theta) d\theta$$

where, if $\alpha \in \mathbb{Z}_+$,

$$\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1) = (\alpha - 1)!$$

We denote the Gamma distribution by $\mathcal{G}(\alpha, \beta)$. There is a simple connection between the Gamma and inverse-Gamma distributions. If we define $\vartheta := 1/\theta$, then we can use the density transformation theorem to show that

$$\begin{aligned} p(\vartheta) &= p(1/\vartheta) \left| \frac{d\theta}{d\vartheta} \right| \\ &= \frac{\vartheta^{1-\alpha}}{\Gamma(\alpha)\beta^\alpha} \exp\left(-\frac{1}{\vartheta\beta}\right) \vartheta^{-2} \\ &= \frac{\vartheta^{-\alpha-1}}{\Gamma(\alpha)\beta^\alpha} \exp\left(-\frac{1}{\vartheta\beta}\right) \end{aligned} \quad (77)$$

Although, note that β in (77) is the reciprocal of β in the Gamma distribution; *i.e.*, if $\theta \sim \mathcal{G}(\alpha, \beta)$, then, for $\vartheta = 1/\theta$, $\vartheta \sim \mathcal{IG}(\alpha, 1/\beta)$.

If $\theta \in (a, b)$, the Beta pdf, with parameters α and β , $\alpha, \beta > 0$, is given by

$$p(\theta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1} \quad (78)$$

which is well-defined over the unit interval, where the beta function $B(\alpha, \beta)$ is defined as

$$B(\alpha, \beta) = \int_0^1 \theta^{\alpha-1} (1-\theta)^{\beta-1} d\theta.$$

The Uniform pdf with parameters a and b , $b - a > 0$, is given by

$$p(\theta) = \begin{cases} \frac{1}{b-a} & \text{if } \theta \in [a, b] \\ 0 & \text{otherwise.} \end{cases}$$

We denote the Uniform distribution by $\mathcal{U}(a, b)$.

To aid in the selection of prior distributions for the parameters of a DSGE model, the ‘prior’ function will plot any of the above pdfs and return the mean, mode, and variance (whenever they’re well-defined for a given parameterization). A full list of the function inputs are given in section 8.8. For example,

```
grid.newpage()
pushViewport(viewport(layout=grid.layout(3,2)))
prior("Normal",c(1,0.1),NR=1,NC=1)
prior("Normal",c(0.2,0.01),NR=1,NC=2)
prior("Gamma",c(2,2),NR=2,NC=1)
prior("Gamma",c(2,1/2),NR=2,NC=2)
prior("IGamma",c(3,2),NR=3,NC=1)
prior("IGamma",c(3,1),NR=3,NC=2)
```

is shown in figure 24, and

```
grid.newpage()
pushViewport(viewport(layout=grid.layout(2,2)))
prior("Beta",c(20,2),NR=1,NC=1)
prior("Beta",c(7,7),NR=1,NC=2)
prior("Uniform",c(0,1),NR=2,NC=1)
prior("Uniform",c(1,5),NR=2,NC=2)
```

is shown in figure 25.

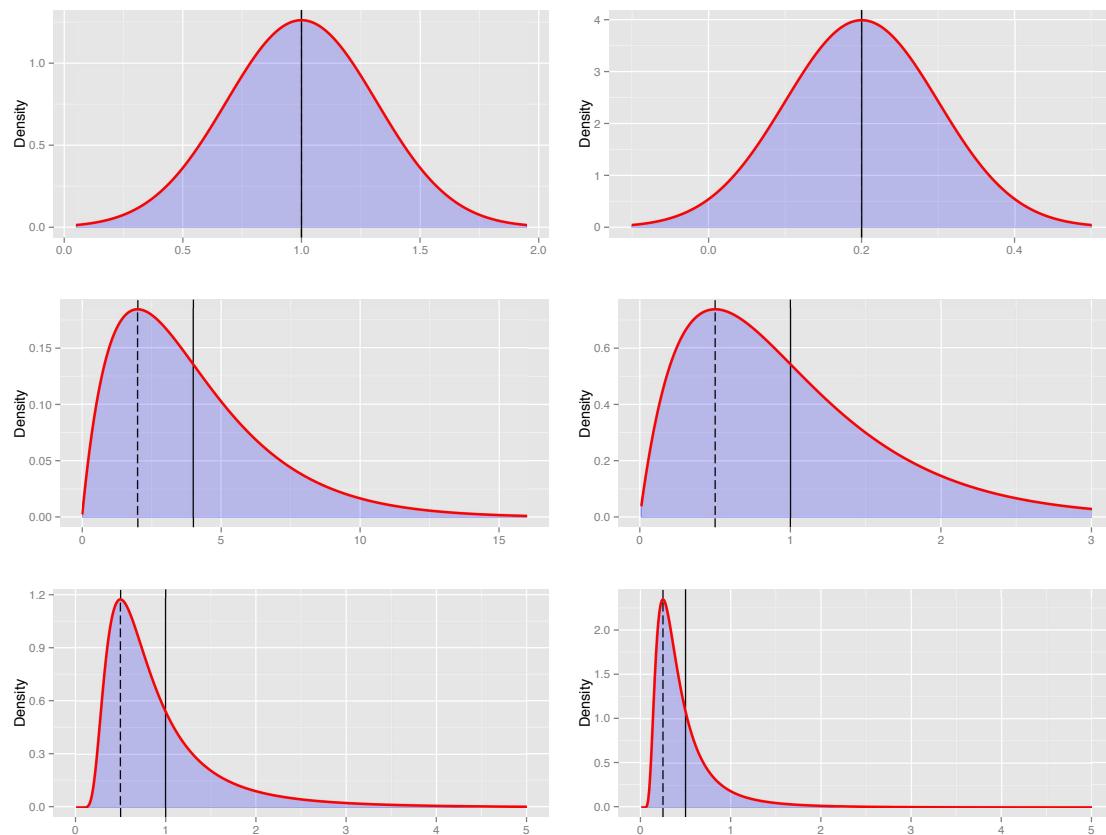


Figure 24: Prior Distributions: Normal, Gamma, and inverse-Gamma.

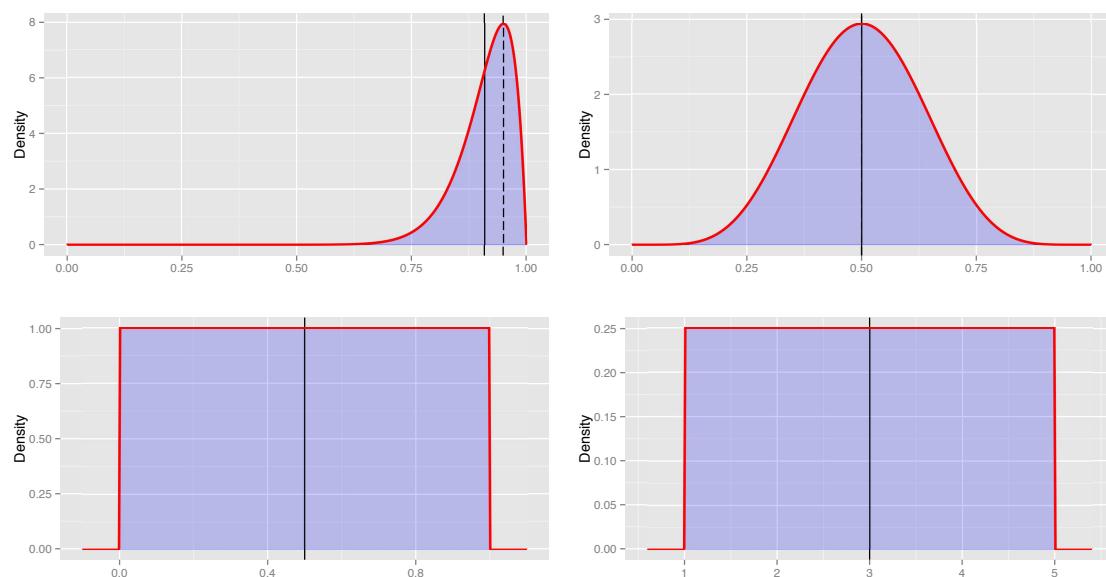


Figure 25: Prior Distributions: Beta and Uniform.

8.4. Posterior Sampling

In BMR, we sample from the posterior distribution of interest, $p(\theta|\mathcal{Y}, M_i)$, via a Random Walk Metropolis MCMC algorithm. The algorithm begins by finding somewhere ‘high’ on the posterior space, such as the mode of the posterior distribution, which can be found by maximizing the natural log of the posterior kernel,

$$\ln \mathcal{K}(\theta|\mathcal{Y}, M_i) = \ln \mathcal{L}(\theta|\mathcal{Y}, M_i) + \ln p(\theta|M_i) \quad (79)$$

where our likelihood is evaluated with a Kalman filter. One computational feature to note is that BMR, as in Warne (2012), will transform every deep parameter such that its support is unbounded, instead of working with θ directly. This occurs for any parameters which are given a Gamma, inverse-Gamma prior, or Beta prior. When this occurs, BMR will adjust the posterior density above to account for a Jacobian term, which follows from the density transformation theorem.

The sampling algorithm is as follows. First, draw θ^0 from a starting distribution $p^0(\theta)$; this is the posterior mode from an optimization routine. Second, draw a proposal $\theta^{(*)}$ from a jumping distribution,

$$\mathcal{N}(\theta^{(h-1)}, c \cdot \Sigma_m) \quad (80)$$

where Σ_m is the inverse of the Hessian computed at the posterior mode and c is a scaling factor which is typically set by the researcher to achieve an acceptance rate of between 20-40%. Third, compute the acceptance ratio,

$$\nu = \frac{\mathcal{K}(\theta^{(*)}|\mathcal{Y}, M_i)}{\mathcal{K}(\theta^{(h-1)}|\mathcal{Y}, M_i)} \quad (81)$$

If our acceptance rate is too high, the algorithm will not visit the tails of the distribution often enough; too low and we may not achieve convergence before sampling ends. Finally, we accept or reject the proposal according to

$$\theta^{(h)} = \begin{cases} \theta^{(*)} & \text{with probability } \min\{\nu, 1\} \\ \theta^{(h-1)} & \text{else} \end{cases} \quad (82)$$

This process is repeated for a given number of user-specified replications. The draws from the

RWM algorithm will be serially correlated and so, to avoid the initial values from dominating the shape of the posterior density before the chains converge, we discard an initial fraction of draws as sample burn-in.

8.5. Marginal Likelihood

To compare different models over the same data, the marginal likelihood (4),

$$p(\mathcal{Y}|M_i) = \int_{\Theta} p(\mathcal{Y}|\theta, M_i)p(\theta|M_i)d\theta$$

is calculated with a Laplacian approximation. The Laplace approximation of the integral is given by

$$p(\mathcal{Y}|M_i) \approx p(\mathcal{Y}|\theta, M_i)p(\theta|M_i)(2\pi)^{k/2}|\Sigma_m|^{1/2} \quad (83)$$

where k is the dimension of θ . The approximation (83) is evaluated at the posterior mode of θ , and is returned in log form.

8.6. Summary

To summarise the process of estimation, we define two main steps and the sub-steps contained therein.

1. Initialization and finding the posterior mode:
 - (i) The user provides:
 - i. j series of data in $T \times j$ format, ensuring that the number of observable series does not exceed to the number of stochastic shocks in the model, $j \leq k$, thus avoiding stochastic singularity;
 - ii. write a function that maps the model's parameters to the relevant matrices of a suitable solution method. The function should also return a $k \times k$ matrix called 'shocks' that comprises the covariance matrix of the shocks, a $k \times 1$ vector of constant terms in the measurement equation called 'ObsCons', and a $j \times j$ matrix called 'MeasErrs' that comprises the covariance matrix of the measurement errors.
 - iii. provide initial values of the parameters to start the posterior mode optimization routine;

- iv. select prior distributions for the estimated parameters, as well as the relevant parameters of these distributions (mean, standard deviation, degrees of freedom, *etc.*), and any upper and/or lower bounds;
- v. set the $(m + n + k) \times j$ matrix that maps the observable series to the reduced-form DSGE model, *i.e.*, the \mathcal{H} matrix of the measurement equation,

$$\mathcal{Y}_t = \mathcal{C} + \mathcal{H}^\top \xi_t + \varepsilon_{\mathcal{Y},t}$$

- vi. select an optimization algorithm to use when computing the posterior mode, along with any upper- and lower-bounds for optimization;
 - vii. set the scaling parameter (c) of the inverse of the Hessian at the posterior mode to adjust the acceptance rate;
 - viii. and, finally, set the number of replications to keep and any sample burn-in.
- (ii) BMR will check if the model can be solved using the initial values provided. The initial values will be transformed such that support of each parameter is in \mathbb{R} .
- (iii) The problem then passes to a numerical optimizer to locate the posterior mode.
- i. The parameters are transformed back to their standard values, then, using the function the user supplied, will be used to solve the DSGE model.
 - ii. With the solved model, we pass the state transition matrix to the Kalman filter

$$\xi_t = \mathcal{F} \xi_{t-1} + \varepsilon_{\xi,t}$$

along with the user-supplied data and \mathcal{H} matrix

$$\mathcal{Y}_t = \mathcal{C} + \mathcal{H}^\top \xi_t + \varepsilon_{\mathcal{Y},t}$$

The Kalman filter will then return the log-likelihood value $\ln(\mathcal{L}(\theta|\mathcal{Y}, M_i))$.

- iii. Finally, the log-value of the posterior kernel,

$$\ln \mathcal{K}(\theta|\mathcal{Y}, M_i) = \ln \mathcal{L}(\theta|\mathcal{Y}, M_i) + \ln p(\theta|M_i),$$

is found by checking the values against the prior distributions $p(\theta|M_i)$ the user supplied, and adjusting the posterior for the Jacobian, should that be

necessary.

- (iv) The optimization routine continues as above, (hopefully) yielding $\theta^{(m)}$ and Σ_m , the posterior mode and covariance matrix of θ at the mode, respectively.
- 2. With $\theta^{(m)}$ and Σ_m , the Metropolis Markov Chain Monte Carlo routine begins. Set $h = 1$, $h \in [1, H_b + H_k]$, subscript b being the ‘burnin’ option and k being ‘keep’.
 - (i) Draw $\theta^{(*)}$ from $\mathcal{N}(\theta^{(h-1)}, c \cdot \Sigma_m)$, where $\theta^{(0)} = \theta^{(m)}$.
 - (ii) The parameters are transformed back to their standard values, then, using the function the user supplied, be used to solve the DSGE model.
 - (iii) With the solved model, we pass the state transition matrix to the Kalman filter, along with the user-supplied data and \mathcal{H} matrix. The Kalman filter will then return the likelihood value $\mathcal{L}(\theta | \mathcal{Y}, M_i)$.
 - (iv) Finally, the log-value of the posterior kernel,

$$\ln \mathcal{K}(\theta | \mathcal{Y}, M_i) = \ln \mathcal{L}(\theta | \mathcal{Y}, M_i) + \ln p(\theta | M_i)$$

is found by checking the values against the prior distributions $p(\theta | M_i)$ the user supplied, and adjusting the posterior for the Jacobian, should that be necessary.

- (v) Draw $v \sim \mathcal{U}(0, 1)$. If

$$v < \frac{\mathcal{K}(\theta^{(*)} | \mathcal{Y}, M_i)}{\mathcal{K}(\theta^{(h-1)} | \mathcal{Y}, M_i)}$$

set $\theta^{(h)} = \theta^{(*)}$, otherwise set $\theta^{(h)} = \theta^{(h-1)}$.

- (vi) If $h > H_b$, store the values.
- (vii) If $h = H_b + H_k$, stop, otherwise set $h = h + 1$ and go back to (i).

And we’re done.

8.7. DSGE Functions

8.7.1. Solve DSGE (gensys, uhlig, SDSGE)

`gensys(Gamma0, Gamma1, C, Psi, Pi)`

- `Gamma0`

Coefficients on present-time variables. This is matrix Γ_0 in (62).

- `Gamma1`

Coefficients on lagged variables. This is matrix Γ_1 in (62).

- `C`

Intercept terms. This is matrix C in (62).

- `Psi`

Coefficients on any exogenous shocks. This is matrix Ψ in (62).

- `Pi`

One-step-ahead expectational errors. This is matrix Π in (62).

The function returns an object of class ‘gensys’, which contains:

- `G1`

Autoregressive solution matrix. This is matrix F in (63).

- `Cons`

Intercept terms. This is matrix \tilde{C} in (63).

- `impact`

Coefficients on the exogenous shocks. This is matrix G in (63).

- `eu`

A 2×1 vector indicating existence and uniqueness (respectively) of the solution. A value of 1 can be read as ‘yes’, while 0 is ‘no’

- `Psi`

User-specified shock matrix.

- **Pi**

User-specified expectational errors matrix.

```
uhlig(A,B,C,D,F,G,H,J,K,L,M,N,whichEig=NULL)
```

- **A, B, C, D**

The ‘uhlig’ function requires the three blocks of matrices, with 12 matrices in total. The A , B , C , and D matrices form the deterministic block.

- **F, G, H, J, K, L, M**

The F , G , and H matrices form the expectational block for the control variables. The J and K matrices are for the ‘jump’ variables, and L and M are for the exogenous shocks.

- **N**

The N matrix defines the autoregressive structure of any exogenous shocks.

- **whichEig**

The function will return the eigenvalues and (right) eigenvectors used to construct the solution matrices, with the eigenvalues sorted in order of smallest to largest (in absolute value). By default, BMR will select the first (smallest) m eigenvalues (out of a total of $2m$ eigenvalues). However, if you prefer to select the eigenvalues yourself, then enter a numeric vector of length m indicating which elements of the eigenvalue matrix you wish to use.

The function returns an object of class ‘uhlig’, which contains:

- **N**

The user-specified N matrix, defining the autoregressive nature of any exogenous shocks.

- **P**

The P matrix from Uhlig’s solution.

- **Q**

The Q matrix from Uhlig’s solution.

- **R**

The R matrix from Uhlig’s solution.

- **S**

The S matrix from Uhlig's solution.

- **EigenValues**

The sorted eigenvalues that form the solution to the P matrix. If a situation of $\pm\infty$ in the real part of an eigenvalue (with a corresponding NaN-valued imaginary part) arises, the eigenvalue will be set to $1E+07 +0i$.

- **EigenVectors**

The eigenvectors corresponding to the sorted eigenvalues.

8.7.2. Simulate DSGE (DSGESim)

```
DSGESim((obj,shocks.cov,sim.periods,burnin=NULL,seedval=1122,hpfiltered=FALSE,lambda=1
```

- **obj**

An object of class 'SDSGE', 'gensys', or 'uhlig'. The user should first solve a model using one of the solver functions ('SDSGE', 'gensys', or 'uhlig'), then pass the solution to 'DSGESim'.

- **shocks.cov**

A matrix of size $k \times k$ that describes the covariance structure of the model shocks.

- **sim.periods**

The number of simulation periods the function should return.

- **burnin**

The length of sample burn-in. The default, 'burnin = NULL', will set burn-in to one-half of the number given in 'sim.periods'.

- **seedval**

Seed the random number generator.

- **hpfiltered**

Whether to pass the simulated series through a Hodrick-Prescott filter before retuning it.

- **lambda**

If ‘hpfiltered = TRUE’, then this is the value of the smoothing parameter in the H-P filter.

The function returns a matrix of size sim.periods $\times (m + n + k)$.

8.7.3. Estimate DSGE (EDSGE)

```
EDSGE(dsgedata, chains=1, cores=1,
       ObserveMat, initialvals, partomats,
       priorform, priorpars, parbounds, parnames=NULL,
       optimMethod="Nelder-Mead",
       optimLower=NULL, optimUpper=NULL,
       optimControl=list(),
       DSGEIRFs=TRUE, irf.periods=20,
       scalepar=1, keep=50000, burnin=10000,
       tables=TRUE)
```

- **dsgedata**

A matrix or data frame of size $T \times j$ containing the data series used for estimation.

- **chains**

A positive integer value indicating the number of MCMC chains to run.

- **cores**

A positive integer value indicating the number of CPU cores that should be used for estimation. This number should be less than or equal to the number of chains. **Do not allocate more cores than your computer can safely handle!**

- **ObserveMat**

The $(m + n + k) \times j$ observable matrix \mathcal{H} that maps the state variables to the observable series in the measurement equation.

- **initialvals**

Initial values to begin the optimization routine.

- **partomats**

This is perhaps the most important function input.

‘partomats’ should be a function that maps the deep parameters of the DSGE model to the matrices of a solution method, and contain: a $k \times k$ matrix labelled ‘shocks’ containing the variances of the structural shocks; a $j \times 1$ matrix labelled ‘MeasCons’ containing any constant terms in the measurement equation; and a $j \times j$ matrix labelled ‘MeasErrs’ containing the variances of the measurement errors..

- **priorform**

The prior distribution of each parameter.

- **priorpars**

The parameters of the relevant prior densities.

For example, if the user selects a Gaussian prior for a parameter, then the first entry will be the mean and the second its variance.

- **parbounds**

The lower- and (where relevant) upper-bounds on the parameter values.

- **parnames**

A character vector containing labels for the parameters.

- **optimMethod**

The optimization algorithm used to find the posterior mode. The user may select: the “Nelder-Mead” simplex method, which is the default; “BFGS”, a quasi-Newton method; “CG” for a conjugate gradient method; “L-BFGS-B”, a limited-memory BFGS algorithm with box constraints; or “SANN”, a simulated-annealing algorithm.

See `?optim` for more details.

If more than one method is entered, e.g., `c(Nelder-Mead, CG)`, optimization will proceed in a sequential manner, updating the initial values with the result of the previous optimization routine.

- **optimLower**

If `optimMethod="L-BFGS-B"`, this is the lower bound for optimization.

- **optimUpper**
If optimMethod="L-BFGS-B", this is the upper bound for optimization.
- **optimControl**
A control list for optimization. See ?optim in R for more details.
- **DSGEIRFs**
Whether to calculate impulse response functions.
- **irf.periods**
If DSGEIRFs=TRUE, then use this option to set the IRF horizon.
- **scalepar**
The scaling parameter, c , for the MCMC run.
- **keep**
The number of replications to keep. If keep is set to zero, the function will end with a normal approximation at the posterior mode.
- **burnin**
The number of sample burn-in points
- **tables**
Whether to print results of the posterior mode estimation and summary statistics of the MCMC run.

The function returns an object of class ‘EDSGE’, which contains:

- **Parameters**
A matrix with ‘keep × chains’ number of rows that contains the estimated, post sample burn-in parameter draws.
- **parMode**
Estimated posterior mode parameter values.
- **ModeHessian**
The Hessian computed at the posterior mode for the transformed parameters.

- **logMargLikelihood**

The log marginal likelihood from a Laplacian approximation at the posterior mode.

- **IRFs**

The IRFs (if any), based on the posterior parameter draws.

- **AcceptanceRate**

The acceptance rate of the chain(s).

- **RootRConvStats**

Gelman's \sqrt{R} -between-chain convergence statistics for each parameter. A value close 1 would signal convergence.

- **ObserveMat**

The user-supplied \mathcal{H} matrix from the Kalman filter recursion.

- **data**

The data used for estimation.

8.7.4. Prior Specification (prior)

```
prior(priorform,priorpars,parname=NULL,moments=TRUE,NR=NULL,NC=NULL)
```

- **priorform**

This should be a valid prior form for the EDSGE or DSGEVAR functions, such as “Gamma” or “Beta”.

- **priorpars**

The relevant parameters of the distribution.

- **parname**

A title for the plot.

- **moments**

Whether to print the mean, mode, and variance of the distribution.

- **NR**

For use with multiple plots.

- NC

For use with multiple plots.

9. DSGE Examples

This section provides two detailed examples of how to use DSGE-related functions in BMR, from simple simulation to estimation. The first example is a well-known real business cycle (RBC) model, and the second is the basic New-Keynesian model. The final model is included as an example of a ‘large’ model with many bells, and yet even more whistles.

9.1. RBC Model

We begin with a well-known RBC model; for those unfamiliar with RBC-type models, I recommend McCandless (2008) as a gentle introduction to dynamic macroeconomics. The social planner wants to maximise lifetime utility of a representative agent,

$$\max_{\{C_{t+i}, N_{t+i}\}} \mathbb{E}_t \left\{ \sum_{i=0}^{\infty} \beta^i \left(\frac{C_{t+i}^{1-\eta}}{1-\eta} - aN_{t+i} \right) \right\} \quad (84)$$

where $\beta \in (0, 1)$ is a discount factor, C_t is consumption at time t and N_t is labour supplied, and we assume constant relative risk aversion (CRRA) function for consumption; i.e., $u(C) = \frac{C^{1-\eta}}{1-\eta}$, where absolute and relative risk aversion are

$$R(C) = -\frac{u''(C)}{u'(C)} = -\frac{-\eta(1-\eta)C^{-\eta-1}}{(1-\eta)C^{-\eta}} = \frac{\eta}{C}$$

$$R_r(C) = -C \frac{u''(C)}{u'(C)} = \eta$$

respectively, $1/\eta$ is the intertemporal elasticity of substitution for consumption between periods. As $\eta \rightarrow 1$, we get log utility.⁹

⁹To see this, let $u(C) = \frac{C^{1-\eta}-1}{1-\eta}$, use l'Hôpital's rule and evaluate the limit $\eta \rightarrow 1$,

$$\lim_{\eta \rightarrow 1} \frac{\frac{d}{d\eta} C^{1-\eta} - 1}{\frac{d}{d\eta} 1 - \eta} = \lim_{\eta \rightarrow 1} \frac{-C^{1-\eta} \cdot \ln(C)}{-1} = \ln(C)$$

where, for the numerator, define $f := C^{1-\eta}$, take the log of both sides, and use implicit differentiation:

$$\frac{1}{f} \frac{df}{d\eta} = \frac{d}{d\eta} (1-\eta) \ln(C) \Rightarrow \frac{df}{d\eta} = f \cdot (-\ln(C)) = -C^{1-\eta} \cdot \ln(C)$$

This is subject to the constraints:

$$\begin{aligned} Y_t &= C_t + I_t \\ Y_t &= A_t K_{t-1}^\alpha N_t^{1-\alpha} \\ K_t &= I_t + (1 - \delta) K_{t-1} \end{aligned}$$

where Y_t is output, A_t is technology, K_t is the capital stock, I_t is investment, and δ is the depreciation rate of capital. The constraints can be combined to yield

$$A_t K_{t-1}^\alpha N_t^{1-\alpha} = C_t + K_t - (1 - \delta) K_{t-1}, \quad (85)$$

and technology is assumed to follow an AR(1) process,

$$\ln A_t = (1 - \rho) \ln A^* + \rho \ln A_{t-1} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma_a^2) \quad (86)$$

Set up the Lagrangian,

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_t \left[\sum_{i=0}^{\infty} \beta^i \left(\frac{C_{t+i}^{1-\eta}}{1-\eta} - a N_{t+i} \right) \right] \\ &\quad + \mathbb{E}_t \left[\sum_{i=0}^{\infty} \beta^i \lambda_{t+i} (A_{t+i} K_{t-1+i}^\alpha N_{t+i}^{1-\alpha} - C_{t+i} - K_{t+i} + (1 - \delta) K_{t-1+i}) \right] \end{aligned}$$

The first-order conditions of the problem are:

$$\begin{aligned} \mathcal{L}_{C_t} &= C_t^{-\eta} - \lambda_t = 0 \\ \mathcal{L}_{C_{t+1}} &= \beta \mathbb{E}_t [C_{t+1}^{-\eta} - \lambda_{t+1}] = 0 \\ \mathcal{L}_{K_t} &= -\lambda_t + \beta \mathbb{E}_t \left[\lambda_{t+1} \left(\alpha \frac{Y_{t+1}}{K_t} + 1 - \delta \right) \right] = 0 \\ \mathcal{L}_{N_t} &= -a + (1 - \alpha) \lambda_t \frac{Y_t}{N_t} = 0 \\ \mathcal{L}_{\lambda_t} &= A_t K_{t-1}^\alpha N_t^{1-\alpha} - C_t - K_t + (1 - \delta) K_{t-1} = 0 \end{aligned}$$

where a subscript on \mathcal{L} denotes a partial derivative with respect to that variable. Define

$$R_{t+1} := \alpha \frac{Y_{t+1}}{K_t} + 1 - \delta$$

so \mathcal{L}_{K_t} becomes

$$-\lambda_t + \beta \mathbb{E}_t [\lambda_{t+1} R_{t+1}] = 0.$$

We know from \mathcal{L}_{C_t} and $\mathcal{L}_{C_{t+1}}$ that $C_t^{-\eta} = \lambda_t$ and $C_{t+1}^{-\eta} = \lambda_{t+1}$, respectively, so

$$C_t^{-\eta} = \beta \mathbb{E}_t [C_{t+1}^{-\eta} R_{t+1}]$$

which means that \mathcal{L}_{N_t} is re-written as

$$\begin{aligned} -a + (1-\alpha) C_t^{-\eta} \frac{Y_t}{N_t} &= 0 \\ \Rightarrow \frac{Y_t}{N_t} &= \frac{a}{1-\alpha} C_t^\eta \end{aligned}$$

The full RBC model is given by the following 7 equations:

$$Y_t = C_t + I_t \quad (87)$$

$$Y_t = A_t K_{t-1}^\alpha N_t^{1-\alpha} \quad (88)$$

$$K_t = I_t + (1-\delta) K_{t-1} \quad (89)$$

$$R_t = \alpha \frac{Y_t}{K_{t-1}} + 1 - \delta \quad (90)$$

$$C_t^{-\eta} = \beta \mathbb{E}_t (C_{t+1}^{-\eta} R_{t+1}) \quad (91)$$

$$\frac{Y_t}{N_t} = \frac{a}{1-\alpha} C_t^\eta \quad (92)$$

$$\log A_t = (1-\rho) \log A^* + \rho \log A_{t-1} + \epsilon_t \quad (93)$$

To provide a form that we can use in BMR, we log-linearize the equilibrium conditions by taking a Taylor series expansion around stationary steady-state values. For a continuously differentiable function, $f(x) \in \mathbf{C}^\infty$ (continuously differentiable to an infinite order), we can approximate its form by ‘expanding’ around the centering value x_0 , which is given by the series:

$$\begin{aligned} f(x) &= f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 \\ &\quad + \frac{f^{(3)}(x_0)}{3!}(x - x_0)^3 + \dots + \frac{f^{(n-1)}(x_0)}{(n-1)!}(x - x_0)^{n-1} + R_n \end{aligned}$$

where

$$R_n = \frac{f^{(n)}(\xi)(x - x_0)^n}{n!}, \quad x_0 < \xi < x,$$

is the remainder; see Hoy et al. (2001). For example, if we take the exponential function e^x (*i.e.*, e being Euler's number, $e = \lim_{k \rightarrow \infty} (1 + \frac{1}{k})^k \approx 2.71828$) we approximate around the centering value x_0 by

$$e^x = e^{x_0} + \frac{e^{x_0}}{1!}(x - x_0)^1 + \frac{e^{x_0}}{2!}(x - x_0)^2 + \frac{e^{x_0}}{3!}(x - x_0)^3 + \frac{e^{x_0}}{4!}(x - x_0)^4 + \frac{e^{x_0}}{5!}(x - x_0)^5 + \dots$$

Setting $x_0 = 0$ and using the first-order approximation we get

$$e^x \approx e^0 + \frac{e^0}{1!}(x - 0)^1$$

or

$$e^x \approx 1 + x$$

as $e^0 = 1$. Note that this approximation holds only for values of x very close to zero.

The Uhlig (1999) method of log-linearisation is quite simple. As as an example, let

$$Y_t = X_t$$

which we can re-write as

$$Y_t = X_t \frac{X^*}{X^*}$$

or

$$Y_t = X^* \frac{X_t}{X^*}$$

where X^* is the 'steady-state value for X '; this is a trivial transformation because $\frac{X^*}{X^*} = 1$. We can take the exponential of a natural log of $\frac{X_t}{X^*}$ without changing anything on the left-hand side, since the logarithmic function (to the base of e) is the inverse of the exponential function.

So,

$$Y_t = X^* \exp\left(\ln\left(\frac{X_t}{X^*}\right)\right)$$

holds by definition. Remember that $\ln\left(\frac{X_t}{X^*}\right) = \ln(X_t) - \ln(X^*)$, so we set

$$x_t := \ln(X_t) - \ln(X^*)$$

as the deviation of X_t from its steady-state value, X^* . Therefore,

$$Y_t = X^* \exp(x_t)$$

also holds, by definition. Taking a first-order Taylor series expansion of $\exp(x_t)$ around the centering value of zero, as above, yields,

$$\exp(x_t) \approx 1 + x_t$$

so

$$Y_t \approx X^*(1 + x_t).$$

We will now apply this method to the RBC model.

9.1.1. Log-Linearising the RBC Model

For our first equilibrium condition,

$$Y_t = C_t + I_t$$

we implement the method described above to give

$$Y^*(1 + y_t) = C^*(1 + c_t) + I^*(1 + i_t)$$

and note that

$$Y^* = C^* + I^*$$

so expanding the equation out

$$Y^* + Y^* y_t = C^* + C^* c_t + I^* + I^* i_t$$

means that we can cancel $Y^* = C^* + I^*$. We're then left with

$$Y^* y_t = C^* c_t + I^* i_t$$

and writing y_t in terms of everything else,

$$y_t = \frac{C^* c_t + I^* i_t}{Y^*} \quad (94)$$

yields the first log-linearized equation.

For our second equilibrium condition,

$$Y_t = A_t K_{t-1}^\alpha N_t^{1-\alpha}$$

Use the method to give

$$Y^* \exp(y_t) = A^* \exp(a_t) (K^*)^\alpha \exp(\alpha k_{t-1}) (N^*)^{1-\alpha} \exp((1-\alpha)n_t)$$

Again, note that

$$Y^* = A^* (K^*)^\alpha (N^*)^{1-\alpha}$$

so we can divide across by Y^* to cancel some terms,

$$\exp(y_t) = \exp(a_t) \exp(\alpha k_{t-1}) \exp((1-\alpha)n_t)$$

The first-order approximation is then

$$(1 + y_t) = (1 + a_t)(1 + \alpha k_{t-1})(1 + (1-\alpha)n_t)$$

If we ignore cross-products,

$$y_t = a_t + \alpha k_{t-1} + (1-\alpha)n_t \quad (95)$$

which is the second log-linearized equation.

Our third equilibrium condition is

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

Re-writing

$$K^*(1 + k_t) = I^*(1 + i_t) + (1 - \delta)K^*(1 + k_{t-1})$$

or

$$\begin{aligned} K^*k_t &= I^*i_t + (1 - \delta)K^*k_{t-1} \\ \Rightarrow k_t &= \frac{I^*i_t + (1 - \delta)K^*k_{t-1}}{K^*} \end{aligned} \quad (96)$$

The fourth equilibrium condition is

$$R_t = \alpha \frac{Y_t}{K_{t-1}} + 1 - \delta,$$

Multiply both sides by K_{t-1} to make this a little easier,

$$R_t K_{t-1} = \alpha Y_t + (1 - \delta)K_{t-1}$$

so

$$R^*K^*(1 + r_t + k_{t-1}) = \alpha Y^*(1 + y_t) + (1 - \delta)K^*(1 + k_{t-1})$$

canceling and dividing across

$$r_t + k_{t-1} = \frac{\alpha Y^* y_t}{R^* K^*} + \frac{(1 - \delta)K^* k_{t-1}}{R^* K^*}$$

or

$$\begin{aligned} r_t &= \frac{\alpha Y^* y_t}{R^* K^*} + \frac{(1 - \delta)K^* k_{t-1}}{R^* K^*} - k_{t-1} \\ &= \frac{\alpha Y^* y_t}{R^* K^*} + \frac{K^* k_{t-1} - \delta K^* k_{t-1} - R^* K^* k_{t-1}}{R^* K^*} \\ \Rightarrow r_t &= \frac{\alpha Y^* y_t}{R^* K^*} + \frac{K^* k_{t-1}(1 - \delta - R^*)}{R^* K^*} \end{aligned}$$

Now note that,

$$R^*K^* = \alpha Y^* + (1 - \delta)K^*$$

so

$$\begin{aligned} Y^* &= \frac{R^*K^* - (1-\delta)K^*}{\alpha} \\ &= \frac{K^*(-1 + \delta - R^*)}{\alpha} \\ \Rightarrow Y^* &= \frac{-K^*(1 - \delta + R^*)}{\alpha} \end{aligned}$$

Notice that this appears in the right hand side of

$$r_t = \frac{\alpha Y^* y_t}{R^* K^*} + \frac{k_{t-1} K^* (1 - \delta - R^*)}{R^* K^*}$$

which implies

$$r_t = \frac{\alpha Y^* y_t}{R^* K^*} + \frac{k_{t-1} (-Y^* \alpha)}{R^* K^*}$$

and if we re-write

$$r_t = \frac{\alpha Y^*}{R^* K^*} (y_t - k_{t-1}). \quad (97)$$

The fifth equation is

$$C_t^{-\eta} = \beta \mathbb{E}_t(C_{t+1}^{-\eta} R_{t+1})$$

Begin as before

$$(C^*)^{-\eta} (1 - \eta c_t) = \beta \mathbb{E}_t[(C^*)^{-\eta} (1 - \eta c_{t+1})(R^*)(1 + r_{t+1})]$$

Note that,

$$(C^*)^{-\eta} = \beta \mathbb{E}_t[(C^*)^{-\eta} (R^*)]$$

where the expectation is trivial. So,

$$(1 - \eta c_t) = \mathbb{E}_t[(1 - \eta c_{t+1})(1 + r_{t+1})]$$

ignoring cross-products,

$$-\eta c_t = \mathbb{E}_t[-\eta c_{t+1} + r_{t+1}]$$

so

$$c_t = \mathbb{E}_t c_{t+1} - \frac{1}{\eta} \mathbb{E}_t r_{t+1} \quad (98)$$

where we divide both sides by $-\eta$, and note that the expectation of a sum is the sum of the expectations.

Almost finished! The sixth equation is

$$\frac{Y_t}{N_t} = \frac{a}{1-\alpha} C_t^\eta.$$

Re-write,

$$Y_t = \frac{a}{1-\alpha} C_t^\eta N_t$$

Follow the same procedure as before,

$$Y^*(1+y_t) = \frac{a}{1-\alpha} (C^*)^\eta (1+\eta c_t) N^*(1+n_t)$$

Note that

$$Y^* = \frac{a}{1-\alpha} (C^*)^\eta N^*$$

so dividing across we get,

$$(1+y_t) = (1+\eta c_t)(1+n_t)$$

Expand this out and ignore cross-products,

$$y_t = \eta c_t + n_t$$

which can be written as

$$n_t = y_t - \eta c_t \tag{99}$$

The final equation describing the exogenous process is simple.

$$\begin{aligned} \log A_t &= (1-\rho) \log A^* + \rho \log A_{t-1} + \epsilon_t \\ \log A_t &= \log A^* - \rho \log A^* + \rho \log A_{t-1} + \epsilon_t \\ (\log A_t - \log A^*) &= \rho(\log A_{t-1} - \log A^*) + \epsilon_t \end{aligned}$$

which we can re-write as

$$a_t = \rho a_{t-1} + \epsilon_t \tag{100}$$

Finally, if we combine these equations,

$$y_t = \frac{C^* c_t + I^* i_t}{Y^*} \quad (101)$$

$$y_t = a_t + \alpha k_{t-1} + (1 - \alpha) n_t \quad (102)$$

$$k_t = \frac{I^* i_t + (1 - \delta) K^* k_{t-1}}{K^*} \quad (103)$$

$$r_t = \frac{\alpha Y^*}{R^* K^*} (y_t - k_{t-1}) \quad (104)$$

$$c_t = \mathbb{E}_t c_{t+1} - \frac{1}{\eta} \mathbb{E} r_{t+1} \quad (105)$$

$$n_t = y_t - \eta c_t \quad (106)$$

$$a_t = \rho a_{t-1} + \epsilon_t \quad (107)$$

we have a new set of log-linearized equations as our system that describes the equilibrium of the economy.

However, we still need to calculate R^* , $\frac{C^*}{Y^*}$, $\frac{I^*}{Y^*}$, and $\frac{Y^*}{K^*}$. In the fifth equilibrium condition we have

$$C_t^{-\eta} = \beta \mathbb{E}_t (C_{t+1}^{-\eta} R_{t+1})$$

or

$$1 = \beta \mathbb{E}_t \left(\left(\frac{C_t}{C_{t+1}} \right)^{\eta} R_{t+1} \right)$$

In the steady-state, $C_t = C_{t+1} = C^*$, and $R_{t+1} = R^*$,

$$\begin{aligned} 1 &= \beta \mathbb{E}_t \left(\left(\frac{C^*}{C^*} \right)^{\eta} R^* \right) \\ 1 &= \beta \mathbb{E}_t (R^*) \\ \Rightarrow R^* &= \frac{1}{\beta} \end{aligned} \quad (108)$$

Using the fourth equation,

$$R_t = \alpha \frac{Y_t}{K_{t-1}} + 1 - \delta$$

we know from the previous steady-state that

$$\begin{aligned}
 R^* &= \frac{1}{\beta} = \alpha \frac{Y^*}{K^*} + 1 - \delta \\
 \beta^{-1} - (1 - \delta) &= \alpha \frac{Y^*}{K^*} \\
 \frac{\beta^{-1} - (1 - \delta)}{\alpha} &= \frac{Y^*}{K^*} \\
 \Rightarrow \frac{Y^*}{K^*} &= \frac{\beta^{-1} - 1 + \delta}{\alpha}
 \end{aligned} \tag{109}$$

Now recall the third equation

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

In the steady-state, $K_t = K_{t-1} = K^*$, so

$$K^* = I^* + K^* - \delta K^*$$

canceling

$$I^* = \delta K^*$$

or

$$\frac{I^*}{K^*} = \delta \tag{110}$$

To get $\frac{C^*}{Y^*}$, we use the steady-state of the first equilibrium condition

$$Y^* = C^* + I^*$$

or

$$\frac{C^*}{Y^*} = 1 - \frac{I^*}{Y^*}$$

We've already found that

$$\frac{I^*}{K^*} = \delta$$

and

$$\frac{Y^*}{K^*} = \frac{\beta^{-1} - 1 + \delta}{\alpha}$$

Dividing the first by the second

$$\frac{\frac{I^*}{K^*}}{\frac{Y^*}{K^*}} = \frac{I^*}{Y^*} = \frac{\delta}{\frac{\beta^{-1}-1+\delta}{\alpha}}$$

which can be re-written as

$$\frac{I^*}{Y^*} = \frac{\alpha\delta}{\beta^{-1}-1+\delta}$$

Therefore,

$$\frac{C^*}{Y^*} = 1 - \frac{\alpha\delta}{\beta^{-1}-1+\delta} \quad (111)$$

And we're finished.

9.1.2. Inputting the Model to BMR

Now that we have a system of log-linearized equations, we need to set the model up in the form

$$\begin{aligned} 0 &= A\xi_{1,t} + B\xi_{1,t-1} + C\xi_{2,t} + Dz_t \\ 0 &= \mathbb{E}_t \{ F\xi_{1,t+1} + G\xi_{1,t} + H\xi_{1,t-1} + J\xi_{2,t+1} + K\xi_{2,t} + Lz_{t+1} + Mz_t \} \\ z_t &= Nz_{t-1} + \varepsilon_t \end{aligned}$$

to find the solution

$$\begin{aligned} \xi_{1,t} &= P\xi_{1,t-1} + Qz_t \\ \xi_{2,t} &= R\xi_{1,t-1} + Sz_t \\ z_t &= Nz_{t-1} + \varepsilon_t \end{aligned}$$

using Uhlig's method of undetermined coefficients. First, break up the equations into three blocks. The first block relates to the pre-determined variables, $A\xi_{1,t} + B\xi_{1,t-1} + C\xi_{2,t} + Dz_t$,

$$\begin{aligned} 0 &= y_t - \frac{C^*}{Y^*}c_t - \frac{I^*}{Y^*}i_t \\ 0 &= y_t - a_t - \alpha k_{t-1} - (1-\alpha)n_t \\ 0 &= k_t - \frac{I^*}{K^*}i_t - (1-\delta)k_{t-1} \\ 0 &= n_t - y_t + \eta c_t \\ 0 &= r_t - \frac{\alpha Y^*}{R^* K^*}y_t + \frac{\alpha Y^*}{R^* K^*}k_{t-1} \end{aligned}$$

The second block contains our expectational equations, of which we have only one:

$$0 = c_t - \mathbb{E}_t c_{t+1} + \frac{1}{\eta} \mathbb{E} r_{t+1}$$

and the third is our exogenous process

$$a_t = \rho a_{t-1} + \varepsilon_t$$

In order, the vectors containing our variables are

$$\begin{aligned}\xi_{1,t} &= [k_t] \\ \xi_{2,t} &= [c_t \ y_t \ n_t \ r_t \ i_t]^\top \\ z_t &= [a_t]\end{aligned}$$

The coefficient matrices are as follows. For the first block:

$$\begin{aligned}A &= [0 \ 0 \ 1 \ 0 \ 0]^\top \\ B &= [0 \ -\alpha \ -(1-\delta) \ 0 \ (\alpha/R^*)(Y^*/K^*)]^\top \\ C &= \begin{bmatrix} -C^*/Y^* & 1 & 0 & 0 & -I^*/Y^* \\ 0 & 1 & -(1-\alpha) & 0 & 0 \\ 0 & 0 & 0 & 0 & -I^*/K^* \\ \eta & -1 & 1 & 0 & 0 \\ 0 & -(\alpha/R^*)(Y^*/K^*) & 0 & 1 & 0 \end{bmatrix} \\ D &= [0 \ -1 \ 0 \ 0 \ 0]^\top\end{aligned}$$

The second block:

$$\begin{aligned}F &= G = H = L = M = [0] \\ J &= [-1 \ 0 \ 0 \ 1/\eta \ 0] \\ K &= [1 \ 0 \ 0 \ 0 \ 0]\end{aligned}$$

For the exogenous process:

$$N = [\rho]$$

Once we have defined these matrices, we use the ‘uhlig’ function in BMR:

```
dsgetest <- uhlig(A,B,C,D,F,G,H,J,K,L,M,N)
```

This will return the four (previously) unknown matrices P , Q , R , and S . As a numerical example, let's use the following parameter values: $\beta = 0.99$, $\alpha = 0.33$, $\delta = 0.015$, $\eta = 1$, $\rho = 0.95$, and $\sigma_a = 1$. The solution

$$\begin{aligned}\xi_{1,t} &= P\xi_{1,t-1} + Qz_t \\ \xi_{2,t} &= R\xi_{1,t-1} + Sz_t \\ z_t &= Nz_{t-1} + \varepsilon_t\end{aligned}$$

is then

$$P = [0.9519702]$$

$$Q = [0.1362265]$$

$$R = [0.50624359 \ -0.02782790 \ -0.53407149 \ -0.02554152 \ -2.20198548]^\top$$

$$S = [0.43745335 \ 2.14214017 \ 1.70468682 \ 0.05323218 \ 9.08176873]^\top$$

with N as before.

With this solution, we can build our state space structure,

$$\xi_t = \mathcal{F}\xi_{t-1} + \mathcal{G}\varepsilon_t$$

as discussed in the previous section (BMR will do this for you), and, to produce IRFs, the relevant BMR function is

```
IRF(dsgetest, shock=1, irf.periods=30, varnames=c("Capital", "Consumption",
"Output", "Labour", "Interest", "Investment", "Technology"), save=TRUE)
```

the result of which is shown below, in figure 26

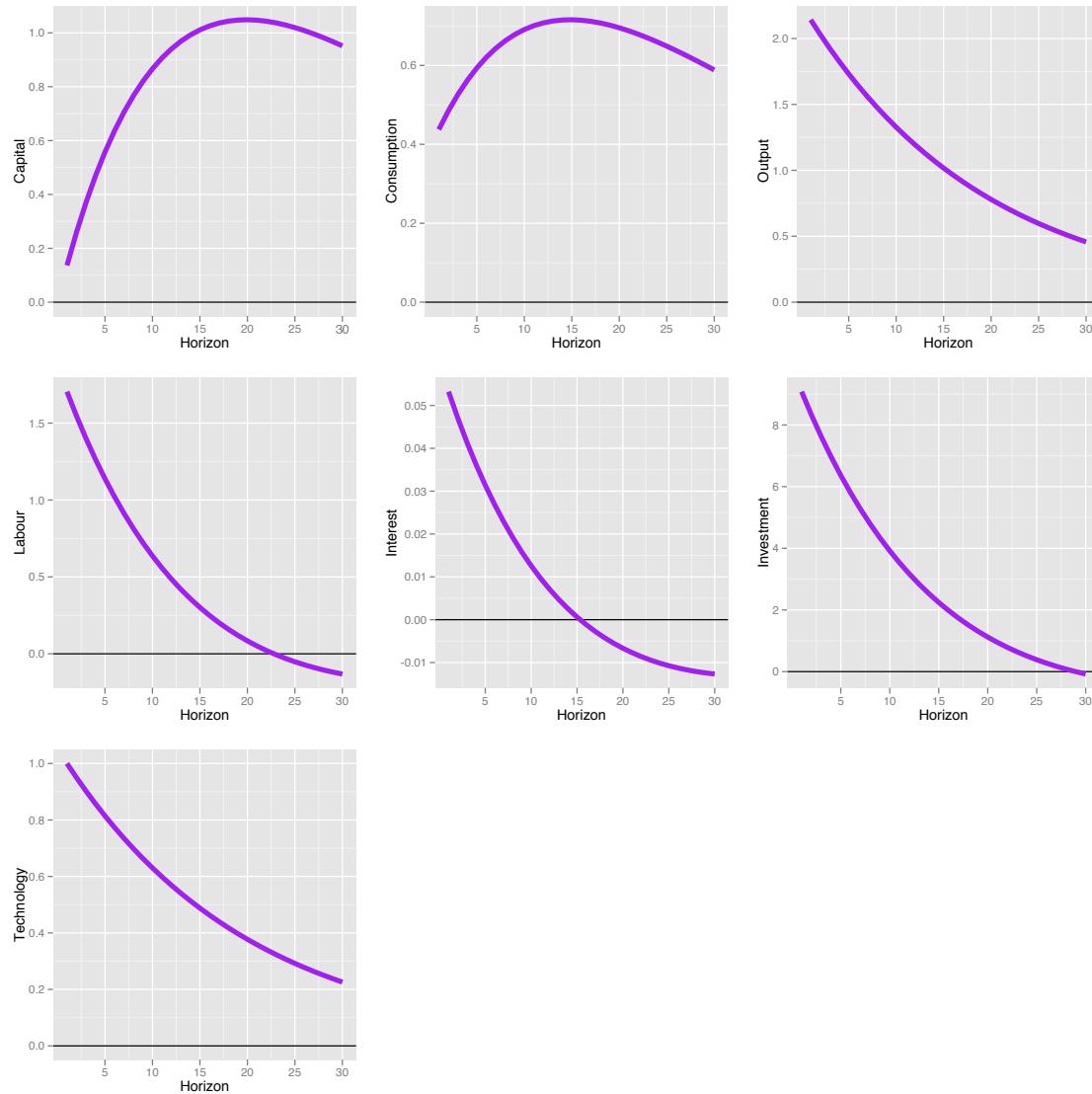


Figure 26: RBC Model: Shock to Technology.

9.1.3. Estimation

We now look to illustrate the use of DSGE estimation functions in BMR. First, we can simulate data from the RBC model to use in estimation. The parameter values are the same as in the previous section— $\alpha = 0.33$, $\delta = 0.015$, $\eta = 1$, $\rho = 0.95$, and $\sigma_a = 1$ —but with $\beta = 0.97$, and we will fix β at this value during estimation as it is not identified.

Use the ‘DSGESim’ function to generate data as follows:

```
dsgetestsim <- DSGESim(dsgetest, 1122, 1, 200, 200, hpfILTERED=FALSE)
```

Using the solved model contained in ‘dsgetest’, this will generate 400 data points, 200 of which are used for sample burn-in, and the final 200 are returned to the ‘dsgetestsim’ object. The data are displayed in figure 27, where, for estimation, we use the output series.

Next we define the observable matrix, \mathcal{H} from the measurement equation

```
ObserveMat <- rbind(0, 0, 1, 0, 0, 0)
```

set the initial values of α , δ , η , ρ , and σ_a as

```
initialvals <- c(0.28, 0.015, 1, 0.9, 1)
```

set the parameter names

```
parnames <- c("Alpha", "Delta", "Eta", "Rho", "SigmaA")
```

give functional forms to the prior distributions; select the relevant parameters for the prior distributions; and place upper- and lower-bounds where necessary.

```
priorform <- c("Normal", "Normal", "Normal", "Beta", "IGamma")
```

```
priorpars <- cbind(c( 0.30,    0.015,      1,     4,   2),
                     c(0.05^2, 0.002^2, 0.2^2, 1.5, 1))
```

```
parbounds <- cbind(c(NA, NA, NA, 0.5, 0),
                     c(NA, NA, NA, 0.999, NA))
```

We then call the ‘EDSGE’ function using these inputs. To find the posterior mode, we will use the Nelder-Mead simplex algorithm. For the MCMC run, set the number of chains and CPU cores to 1, with scaling parameter of 0.75, keep 50,000 draws and discard 20,000 as sample burn-in.

```
RBCest <- EDSGE(dsgeadata,chains=1,cores=1,
                  ObserveMat,initialvals,partomats,
                  priorform,priorpars,parbounds,parnames,
                  optimMethod="Nelder-Mead",
                  optimLower=NULL,optimUpper=NULL,
                  DSGEIRFs=TRUE,irf.periods=30,
                  scalepar=0.75,keep=50000,burnin=20000)
```

Trying to solve the model with your initial values... Done.

Beginning optimization, Tue Jul 8 11:14:25 2014.

Using Optimization Method: Nelder-Mead.

Optimization over, Tue Jul 8 11:14:25 2014.

Optimizer Convergence Code: 0; successful completion.

Optimizer Iterations:

function gradient

412 NA

Log Marginal Likelihood: -432.1544.

Parameter Estimates and Standard Errors (SE) at the Posterior Mode:

	Estimate	SE
Alpha	0.31179806	0.046665668
Delta	0.01499326	0.002000901
Eta	0.94864026	0.204332224
Rho	0.94495738	0.022811016

```
SigmaA 1.01446782 0.116342248
```

Beginning MCMC run, Tue Jul 8 11:14:25 2014.

MCMC run finished, Tue Jul 8 11:15:24 2014.

Acceptance Rate: 0.37422.

Parameter Estimates and Standard Errors:

	Posterior.Mode	SE.Mode	Posterior.Mean	SE.Posterior
Alpha	0.31179806	0.046665668	0.31005536	0.046716633
Delta	0.01499326	0.002000901	0.01499752	0.002000388
Eta	0.94864026	0.204332224	0.94019062	0.207895270
Rho	0.94495738	0.022811016	0.94793787	0.019871423
SigmaA	1.01446782	0.116342248	1.03093146	0.136488835

Computing IRFs now... Done.

With our estimated model, we can plot the parameter values using

```
plot(RBCest,save=TRUE)
```

shown in figure 28, and the impulse response functions with

```
IRF(RBCest, FALSE, varnames=c("Capital", "Consumption", "Output", "Labour",
"Interest", "Investment", "Technology"), save=TRUE)
```

shown in figure 29.

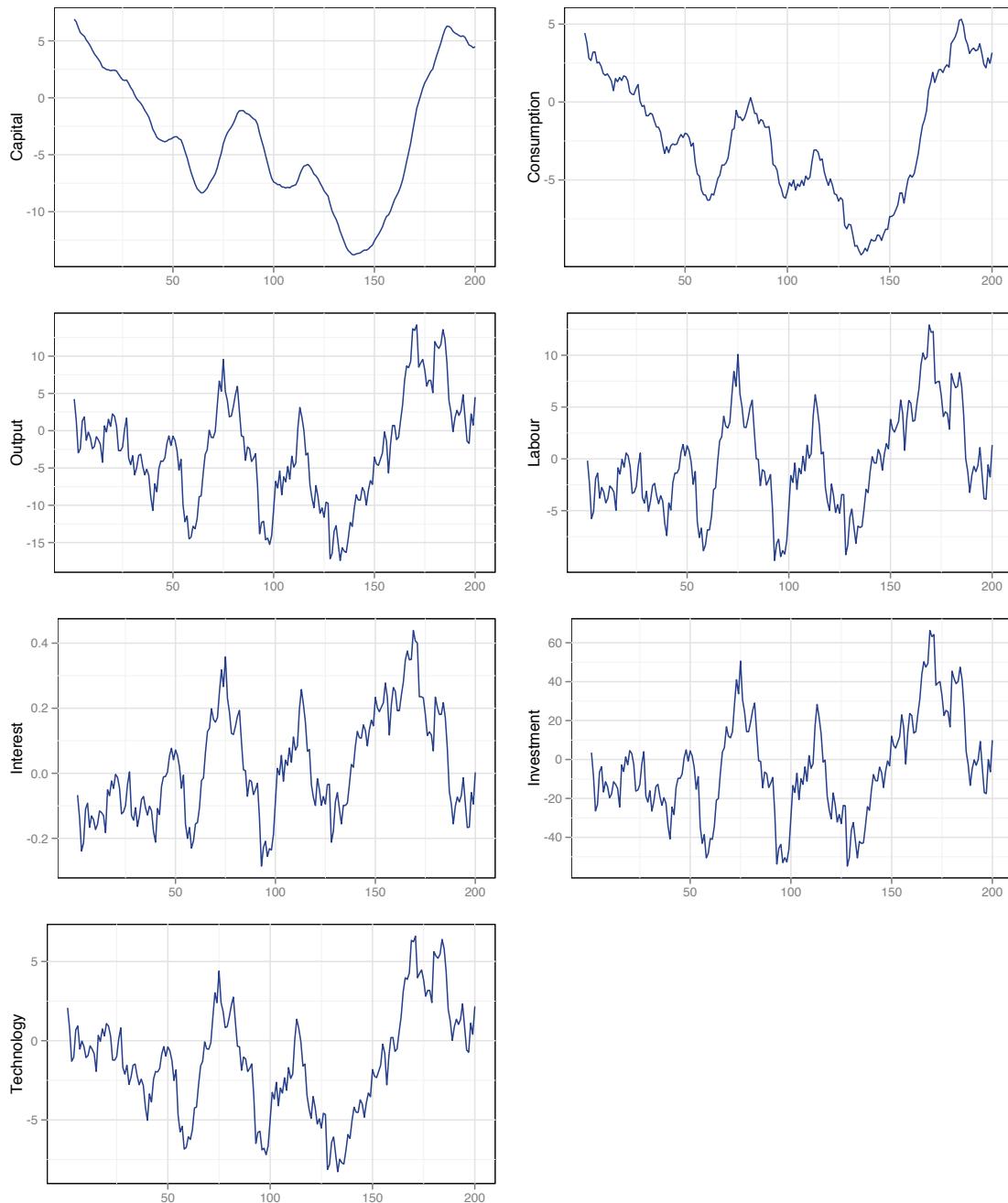


Figure 27: RBC Model: Simulated Data.

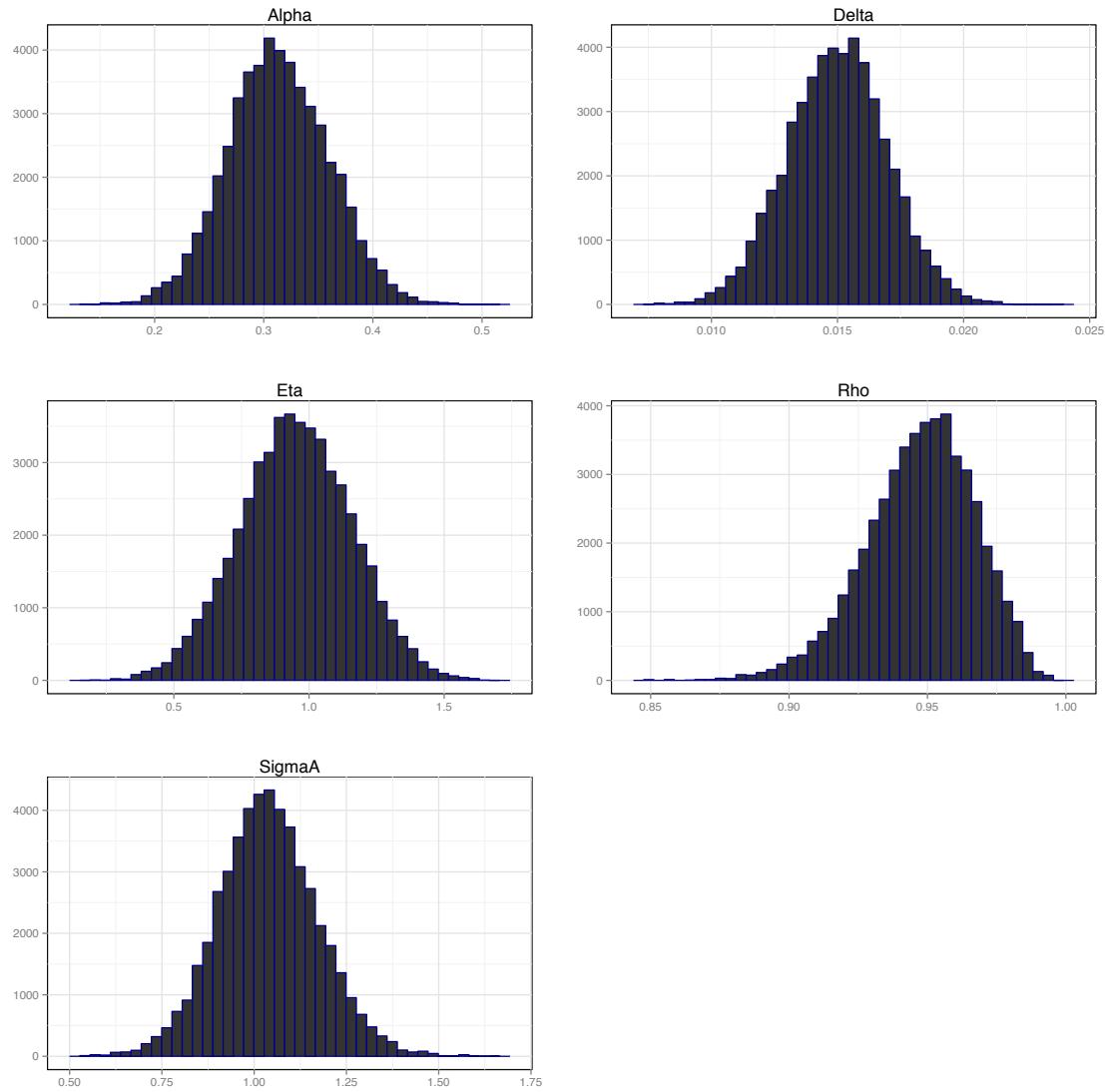


Figure 28: RBC Model: Posterior Distributions of θ .

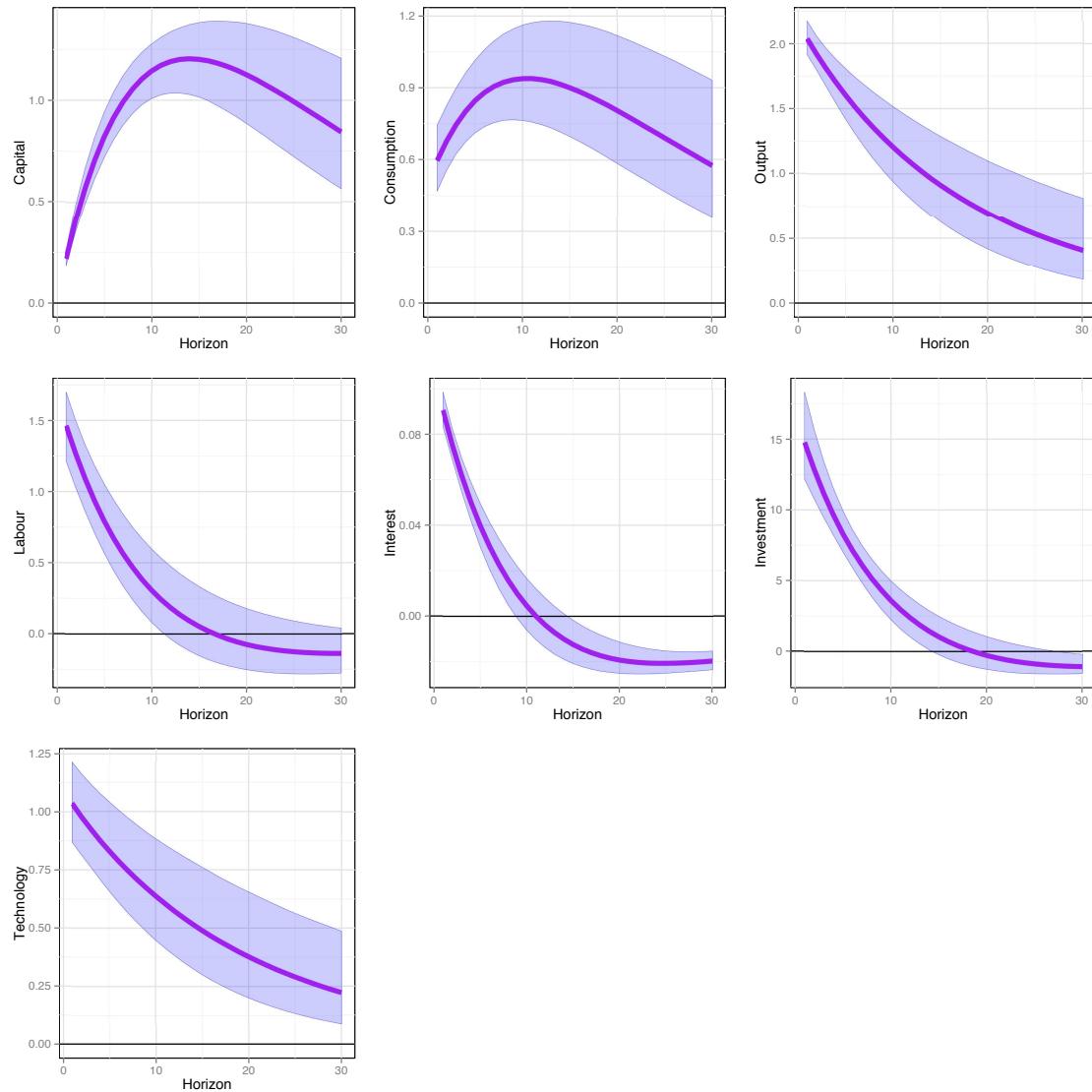


Figure 29: RBC Model: Median Bayesian IRFs with 10th and 90th Percentiles.

9.2. The Basic New-Keynesian Model

This section details the well-known basic New-Keynesian model; as references see (Galí, 2008, Ch. 3), (Lawless and Whelan, 2011, Appendix), and (Woodford, 2003, Ch. 3). The model consists of three main endogenous processes: inflation, real output, and a nominal interest rate, represented in the form of a Phillips curve (of sorts), a forward-looking IS equation, and a monetary policy reaction function, respectively. The primary distinction with the RBC model discussed previously, is the inclusion of nominal rigidities, in the form of staggered pricing by firms, and monopolistic competition in the production sector. These rigidities imply the non-neutrality of money in the short-run, and thus a role to the central bank in minimizing welfare losses arising from business cycles.

9.2.1. Dixit-Stiglitz Aggregators

Dixit-Stiglitz aggregators are abound in the DSGE literature, so I will derive the demand functions implied by this approach. In this framework, a firm's production is a function of a composite of goods, indexed over the continuum $[0, 1]$, represented by a Dixit-Stiglitz aggregator

$$Y_t := \left(\int_0^1 Y_t(i)^{\frac{\vartheta-1}{\vartheta}} di \right)^{\frac{\vartheta}{\vartheta-1}}, \quad (112)$$

where $\vartheta > 1$ is the elasticity of substitution between goods and $i \in [0, 1]$ serves to index integration; ϑ also has an interpretation as a markup over marginal cost, which we will see later in this section. The firm's problem is to minimise cost

$$\int_0^1 P_t(i)Y_t(i)di := E_t,$$

subject to (112), where P_t is price and E_t is total expenditure (at time t). (The problem can equally be restated as profit maximization, rather than minimizing cost.) The standard constrained optimization approach applies, so we can set up a Lagrangian

$$\mathcal{L} = \int_0^1 P_t(i)Y_t(i)di + \lambda \left(Y_t - \left(\int_0^1 Y_t(i)^{\frac{\vartheta-1}{\vartheta}} di \right)^{\frac{\vartheta}{\vartheta-1}} \right). \quad (113)$$

The simplest way to derive the demand functions is to maximise (113) for any two arbitrary

goods. Take the first-order condition (FOC) of (113) for, say, good j ,

$$P_t(j) - \lambda \frac{\vartheta}{\vartheta-1} \left(\int_0^1 Y_t(i)^{\frac{\vartheta-1}{\vartheta}} di \right)^{\frac{\vartheta}{\vartheta-1}-1} \frac{\vartheta-1}{\vartheta} Y_t(j)^{\frac{\vartheta-1}{\vartheta}-1} = 0 \quad (114)$$

Now, do the same for, say, good k ,

$$P_t(k) - \lambda \frac{\vartheta}{\vartheta-1} \left(\int_0^1 Y_t(i)^{\frac{\vartheta-1}{\vartheta}} di \right)^{\frac{\vartheta}{\vartheta-1}-1} \frac{\vartheta-1}{\vartheta} Y_t(k)^{\frac{\vartheta-1}{\vartheta}-1} = 0 \quad (115)$$

Divide (114) by (115),

$$\frac{\lambda \frac{\vartheta}{\vartheta-1} \left(\int_0^1 Y_t(i)^{\frac{\vartheta-1}{\vartheta}} di \right)^{\frac{\vartheta}{\vartheta-1}-1} \frac{\vartheta-1}{\vartheta} Y_t(j)^{\frac{\vartheta-1}{\vartheta}-1}}{\lambda \frac{\vartheta}{\vartheta-1} \left(\int_0^1 Y_t(i)^{\frac{\vartheta-1}{\vartheta}} di \right)^{\frac{\vartheta}{\vartheta-1}-1} \frac{\vartheta-1}{\vartheta} Y_t(k)^{\frac{\vartheta-1}{\vartheta}-1}} = \frac{P_t(j)}{P_t(k)} \quad (116)$$

Notice that we can cancel quite a lot from (116), so we're left with

$$\frac{Y_t(j)^{\frac{\vartheta-1}{\vartheta}-1}}{Y_t(k)^{\frac{\vartheta-1}{\vartheta}-1}} = \frac{P_t(j)}{P_t(k)} \quad (117)$$

Now for some manipulation of (117):

$$\begin{aligned} \frac{Y_t(j)^{\frac{\vartheta-1-\vartheta}{\vartheta}}}{Y_t(k)^{\frac{\vartheta-1-\vartheta}{\vartheta}}} &= \frac{P_t(j)}{P_t(k)} \\ \frac{Y_t(j)^{-\frac{1}{\vartheta}}}{Y_t(k)^{-\frac{1}{\vartheta}}} &= \frac{P_t(j)}{P_t(k)} \\ \frac{Y_t(j)}{Y_t(k)} &= \left(\frac{P_t(j)}{P_t(k)} \right)^{-\vartheta} \\ Y_t(j) &= P_t(j)^{-\vartheta} P_t(k)^{\vartheta} Y_t(k) \end{aligned}$$

Multiply both sides by $P_t(j)$ and integrate over good j ,

$$\begin{aligned} Y_t(j)P_t(j) &= P_t(j)P_t(j)^{-\vartheta}P_t(k)^\vartheta Y_t(k) \\ \int_0^1 Y_t(j)P_t(j) dj &= \int_0^1 P_t(j)^{1-\vartheta}P_t(k)^\vartheta Y_t(k) dj \\ \int_0^1 Y_t(j)P_t(j) dj &= P_t(k)^\vartheta Y_t(k) \int_0^1 P_t(j)^{1-\vartheta} dj \end{aligned}$$

where $\int_0^1 Y_t(j)P_t(j) dj = E_t$, by definition. Now define the price index by:

$$P_t := \left(\int_0^1 P_t(i)^{1-\vartheta} di \right)^{\frac{1}{1-\vartheta}} \quad (118)$$

which we can rewrite as

$$P_t^{1-\vartheta} = \left(\int_0^1 P_t(i)^{1-\vartheta} di \right).$$

Using this equation,

$$\begin{aligned} \int_0^1 Y_t(j)P_t(j) dj &= P_t(k)^\vartheta Y_t(k) \int_0^1 P_t(j)^{1-\vartheta} dj \\ E_t &= P_t(k)^\vartheta Y_t(k) P_t^{1-\vartheta} \\ \Rightarrow Y_t(k) &= \frac{E_t}{P_t(k)^\vartheta P_t^{1-\vartheta}} \\ Y_t(k) &= \frac{1}{P_t(k)^\vartheta P_t^{-\vartheta}} \frac{E_t}{P_t} \\ \Rightarrow Y_t(k) &= \left(\frac{P_t(k)}{P_t} \right)^{-\vartheta} \frac{E_t}{P_t} \end{aligned} \quad (119)$$

If we substitute (119) into the definition $Y_t = \left(\int_0^1 Y_t(k)^{\frac{\vartheta-1}{\vartheta}} dk \right)^{\frac{\vartheta}{\vartheta-1}}$, we get

$$\begin{aligned} Y_t &= \left(\int_0^1 \left(\left(\frac{P_t(k)}{P_t} \right)^{-\vartheta} \frac{E_t}{P_t} \right)^{\frac{\vartheta-1}{\vartheta}} dk \right)^{\frac{\vartheta}{\vartheta-1}} \\ &= \frac{E_t}{P_t} \left(\int_0^1 \left(\left(\frac{P_t(k)}{P_t} \right)^{-\vartheta} \right)^{\frac{\vartheta-1}{\vartheta}} dk \right)^{\frac{\vartheta}{\vartheta-1}} \\ &= \frac{E_t}{P_t} \left(\int_0^1 P_t(k)^{-(\vartheta-1)} P_t^{\vartheta-1} dk \right)^{\frac{\vartheta}{\vartheta-1}} \\ &= \frac{E_t}{P_t} P_t^\vartheta \left(\int_0^1 P_t(k)^{1-\vartheta} dk \right)^{\frac{\vartheta}{\vartheta-1}} \end{aligned}$$

Now recall that the price index is $P_t = \left(\int_0^1 P_t(i)^{1-\vartheta} di \right)^{\frac{1}{1-\vartheta}}$, so raising both sides by the exponent $-\vartheta$

$$\begin{aligned} P_t^{-\vartheta} &= \left(\int_0^1 P_t(i)^{1-\vartheta} di \right)^{\frac{-\vartheta}{1-\vartheta}} \\ &= \left(\int_0^1 P_t(i)^{1-\vartheta} di \right)^{\frac{\vartheta}{\vartheta-1}} \end{aligned}$$

which we can use on

$$Y_t = \frac{E_t}{P_t} P_t^\vartheta \left(\int_0^1 P_t(k)^{1-\vartheta} dk \right)^{\frac{\vartheta}{\vartheta-1}}$$

to get

$$\begin{aligned} Y_t &= \frac{E_t}{P_t} P_t^\vartheta P_t^{-\vartheta} \\ &= E_t P_t^{-1+\vartheta-\vartheta} = E_t P_t^{-1} \\ \Rightarrow P_t Y_t &= E_t \end{aligned}$$

i.e., the price index times the quantity index equals expenditure (surprise, surprise!). As

$$\int_0^1 P_t(k) Y_t(k) dk = E_t,$$

we see that

$$\int_0^1 P_t(k)Y_t(k)dk = P_t Y_t.$$

Using $Y_t = E_t/P_t$ we can substitute into

$$Y_t(k) = \left(\frac{P_t(k)}{P_t} \right)^{-\vartheta} \frac{E_t}{P_t}$$

to get the final equation

$$Y_t(k) = \left(\frac{P_t(k)}{P_t} \right)^{-\vartheta} Y_t \quad (120)$$

which is the demand function for an arbitrary good k .

9.2.2. The Household

We begin with the maximization problem faced by the household, from which we will derive a relationship between output and interest rates. The household wants to maximise

$$\sum_{k=0}^{\infty} \beta^k U(C_{t+k}, N_{t+k}) := \sum_{k=0}^{\infty} \beta^k \mathbb{E}_t \left\{ \frac{C_{t+k}^{1-\frac{1}{\eta}}}{1 - \frac{1}{\eta}} - \frac{N_{t+k}^{1+\phi}}{1 + \phi} \right\} \quad (121)$$

where $C_t = \left(\int_0^1 C_t(i)^{\frac{\vartheta-1}{\vartheta}} di \right)^{\frac{\vartheta}{\vartheta-1}}$ is consumption and N_t is labour supplied (with a factor price of W_t), subject to a sequence of intertemporal budget constraints

$$P_t C_t + B_t = W_t N_t + (1 + i_{t-1}) B_{t-1} \quad (122)$$

every period, where B_t is any bond holdings in period t and i_t is the nominal interest rate.

The first-order conditions with respect to B_t , C_t , C_{t+1} , and N_t of this problem are

$$\begin{aligned} \mathbb{E}_t \{ \lambda_{t+1}(1 + i_t) - \lambda_t \} &= 0 \\ C_t^{-1/\eta} + \lambda_t P_t &= 0 \\ \mathbb{E}_t \beta \{ C_{t+1}^{-1/\eta} + \lambda_{t+1} P_{t+1} \} &= 0 \\ -N_t^\phi - \lambda_t W_t &= 0 \end{aligned}$$

where λ_t is the usual Lagrangian multiplier. If we combine the first-order conditions for B_t ,

C_t , and C_{t+1} we get

$$P_t^{-1}C_t^{-1/\eta} = \mathbb{E}_t \left[\beta(1+i_t)P_{t+1}^{-1}C_{t+1}^{-1/\eta} \right]$$

Market clearing requires that $Y_t = C_t$, and for notational simplicity define $R_t = 1 + i_t$,

$$1 = \mathbb{E}_t \left[\beta R_t \left(\frac{P_t}{P_{t+1}} \right) \left(\frac{Y_t}{Y_{t+1}} \right)^{\frac{1}{\eta}} \right]$$

Log-linearising the above equation yields,

$$1 = \mathbb{E}_t \left[\beta \left(\frac{P^*}{P^*} \right) \left(\frac{Y^*}{Y^*} \right)^{\frac{1}{\eta}} \exp(i_t + p_t - p_{t+1} + \frac{1}{\eta}y_t - \frac{1}{\eta}y_{t+1}) \right]$$

where we note that $\ln(1 + i_t) \approx i_t$ when i_t is small. A first-order approximation is

$$1 = \mathbb{E}_t \left[\beta R^*(1 + i_t + p_t - p_{t+1} + \frac{1}{\eta}y_t - \frac{1}{\eta}y_{t+1}) \right]$$

Note that, in the steady-state $R^* = \frac{1}{\beta}$, so we can re-write as follows

$$1 - 1 = \mathbb{E}_t \left(i_t + p_t - p_{t+1} + \frac{1}{\eta}y_t - \frac{1}{\eta}y_{t+1} \right)$$

or

$$y_t = \mathbb{E}_t y_{t+1} - \eta i_t - \eta p_t + \eta \mathbb{E}_t p_{t+1}$$

Define inflation as $\mathbb{E}_t \pi_{t+1} := \mathbb{E}_t(p_{t+1} - p_t)$,

$$y_t = \mathbb{E}_t y_{t+1} - \eta(i_t - \mathbb{E}_t \pi_{t+1}) \tag{123}$$

Equation (123) is commonly referred to as a forward looking IS equation, where output today depends negatively on the real interest rate.

9.2.3. Marginal Cost and Returns to Scale

Before proceeding to look at the problem faced by the firm, we will briefly explore the relationship between marginal cost and returns to scale. The reason why we're doing this is to avoid a lengthy detour later when we replace a firm-specific marginal cost with an average

marginal cost for the economy. Define the production function of the economy to be

$$Y_t := A_t N_t^{1-\alpha}, \quad (124)$$

where A_t is technology, driven by a stationary AR(1) process, $a_t = \ln(A_t)$

$$a_t = \rho_a a_{t-1} + \varepsilon_{a,t}, \quad (125)$$

$\varepsilon_{a,t} \sim \mathcal{N}(0, \sigma_a^2)$. The marginal productivity of labour is then

$$\frac{\partial Y_t}{\partial N_t} = A_t (1 - \alpha) N_t^{-\alpha} \quad (126)$$

or, in log-linear form: $a_t + \ln(1 - \alpha) - \alpha n_t$. From textbook microeconomics, we can show that marginal cost is equal to the wage rate W divided by the marginal product of labour.¹⁰ Let Υ_t denote marginal cost, and we have

$$\Upsilon_t = W_t \frac{1}{A_t (1 - \alpha) N_t^{-\alpha}}$$

which in log-linear form is

$$\begin{aligned} \ln \Upsilon_t &= w_t - (a_t + \ln(1 - \alpha) - \alpha n_t) \\ &= w_t - \ln(1 - \alpha) - (a_t - \alpha n_t) \\ &= w_t - \ln(1 - \alpha) - \frac{1}{1 - \alpha} (a_t - \alpha y_t) \end{aligned} \quad (127)$$

where we've used a log-linear form of the production function, $y_t = a_t + (1 - \alpha)n_t$. As we're used to dealing with (log) deviations from steady-state values, let $\mu_t = \ln \Upsilon_t - \ln \Upsilon^*$, i.e., drop the constant $\ln(1 - \alpha)$. We can iterate (127) forward to period $t + k$ for both the flexible price case and the firm which last reset prices at time t , which we denote by $\mu_{t+k|t}$, to see

$$\begin{aligned} \mu_{t+k} &= w_{t+k} - \frac{1}{1 - \alpha} (a_{t+k} - \alpha y_{t+k}) \\ \mu_{t+k|t} &= w_{t+k} - \frac{1}{1 - \alpha} (a_{t+k} - \alpha y_{t+k|t}) \end{aligned}$$

¹⁰For those unfamiliar with this, note that total cost $TC = WN$ (there's no capital in this model). Marginal cost is $\frac{\partial TC}{\partial Y} = \frac{\partial WN}{\partial Y} = W \frac{\partial N}{\partial Y}$. Notice that, by the inverse function theorem, $\frac{\partial N}{\partial Y}$ is the inverse of the marginal product of labour, $\frac{\partial Y}{\partial N}$, thus $\Upsilon = \frac{W}{MPN}$.

respectively. Taking the latter away from the former yields

$$\mu_{t+k|t} = \mu_{t+k} + \frac{\alpha}{1-\alpha}(y_{t+k|t} - y_{t+k}) \quad (128)$$

If $\alpha = 0$, we have $\mu_{t+k|t} = \mu_{t+k}$. Keep (128) in mind for later.

9.2.4. The Firm

With the demand function $Y_t(k) = \left(\frac{P_t(k)}{P_t}\right)^{-\vartheta} Y_t$ in mind, we move on to the problem of the firm, where the pricing mechanism in the economy is based on Calvo (1983). In this economy, $(1-\vartheta)$ of firms are able to reset prices in a given period (for example, time t), which they set equal to X_t , and the proportion of firms who cannot reset their prices at time t set prices equal to last period's price, P_{t-1} .

The firm which last reset prices at time t wants to maximise profit

$$\max_{X_t} \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k|t}(j)X_t - C(Y_{t+k|t}(j))) \right], \quad (129)$$

where $C(\cdot)$ is the nominal cost function, subject to the $t+k$ demand function

$$Y_{t+k|t}(j) = \left(\frac{X_t}{P_{t+k}}\right)^{-\vartheta} Y_{t+k} \quad (130)$$

After substituting for the demand functions, the maximization problem is given by

$$\max_{X_t} \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^{\vartheta} X_t^{1-\vartheta} - C(Y_{t+k} P_{t+k}^{\vartheta} X_t^{-\vartheta})) \right] \quad (131)$$

The first-order condition with respect to X_t is given by

$$\begin{aligned} \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^{\vartheta} (1-\vartheta) X_t^{-\vartheta} - C'(Y_{t+k} P_{t+k}^{\vartheta} X_t^{-\vartheta}) (-\vartheta Y_{t+k} P_{t+k}^{\vartheta} X_t^{-\vartheta-1})) \right] &= 0 \quad (132) \\ \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^{\vartheta} (1-\vartheta) X_t^{-\vartheta}) + \sum_{k=0}^{\infty} (\theta\beta)^k (\Upsilon_{t+k|t} \vartheta Y_{t+k} P_{t+k}^{\vartheta} X_t^{-\vartheta-1}) \right] &= 0 \\ \mathbb{E}_t \left[(1-\vartheta) X_t^{-\vartheta} \sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^{\vartheta}) + (\vartheta X_t^{-\vartheta-1}) \sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^{\vartheta} \Upsilon_{t+k|t}) \right] &= 0 \end{aligned}$$

We can this re-write as

$$\begin{aligned} \frac{(\vartheta - 1)X_t^{-\vartheta}}{X_t^{-\vartheta-1}} \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^\vartheta) \right] &= \vartheta \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^\vartheta \Upsilon_{t+k|t}) \right] \\ X_t \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^\vartheta) \right] &= \frac{\vartheta}{(\vartheta - 1)} \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^\vartheta \Upsilon_{t+k|t}) \right] \end{aligned}$$

implying

$$X_t = \frac{\vartheta}{\vartheta - 1} \frac{\mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^\vartheta \Upsilon_{t+k|t}) \right]}{\mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y_{t+k} P_{t+k}^\vartheta) \right]}. \quad (133)$$

In the steady-state, where $X_t = X_{t-1} = X^*$,

$$\begin{aligned} X^* &= \frac{\vartheta}{\vartheta - 1} \frac{\mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y^* (P^*)^\vartheta \Upsilon^*) \right]}{\mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (Y^* (P^*)^\vartheta) \right]} \\ &= \frac{\vartheta}{\vartheta - 1} \Upsilon^* \end{aligned}$$

This is where we can clearly see the interpretation of ϑ as a markup over marginal cost; if, for example, $\vartheta = 10$, $X^* = 1.111 \cdot \Upsilon^*$ in steady-state. If we log-linearize (132), the left- and right-hand sides are

$$(1 - \vartheta) Y_{t+k} P_{t+k}^\vartheta X_t^{-\vartheta} \approx (1 - \vartheta) Y^* (P^*)^\vartheta (X^*)^{-\vartheta} (1 + y_{t+k} + \vartheta p_{t+k} - \vartheta x_t) \quad (134)$$

$$\begin{aligned} \vartheta \Upsilon_{t+k|t} Y_{t+k} P_{t+k}^\vartheta X_t^{-\vartheta-1} &\approx \vartheta \Upsilon^* Y^* (P^*)^\vartheta (X^*)^{-\vartheta-1} (1 + \mu_{t+k|t} + y_{t+k} + \vartheta p_{t+k} - (\vartheta + 1)x_t) \\ &\quad (135) \end{aligned}$$

respectively. Replacing Υ^* in (135)

$$\vartheta \frac{\vartheta - 1}{\vartheta} X^* Y^* (P^*)^\vartheta (X^*)^{-\vartheta-1} (1 + \mu_{t+k|t} + y_{t+k} + \vartheta p_{t+k} - (\vartheta + 1)x_t)$$

Therefore, a first-order approximation of (132) is

$$\begin{aligned} \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k ((1 - \vartheta) Y^* (P^*)^\vartheta (X^*)^{-\vartheta} (1 + y_{t+k} + \vartheta p_{t+k} - \vartheta x_t) \right. \\ \left. - (1 - \vartheta) Y^* (P^*)^\vartheta (X^*)^{-\vartheta} (1 + \mu_{t+k|t} + y_{t+k} + \vartheta p_{t+k} - (\vartheta + 1)x_t)) \right] = 0 \end{aligned}$$

which can be reduced to

$$\mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (x_t - \mu_{t+k|t}) \right] = 0$$

Now, recall our relationship between $\mu_{t+k|t}$ and μ_{t+k} in equation (128): if we log-linearize the demand function

$$Y_{t+k|t}(j) = \left(\frac{X_t}{P_{t+k}} \right)^{-\vartheta} Y_{t+k}$$

we see that $y_{t+k|t} - y_{t+k} = -\vartheta(x_t - p_{t+k})$, and combining this with (128) we get

$$\mu_{t+k|t} = \mu_{t+k} - \frac{\alpha\vartheta}{1-\alpha}(x_t - p_{t+k}) \quad (136)$$

We can plug this into $\mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (x_t - \mu_{t+k|t}) \right] = 0$,

$$\begin{aligned} 0 &= \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k (x_t - \mu_{t+k} + \frac{\alpha\vartheta}{1-\alpha}(x_t - p_{t+k})) \right] \\ 0 &= \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k \frac{1+\alpha(\vartheta-1)}{1-\alpha} x_t - \sum_{k=0}^{\infty} (\theta\beta)^k \left(\frac{(1-\alpha)\mu_{t+k} + \alpha\vartheta p_{t+k}}{1-\alpha} \right) \right] \\ 0 &= \mathbb{E}_t \left[\frac{1+\alpha(\vartheta-1)}{1-\theta\beta} x_t - \sum_{k=0}^{\infty} (\theta\beta)^k ((1-\alpha)\mu_{t+k} + \alpha\vartheta p_{t+k}) \right] \end{aligned}$$

Rearranging

$$\begin{aligned} \frac{1+\alpha(\vartheta-1)}{1-\theta\beta} x_t &= \mathbb{E}_t \left[\sum_{k=0}^{\infty} (\theta\beta)^k ((1-\alpha)\mu_{t+k} + \alpha\vartheta p_{t+k}) \right] \\ \Rightarrow x_t &= \frac{1-\theta\beta}{1+\alpha(\vartheta-1)} \sum_{k=0}^{\infty} (\theta\beta)^k \mathbb{E}_t ((1-\alpha)\mu_{t+k} + \alpha\vartheta p_{t+k}) \end{aligned}$$

We can save ourselves some algebraic trouble later by using real marginal cost, $\mu_t^r = \mu_t - p_t$, instead of nominal marginal cost. The new form is

$$\begin{aligned} x_t &= \frac{1-\theta\beta}{1+\alpha(\vartheta-1)} \sum_{k=0}^{\infty} (\theta\beta)^k \mathbb{E}_t ((1-\alpha)(\mu_{t+k}^r + p_{t+k}) + \alpha\vartheta p_{t+k}) \\ &= \frac{1-\theta\beta}{1+\alpha(\vartheta-1)} \sum_{k=0}^{\infty} (\theta\beta)^k \mathbb{E}_t ((1-\alpha)\mu_{t+k}^r + (1+\alpha(\vartheta-1))p_{t+k}) \end{aligned}$$

Notice that we have $1+\alpha(\vartheta-1)$ next to p_{t+k} , which is also the denominator of $\frac{1-\theta\beta}{1+\alpha(\vartheta-1)}$.

Define $\Theta = \frac{1-\alpha}{1+\alpha(\vartheta-1)}$, and we now have the simpler expression

$$x_t = (1 - \theta\beta) \sum_{k=0}^{\infty} (\theta\beta)^k \mathbb{E}_t (\Theta \mu_{t+k}^r + p_{t+k})$$

From equation (118), the (aggregate) price level,

$$\begin{aligned} P_t &= \left(\int_0^1 P_t(i)^{1-\vartheta} di \right)^{\frac{1}{1-\vartheta}} \\ P_t^{1-\vartheta} &= \int_0^1 P_t(i)^{1-\vartheta} di, \end{aligned}$$

can be written as:

$$\begin{aligned} P_t^{1-\vartheta} &= (1 - \theta) X_t^{1-\vartheta} + \theta \int_0^\theta P_{t-1}(i)^{1-\vartheta} di \\ &= [(1 - \theta) X_t^{1-\vartheta} + \theta P_{t-1}^{1-\vartheta}] \end{aligned}$$

If we assume a zero inflation steady-state, $P_t = X_t = P_{t-1} = P^*$, we can log-linearize the above equation as follows:

$$(P^*)^{1-\vartheta} (1 + (1 - \vartheta)p_t) = [(1 - \theta)(P^*)^{1-\vartheta} (1 + (1 - \vartheta)x_t) + \theta(P^*)^{1-\vartheta} (1 + (1 - \vartheta)p_{t-1})]$$

which simplifies to

$$p_t = (1 - \theta)x_t + \theta p_{t-1} \quad (137)$$

Note that

$$x_t = (1 - \theta\beta) \sum_{k=0}^{\infty} (\theta\beta)^k \mathbb{E}_t (\Theta \mu_{t+k}^r + p_{t+k})$$

can be written in a format such as

$$x_t = (1 - \theta\beta)(\Theta \mu_t^r + p_t) + \theta\beta \mathbb{E}_t x_{t+1} \quad (138)$$

as the equation with the infinite sum is the standard solution to the first-order stochastic

difference equation. Rewrite (137) as

$$x_t = \frac{1}{(1-\theta)}(p_t - \theta p_{t-1})$$

and substitute into (138) to get

$$\begin{aligned} x_t &= (1-\theta\beta)(\Theta\mu_t^r + p_t) + (\theta\beta)\mathbb{E}_t x_{t+1} \\ \Rightarrow \frac{1}{(1-\theta)}(p_t - \theta p_{t-1}) &= (1-\theta\beta)(\Theta\mu_t^r + p_t) + (\theta\beta)\mathbb{E}_t \left(\frac{1}{(1-\theta)}(p_{t+1} - \theta p_t) \right) \\ p_t - \theta p_{t-1} &= (1-\theta)(1-\theta\beta)(\Theta\mu_t^r + p_t) + (\theta\beta)\mathbb{E}_t(p_{t+1} - \theta p_t) \\ p_t - \theta p_{t-1} &= (1-\theta)(1-\theta\beta)(\Theta\mu_t^r + p_t) + (\theta\beta)\mathbb{E}_t(p_{t+1}) - (\theta\beta)\theta p_t \\ p_t(1 + \theta^2\beta) - \theta p_{t-1} &= (1-\theta)(1-\theta\beta)(\Theta\mu_t^r + p_t) + \theta\beta\mathbb{E}_t(p_{t+1}) \end{aligned} \quad (139)$$

Divide both sides by θ and subtract βp_t from both sides,

$$\begin{aligned} p_t \frac{(1 + \theta^2\beta)}{\theta} - p_{t-1} - \beta p_t &= \frac{(1-\theta)(1-\theta\beta)}{\theta}(\Theta\mu_t^r + p_t) + \beta\mathbb{E}_t(p_{t+1}) - \beta p_t \\ p_t \frac{(1 - \theta\beta + \theta^2\beta)}{\theta} - p_{t-1} &= \frac{(1-\theta)(1-\theta\beta)}{\theta}(\Theta\mu_t^r + p_t) + \beta\mathbb{E}_t(p_{t+1} - p_t) \\ p_t \frac{(1 - \theta\beta + \theta^2\beta)}{\theta} - p_{t-1} &= \frac{(1-\theta)(1-\theta\beta)}{\theta}(\Theta\mu_t^r + p_t) + \beta\mathbb{E}_t\pi_{t+1} \end{aligned} \quad (140)$$

Note that,

$$\frac{(1 - \theta\beta + \theta^2\beta)}{\theta} = 1 + \frac{(1-\theta)(1-\theta\beta)}{\theta}$$

so

$$\begin{aligned} p_t \left(1 + \frac{(1-\theta)(1-\theta\beta)}{\theta} \right) - p_{t-1} &= \frac{(1-\theta)(1-\theta\beta)}{\theta}(\Theta\mu_t^r + p_t) + \beta\mathbb{E}_t\pi_{t+1} \\ \pi_t &= \beta\mathbb{E}_t\pi_{t+1} + \frac{(1-\theta)(1-\theta\beta)}{\theta}(\Theta\mu_t^r + p_t - p_t) \end{aligned} \quad (141)$$

Or

$$\pi_t = \beta\mathbb{E}_t\pi_{t+1} + \frac{(1-\theta)(1-\theta\beta)}{\theta}\Theta\mu_t^r \quad (142)$$

Equation (142) is known as the New-Keynesian Phillips Curve (NKPC).

As we've assumed diminishing returns to scale in the production function, *i.e.*, that higher output reduces marginal productivity and raises marginal cost, this implies that real marginal

cost is a function of the output gap

$$y_t^g = y_t - y_t^n \quad (143)$$

where y_t^n is the path of output that would have obtained in a zero inflation, frictionless economy. Using the first-order conditions with respect to C_t and N_t from the maximization problem in (121) (subject to (122)), we see

$$P_t^{-1} C_t^{-1/\eta} = N_t^\phi W_t^{-1},$$

which in log-linear form is

$$\frac{1}{\eta} c_t + \phi n_t = w_t - p_t.$$

Recall that marginal cost is

$$\begin{aligned} \mu_t^r + p_t &= w_t - \frac{1}{1-\alpha}(a_t - \alpha y_t) \\ \mu_t^r &= (w_t - p_t) + \frac{1}{1-\alpha}(\alpha y_t - a_t) \\ &= \left(\frac{1}{\eta} c_t + \phi n_t \right) + \frac{1}{1-\alpha}(\alpha y_t - a_t) \end{aligned}$$

Now, using the fact that $y_t = c_t$ and $n_t = \frac{y_t - a_t}{1-\alpha}$ (from the market clearing condition and the production function, respectively), we can proceed as

$$\begin{aligned} \mu_t^r &= \frac{1}{\eta} y_t + \phi \frac{y_t - a_t}{1-\alpha} + \frac{\alpha y_t - a_t}{1-\alpha} \\ &= \left(\frac{1}{\eta} + \frac{\phi + \alpha}{1-\alpha} \right) y_t - \frac{1+\phi}{1-\alpha} a_t \end{aligned}$$

Under flexible prices $\mu_t^r = 0$, i.e., the log-deviation of real marginal cost from its steady-

state value should be zero, thus

$$\begin{aligned}
0 &= \left(\frac{1}{\eta} + \frac{\phi + \alpha}{1 - \alpha} \right) y_t^n - \frac{1 + \phi}{1 - \alpha} a_t \\
y_t^n &= \left(\frac{1}{\eta} + \frac{\phi + \alpha}{1 - \alpha} \right)^{-1} \frac{1 + \phi}{1 - \alpha} a_t \\
&= \left(\frac{1 - \alpha}{\eta(1 - \alpha)} + \frac{\eta(\phi + \alpha)}{\eta(1 - \alpha)} \right)^{-1} \frac{1 + \phi}{1 - \alpha} a_t \\
&= \left(\frac{\eta(1 - \alpha)}{1 - \alpha + \eta(\phi + \alpha)} \right) \frac{1 + \phi}{1 - \alpha} a_t \\
\Rightarrow y_t^n &= \left(\frac{\eta(1 + \phi)}{1 - \alpha + \eta(\phi + \alpha)} \right) a_t
\end{aligned}$$

Therefore, we can write real marginal cost as

$$\mu_t^r = \left(\frac{1}{\eta} + \frac{\phi + \alpha}{1 - \alpha} \right) (y_t - y_t^n)$$

and the final form of the NKPC is

$$\pi_t = \beta \mathbb{E}_t \pi_{t+1} + \kappa y_t^g \quad (144)$$

where

$$\kappa = \frac{(1 - \theta)(1 - \theta\beta)}{\theta} \Theta \left(\frac{1}{\eta} + \frac{\phi + \alpha}{1 - \alpha} \right) \quad (145)$$

One important point to note is that a lot of this additional work is based on some value of $\alpha \in (0, 1)$. If $\alpha = 0$ then

$$\kappa = \frac{(1 - \theta)(1 - \theta\beta)(\eta^{-1} + \phi)}{\theta}$$

which is the standard definition for models without diminishing returns to scale.

Going back to the forward-looking IS equation, we can include the output gap to give

$$y_t^g = \mathbb{E}_t y_{t+1}^g - \eta(i_t - \mathbb{E}_t \pi_{t+1}) + \mathbb{E}_t y_{t+1}^n - y_t^n$$

Or, more naturally, as

$$y_t^g = \mathbb{E}_t y_{t+1}^g - \eta(i_t - \mathbb{E}_t \pi_{t+1} - r_t^n) \quad (146)$$

where

$$r_t^n = \eta^{-1} \mathbb{E}_t \Delta y_{t+1}^n \quad (147)$$

and Δ is the first-difference operator. This equation defines a “natural” real interest rate, r_t^n (consistent with $y_t^g = \mathbb{E}_t y_{t+1}^g$) determined by technology and preferences.

To determine the nominal interest rate, we can use a monetary policy reaction function

$$i_t = \phi_\pi \pi_t + \phi_y y_t^g + v_t \quad (148)$$

a version of the so-called ‘Taylor rule’. The ‘shock’ in this equation, v_t , can be interpreted as a change to the non-systematic component of monetary policy. Equations (144), (146), and (148) form the three equation New-Keynesian model.

Before we finish, it’s useful to give an alternative form for the natural rate,

$$r_t^n = \eta^{-1} \mathbb{E}_t \Delta y_{t+1}^n$$

We’ve already seen that

$$y_t^n = \frac{\eta(1+\phi)}{1-\alpha+\eta(\phi+\alpha)} a_t,$$

r_t^n then becomes

$$\begin{aligned} r_t^n &= \frac{1+\phi}{1-\alpha+\eta(\phi+\alpha)} \mathbb{E}_t \Delta a_{t+1} \\ \Rightarrow r_t^n &= \frac{1+\phi}{1-\alpha+\eta(\phi+\alpha)} (\rho_a - 1) a_t \end{aligned} \quad (149)$$

as $\mathbb{E}_t a_{t+1} = \rho a_t$.

9.2.5. Inputting the Model to BMR

The full model is a system of 8 equations, six endogenous processes

$$y_t^g = \mathbb{E}_t y_{t+1}^g - \eta(i_t - \mathbb{E}_t \pi_{t+1} - r_t^n) \quad (150)$$

$$\pi_t = \beta \mathbb{E}_t \pi_{t+1} + \kappa y_t^g \quad (151)$$

$$i_t = \phi_\pi \pi_t + \phi_y y_t^g + \nu_t \quad (152)$$

$$r_t^n = \frac{1 + \phi}{1 - \alpha + \eta(\phi + \alpha)} (\rho_a - 1) a_t \quad (153)$$

$$y_t = a_t + (1 - \alpha)n_t \quad (154)$$

$$y_t^g = y_t - \frac{\eta(1 + \phi)}{1 - \alpha + \eta(\phi + \alpha)} a_t \quad (155)$$

with two exogenous shocks, technology and monetary policy,

$$a_t = \rho_a a_{t-1} + \varepsilon_{a,t} \quad (156)$$

$$\nu_t = \rho_\nu \nu_{t-1} + \varepsilon_{\nu,t} \quad (157)$$

respectively.

Note that there are no lags in any of the above equations, except for the shocks, so we have no natural candidates for control variables. With a problem like this, we can set up the matrices to use the ‘brute-force’ method

$$\begin{aligned} 0 &= \mathbb{E}_t \{ F \zeta_{t+1} + G \zeta_t + H \zeta_{t-1} + L z_{t+1} + M z_t \} \\ z_t &= N z_{t-1} + \varepsilon_t \end{aligned}$$

with solution

$$\zeta_t = P \zeta_{t-1} + Q z_t$$

$$z_t = N z_{t-1} + \varepsilon_t$$

and final form being

$$\xi_t = \mathcal{F} \xi_{t-1} + \mathcal{G} \varepsilon_t.$$

In order, the matrices of variables are

$$\zeta_t = [y_t^g \ y_t \ \pi_t \ r_t^n \ i_t \ n]^\top$$

$$z_t = [a_t \ v_t]^\top$$

The matrices of deep parameters are as follows. For the 6 main equations

$$A = B = C = D = J = K = []_{0 \times 0}$$

$$F = \begin{bmatrix} -1 & 0 & -\eta/4 & 0 & 0 & 0 \\ 0 & 0 & -\beta/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & -\eta & \eta/4 & 0 \\ -\kappa & 0 & 0.25 & 0 & 0 & 0 \\ -\phi_y & 0 & -\phi_\pi/4 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -(1-\alpha) \\ 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$H = [\mathbf{0}_{6 \times 6}]$$

For the 2 shocks

$$L = [\mathbf{0}_{6 \times 2}]$$

$$M = \begin{bmatrix} 0 & 0 & 0 & -\psi(\rho_a - 1)/\eta & -1 & \psi \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}^\top$$

$$N = \begin{bmatrix} \rho_a & 0 \\ 0 & \rho_v \end{bmatrix}$$

$$\text{where } \psi = \frac{\eta(1+\phi)}{1-\alpha+\eta(\phi+\alpha)}.$$

The reader may have noticed that some series were divided by 4 in the F and G matrices. The reason for this is to replicate the IRFs shown in (Galí, 2008, Ch. 3), as Galí annualises inflation and interest rates. Further to this effort, let's calibrate the parameters to be: $\eta = 1$, $\alpha = 1/3$, $\beta = 0.99$, $\theta = 2/3$, $\phi = 1$, $\phi_\pi = 1.5$, $\phi_y = 0.125$, $\rho_a = 0.9$, $\rho_v = 0.5$. The solution is then

$$P = [\mathbf{0}_{6 \times 6}]$$

$$Q = \begin{bmatrix} -0.107402 & 0.8925972 & -0.50556250 & -0.1 & -0.8120451 & -0.1603027 \\ -1.137656 & -1.137656 & -1.155860 & 0 & 1.697382 & -1.69799 \end{bmatrix}^\top$$

The impulse response functions to technology and monetary policy shocks, of order $\sigma_a = 1$ and $\sigma_v = 0.25$, are shown in figures 30 and 31, respectively.

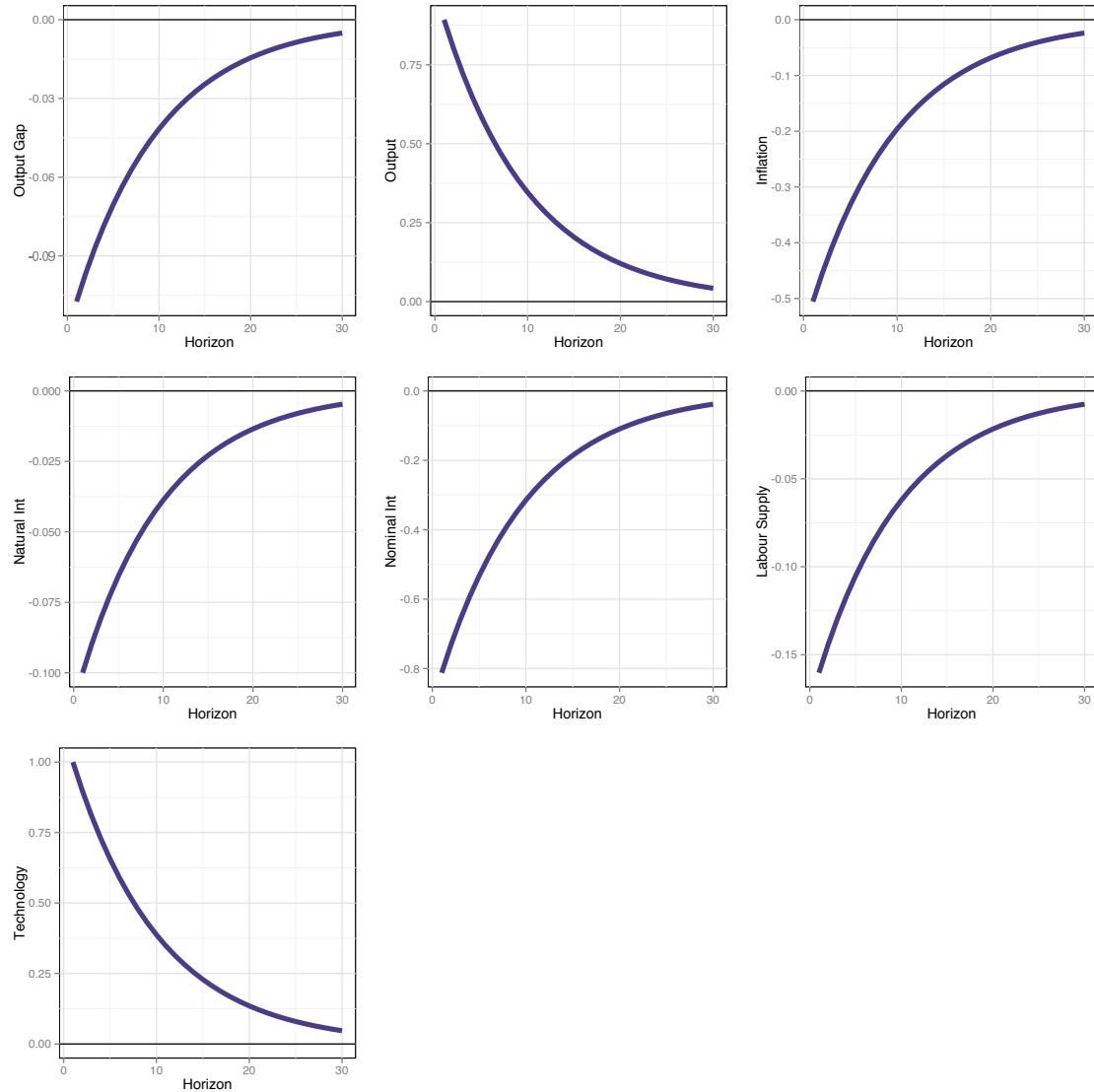


Figure 30: Shock to Technology.

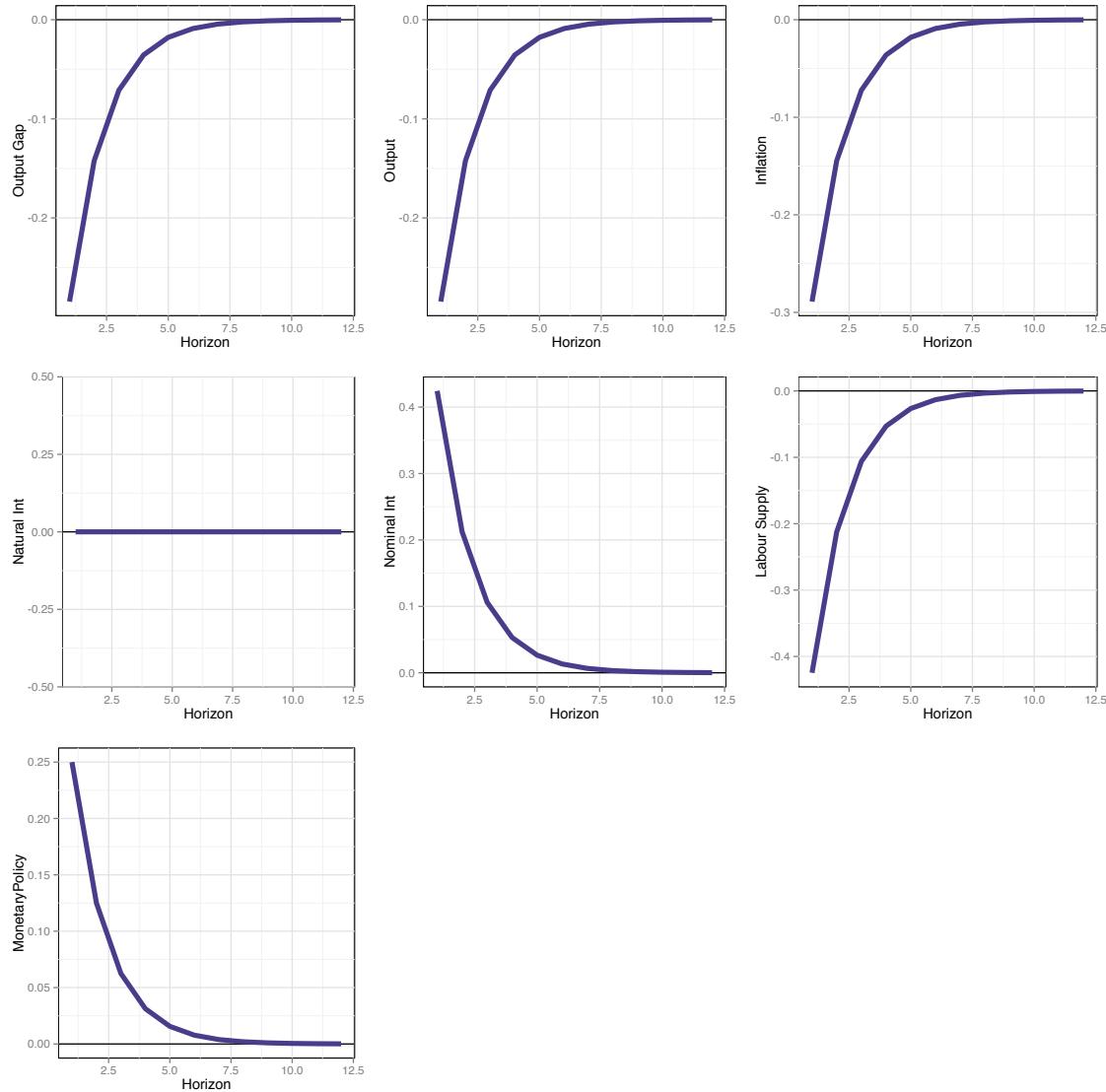


Figure 31: Shock to Monetary Policy.

9.2.6. Estimation

Estimating an RBC model with model-generated data is fine for illustrative purposes. In this section, we will use the updated Stock and Watson dataset (from the monetary policy VAR example) to estimate the parameters of the basic New-Keynesian model. To avoid a situation of stochastic singularity, we cannot have more observable series than ‘shocks,’ thus we restrict our attention to inflation and the Federal Funds rate. As we’re dealing with mean-zero series in the model, the reader can demean the series first.

Begin by defining the observable matrix \mathcal{H} , and recall that our series are ordered according to $\zeta_t = [y_t^g \ y_t \ \pi_t \ r_t^n \ i_t \ n]^\top$, with a technology shock and monetary policy shock at the end. Thus,

```
ObserveMat <- cbind(c(0,0,4,0,0,0,0,0),
                      c(0,0,0,0,4,0,0,0))
```

Define the parameter names in their respective order

```
parnames <- c("Alpha", "Beta", "Vartheta", "Theta", "Eta", "Phi",
              "Phi.Pi", "Phi.Y", "RhoA", "RhoV", "SigmaT", "SigmaM")
```

Our initial values are similar to the values we used for our calibration exercise,

```
initialvals <- c(.33, 0.97, 6, 0.6667, 1, 1, 1.5, 0.5/4, 0.90, 0.5, 1, 0.25)
```

Assign prior densities to each of the parameters

```
priorform <- c("Normal", "Beta", "Normal", "Beta", "Normal", "Normal",
               "Normal", "Normal", "Beta", "Beta", "IGamma", "IGamma")
```

and the relevant parameters of these densities,

```
priorpars <- cbind(c( 0.33, 20, 5, 3, 1, 1, 1.5, 0.7, 3, 4, 2, 2),
                      c(0.05^2, 2, 1, 2, 0.1^2, 0.3^2, 0.1^2, 0.1^2, 2, 1.5, 2, 1))
```

with appropriate upper- and lower-bounds

```
parbounds <- cbind(c(NA, 0.95, NA, 0.01, NA, NA, NA, 0.01, 0.01, NA, NA),
                     c(NA, 0.99999, NA, 0.999, NA, NA, NA, 0.999, 0.999, NA, NA))
```

To estimate the posterior mode, we first use a simplex method, then a conjugate gradient method. We will run 2 chains of 125,000 draws each, 75,000 of which will be discarded as burn-in, with the scaling parameter set to 0.27. Rather than running the chains sequentially, we set the number of CPU cores to the number of chains, so that the chains will run in parallel.
(Make sure you don't set more cores than your computer can handle!)

```
NKMest <- EDSGE(dsgeadata,chains=2,cores=2,
  ObserveMat,initialvals,partomats,
  priorform,priorpars,parbounds,parnames,
  optimMethod=c("Nelder-Mead","CG"),
  optimLower=NULL,optimUpper=NULL,
  optimControl=list(maxit=10000),
  DSGEIRFs=TRUE,irf.periods=40,
  scalepar=0.27,keep=50000,burnin=75000)
```

Trying to solve the model with your initial values... Done.

Beginning optimization, Tue Jul 8 11:33:38 2014.

Using Optimization Method: Nelder-Mead.

Using Optimization Method: CG. Change in the log posterior: 1.69529.

Optimization over, Tue Jul 8 11:34:05 2014.

Optimizer Convergence Code: 0; successful completion.

Optimizer Iterations:

```
function gradient
  1908      698
```

Log Marginal Likelihood: -554.2439.

Parameter Estimates and Standard Errors (SE) at the Posterior Mode:

	Estimate	SE
Alpha	0.3321654	0.049859677

Beta	0.9955405	0.004056926
Vartheta	5.0705516	0.992600544
Theta	0.7766846	0.042850061
Eta	0.9864517	0.099302813
Phi	0.9771752	0.300882397
Phi.Pi	1.4865530	0.098942742
Phi.Y	0.6790696	0.099636942
Rho.A	0.9507074	0.019816684
Rho.V	0.7680469	0.042108072
Sigma.T	2.7408834	0.662286625
Sigma.M	1.1123389	0.162952324

Beginning MCMC run, Tue Jul 8 11:34:05 2014.

MCMC run finished, Tue Jul 8 11:37:05 2014.

Acceptance Rate: Chain 1: 0.36978; Chain 2: 0.37132.

Root-R Chain-Convergence Statistics:

Alpha	Beta	Vartheta	Theta	Eta	Phi	Phi.Pi	Phi.Y	Rho.A
Stat: 1.000001	1.000025	1.00016	0.9999959	1.000038	1.000004	1.000001	1.000025	1.00016
	Rho.V	Sigma.T	Sigma.M					
Stat: 0.9999959	1.000038	1.000004						

Parameter Estimates and Standard Errors:

	Posterior.Mode	SE.Mode	Posterior.Mean	SE.Posterior
Alpha	0.3321654	0.049859677	0.3329884	0.050109563
Beta	0.9955405	0.004056926	0.9955043	0.002924794
Vartheta	5.0705516	0.992600544	5.1012167	1.004772073
Theta	0.7766846	0.042850061	0.7776043	0.042314780
Eta	0.9864517	0.099302813	0.9833182	0.100543766
Phi	0.9771752	0.300882397	0.9743492	0.305949294

Phi.Pi	1.4865530	0.098942742	1.4880491	0.097892153
Phi.Y	0.6790696	0.099636942	0.6826578	0.099160984
Rho.A	0.9507074	0.019816684	0.9526904	0.016571630
Rho.V	0.7680469	0.042108072	0.7698100	0.042105670
Sigma.T	2.7408834	0.662286625	3.1518228	1.062444794
Sigma.M	1.1123389	0.162952324	1.1548837	0.205666807

Computing IRFs now... Done.

And we're finished with estimation. The reader can plot the results using:

```
plot(NKMest,save=TRUE)
IRF(NKMest, FALSE, varnames=c("Output Gap", "Output", "Inflation",
  "Natural Int", "Nominal Int", "Labour Supply", "Technology",
  "MonetaryPolicy"), save=TRUE)
```

the results of which are shown in figures 33, 34, and 35.

Finally, we can evaluate the optimization routine by plotting the value of the log posterior using the mode values and one standard deviation either side of them (scaled by c), $\pm c \cdot \sigma$, where σ is from the inverse-Hessian at the posterior mode, holding all other parameters fixed at their mode values. The relevant code is:

```
modecheck(NKMest,1000,1,plottransform=FALSE,save=TRUE)
```

and the result is illustrated in figure 32, with the green line being the value of the log posterior for different values of the parameters, the dashed line being the mode value for that parameter.

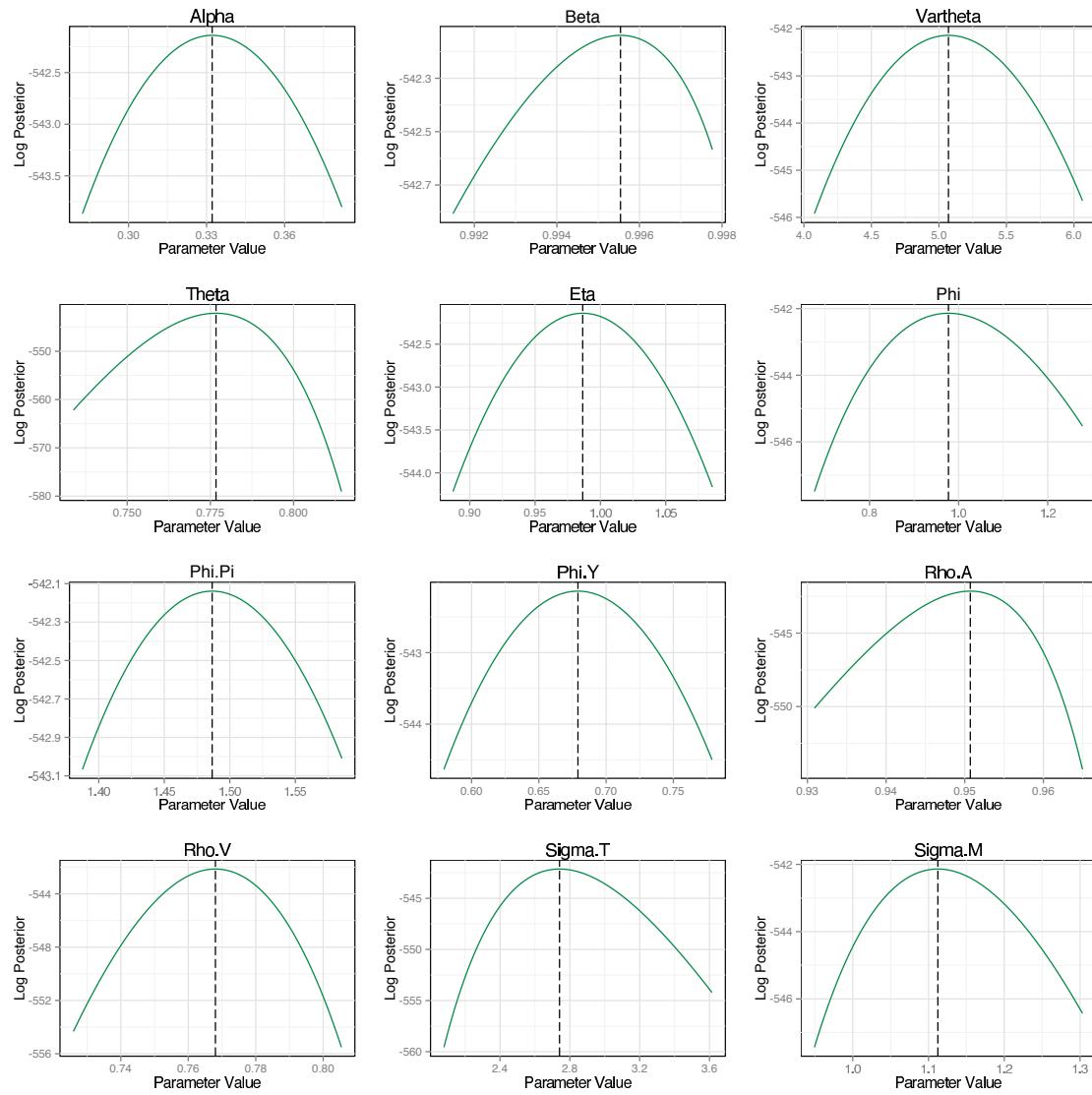
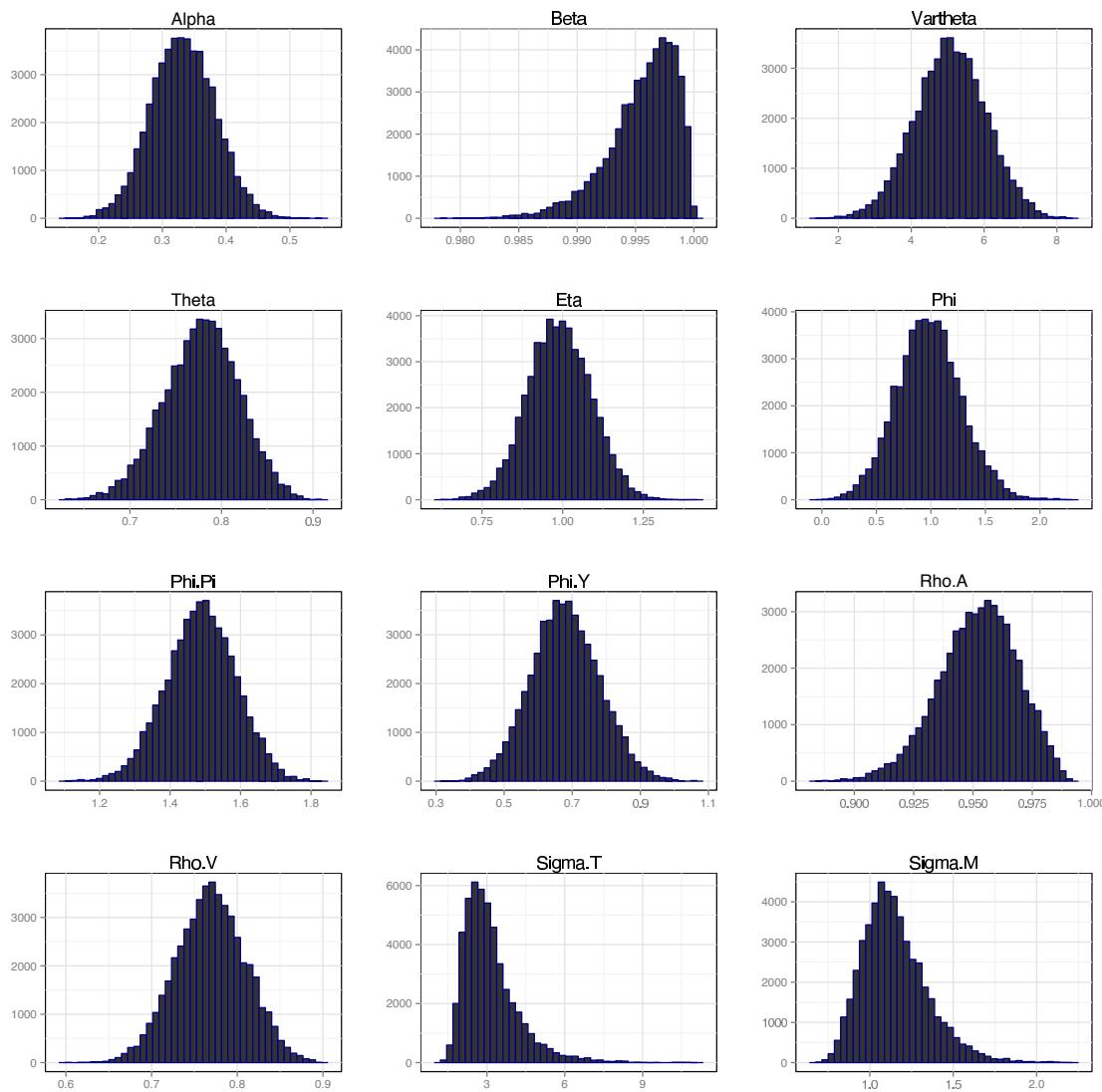


Figure 32: Plot of the Log-Posterior at the Posterior Mode.

Figure 33: Posterior distributions of θ .

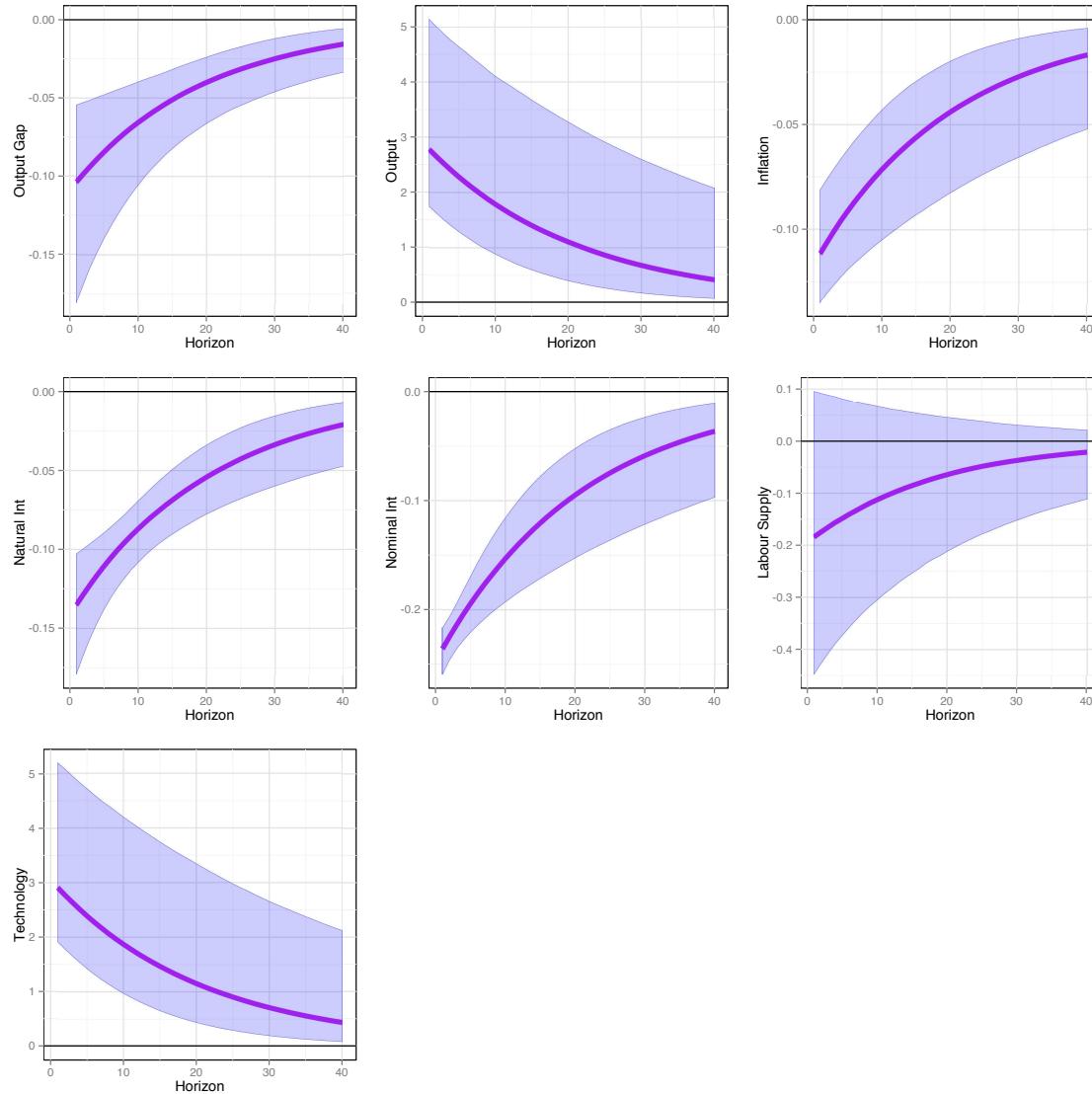


Figure 34: Shock to Technology.

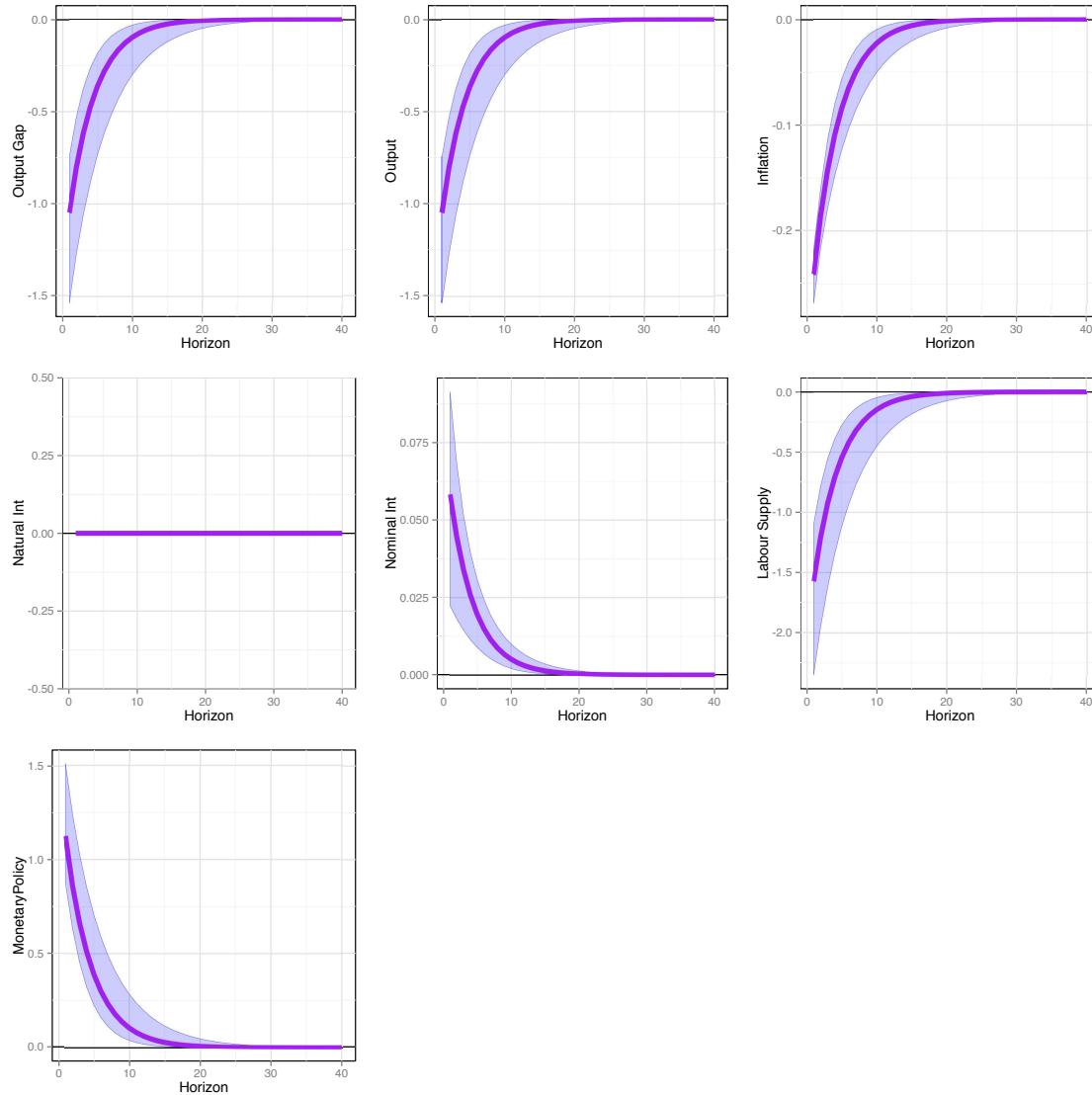


Figure 35: Shock to Monetary Policy.

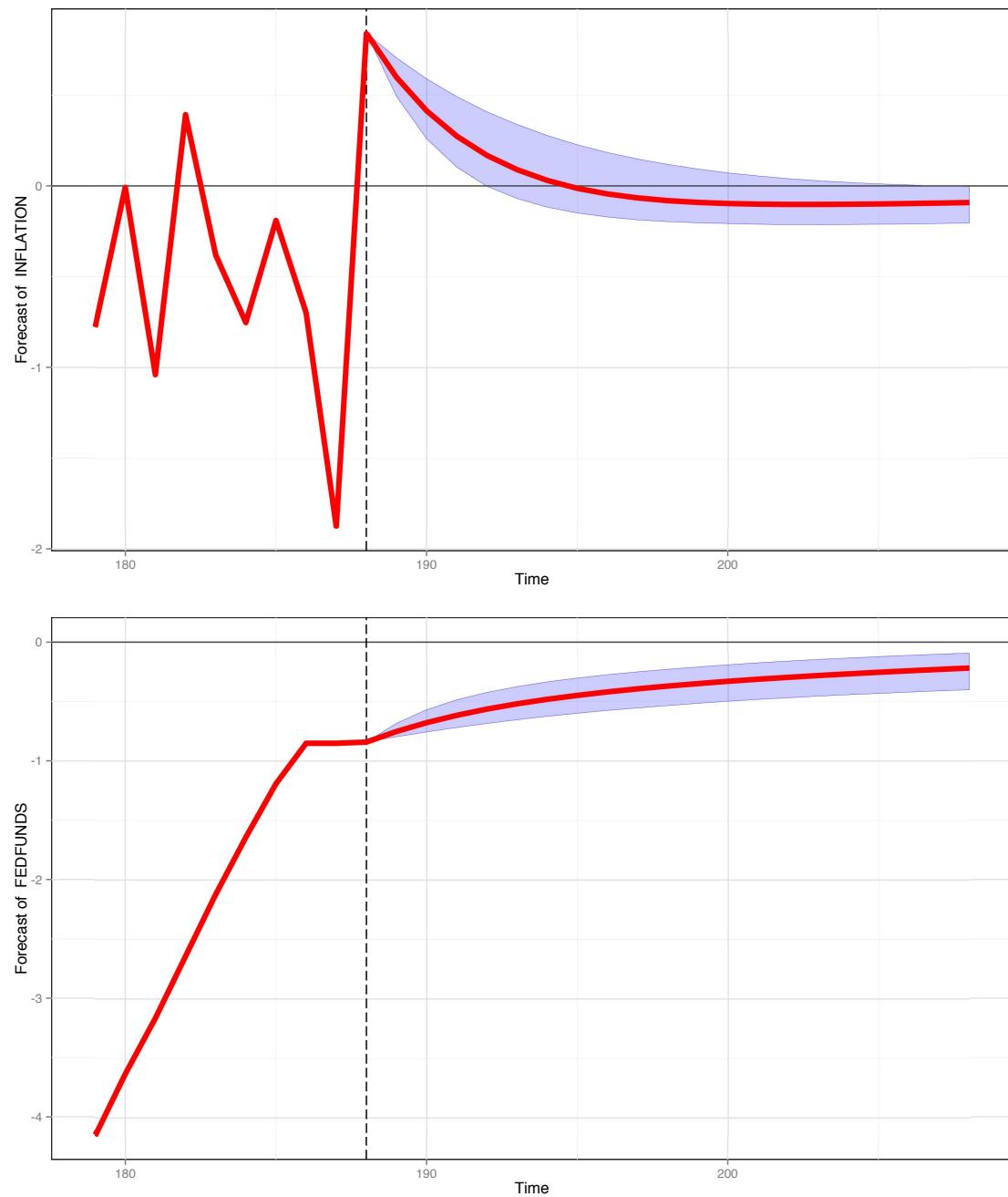


Figure 36: Forecast.

9.3. A Rather More Complicated Affair

The ‘baseline’ model discussed in Fernández-Villaverde and Rubio-Ramírez (2006), Fernández-Villaverde et al. (2009), and Fernández-Villaverde (2010) forms an excellent example of the foundations of a state of the art DSGE model, equipped with many of the rigidities found in operational-level medium-sized DSGE models currently being used in many macroeconomic policy making institutions. The model is similar to the well-known models of Christiano et al. (2005) and Smets and Wouters (2007), and, for those interested in working through something a little smaller, a somewhat more complicated model than that of Schorfheide (2011) by including habit formation in consumption and wage rigidities.

Households maximise lifetime utility according to

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t d_t \left\{ \ln(C_{j,t} - hC_{j,t-1}) + \nu \ln \left(\frac{M_{j,t}}{P_t} \right) - \varphi_t \psi \frac{(L_{j,t}^s)^{1+\vartheta}}{1+\vartheta} \right\} \quad (158)$$

where C_t is consumption, $h \in [0, 1]$ is a habit formation parameter, M_t is money holdings, L_t^s is labour, $1/\vartheta$ is the Frisch elasticity of labour supply, and d_t and φ_t are intertemporal preference and labour supply shocks, respectively, driven by

$$\ln d_t = \rho_d \ln d_{t-1} + \varepsilon_{d,t} \quad (159)$$

$$\ln \varphi_t = \rho_\varphi \ln \varphi_{t-1} + \varepsilon_{\varphi,t} \quad (160)$$

$\varepsilon_{d,t} \sim \mathcal{N}(0, \sigma_d^2)$ and $\varepsilon_{\varphi,t} \sim \mathcal{N}(0, \sigma_\varphi^2)$. The per-period budget constraint of households is given by

$$\begin{aligned} C_{j,t} + I_{j,t} + \frac{M_{j,t}}{P_t} + \frac{B_{j,t+1}}{P_t} + \int q_{j,t+1,t} a_{j,t+1,t} d\omega_{j,t+1,t} \\ = W_{j,t} L_{j,t}^s + (r_t u_{j,t} - \mu^{-1} \Phi[u_{j,t}]) K_{j,t-1} + \frac{M_{j,t-1}}{P_t} + R_{t-1} \frac{B_{j,t-1}}{P_t} + a_{j,t} + T_t + F_t \end{aligned}$$

where I_t is investment, B_t are government bonds, a_t is the amount of Arrow-Debreu securities that pay C_t in event $\omega_{j,t+1,t}$ which cost q_t , W_t is the wage rate, r_t is the rental price of capital, $u_{j,t}$ is the utilisation rate of capital, $\mu^{-1} \Phi[u_{j,t}]$ is the cost of $u_{j,t}$, T_t are government transfers, and F_t is the household’s share of firm profits.

The law of motion for capital, K_t , is given by

$$K_{j,t} = (1 - \delta)K_{j,t-1} + \mu_t \left(1 - S \left[\frac{I_{j,t}}{I_{j,t-1}} \right] \right) I_{j,t} \quad (161)$$

where μ_t is an investment-specific technological shock,

$$\mu_t = \mu_{t-1} \exp(\Lambda_\mu + z_{\mu,t}) \quad (162)$$

$$z_{\mu,t} = \varepsilon_{\mu,t}, \varepsilon_{\mu,t} \sim \mathcal{N}(0, \sigma_\mu^2).$$

The first-order conditions of the households' maximization problem, with respect to $C_{j,t}$, $B_{j,t}$, $u_{j,t}$, $K_{j,t}$, $I_{j,t}$, and $M_{j,t}$ are given by

$$\begin{aligned} \lambda_{j,t} &= \frac{d_t}{(C_{j,t} - hC_{j,t-1})} - h\beta \mathbb{E}_t \left\{ \frac{d_{t+1}}{(C_{j,t+1} - hC_{j,t})} \right\} \\ \lambda_{j,t} &= \beta \mathbb{E}_t \left\{ \lambda_{j,t+1} \frac{R_t}{\Pi_{t+1}} \right\} \\ r_t &= \mu_t^{-1} \Phi'[u_{j,t}] \\ \varphi_{j,t} &= \beta \mathbb{E}_t \left\{ \frac{\lambda_{j,t+1}}{\lambda_{j,t}} ((1 - \delta)\varphi_{j,t+1} + r_{t+1} u_{j,t+1} + \mu_{t+1}^{-1} \Phi[u_{j,t+1}]) \right\} \\ 1 &= \varphi_{j,t} \mu_t \left(1 - S \left[\frac{I_{j,t}}{I_{j,t-1}} \right] - S' \left[\frac{I_{j,t}}{I_{j,t-1}} \right] \frac{I_{j,t}}{I_{j,t-1}} \right) + \\ &\quad \beta \mathbb{E}_t \left\{ \frac{\varphi_{j,t+1} \lambda_{j,t+1}}{\lambda_{j,t}} \mu_{t+1} S' \left[\frac{I_{j,t}}{I_{j,t-1}} \right] \left(\frac{I_{j,t}}{I_{j,t-1}} \right)^2 \right\} \end{aligned}$$

$\Pi_{t+1} = P_t/P_{t-1}$, where the first-order conditions's for wages , $W_{j,t}$, and labour supply, $L_{j,t}^s$, are given below.

The 'labour packer' uses the production function

$$L_t^d = \left(\int_0^1 \left(L_{j,t}^s \right)^{\frac{\eta-1}{\eta}} dj \right)^{\frac{\eta}{\eta-1}} \quad (163)$$

to aggregate labour and sell it as a 'labour good' to producers. Their maximization problem is given by

$$\max_{L_{j,t}^s} W_t L_t^d - \int_0^1 W_{j,t} L_{j,t}^s dj \quad (164)$$

As we observed with the basic New-Keynesian model, this will yield demand functions of the

form

$$L_{i,t}^s = \left(\frac{W_{i,t}}{W_t} \right)^{-\eta} L_t^d$$

by using a zero profit condition,

$$\int_0^1 W_{j,t} L_{j,t}^s dj = W_t L_t^d \quad (165)$$

and the aggregate wage index, W_t , is given by

$$W_t = \left(\int_0^1 W_{j,t}^{1-\eta} dj \right)^{\frac{1}{1-\eta}}. \quad (166)$$

A fraction, $1 - \theta_w$, of households can adjust their wages in a given period, say t , and the fraction who cannot reset their wages, θ_w , partially index their wages to past inflation, $\Pi_t = P_t / P_{t-1}$, where indexation is controlled by the parameter $\chi_w \in [0, 1]$. The maximization problem for the household is

$$\max_{W_{j,t}} \mathbb{E}_t \sum_{\tau=0}^{\infty} \theta_w^\tau \beta^\tau \left\{ -d_{t+\tau} \varphi_{t+\tau} \psi \frac{(L_{j,t+\tau}^s)^{1+\vartheta}}{1+\vartheta} + \lambda_{j,t+\tau} \prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^{\chi_w}}{\Pi_{t+s}} W_{j,t} L_{j,t+\tau}^s \right\} \quad (167)$$

subject to

$$L_{j,t+\tau}^s = \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^{\chi_w}}{\Pi_{t+s}} \frac{W_{j,t}}{W_{t+\tau}} \right)^{-\eta} L_{t+\tau}^d \quad (168)$$

The first-order condition of this problem is

$$\begin{aligned} & \frac{\eta-1}{\eta} W_t^* \mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_w)^\tau \lambda_{t+\tau} \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^{\chi_w}}{\Pi_{t+s}} \right)^{1-\eta} \left(\frac{W_t^*}{W_{t+\tau}} \right)^{-\eta} L_{t+\tau}^d \\ &= \mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_w)^\tau \left(d_{t+\tau} \varphi_{t+\tau} \psi \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^{\chi_w}}{\Pi_{t+s}} \frac{W_t^*}{W_{t+\tau}} \right)^{-\eta(1+\vartheta)} (L_{t+\tau}^d)^{1+\vartheta} \right) \end{aligned} \quad (169)$$

We can define this recursively by using auxiliary variables: set the left-hand side of the first-

order condition above equal to $\mathcal{F}_t^{(1)}$, and $\mathcal{F}_t^{(2)}$ for the right-hand side, with $\mathcal{F}_t^{(1)} = \mathcal{F}_t^{(2)}$,

$$\begin{aligned}\mathcal{F}_t^{(1)} &= \frac{\eta-1}{\eta} W_t^* \mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_w)^\tau \lambda_{t+\tau} \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^{\chi_w}}{\Pi_{t+s}} \right)^{1-\eta} \left(\frac{W_{t+\tau}}{W_t^*} \right)^\eta L_{t+\tau}^d \\ \mathcal{F}_t^{(2)} &= \mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_w)^\tau d_{t+\tau} \varphi_{t+\tau} \psi \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^{\chi_w}}{\Pi_{t+s}} \right)^{-\eta(1+\vartheta)} \left(\frac{W_{t+\tau}}{W_t^*} \right)^{\eta(1+\vartheta)} (L_{t+\tau}^d)^{1+\vartheta}\end{aligned}$$

Now note that we can rewrite $\mathcal{F}_t^{(1)}$ as

$$\mathcal{F}_t^{(1)} = \frac{\eta-1}{\eta} (W_t^*)^{1-\eta} \lambda_t W_t^\eta L_t^d + \beta \theta_w \mathbb{E}_t \left(\frac{\Pi_t^{\chi_w}}{\Pi_{t+1}} \right)^{1-\eta} \left(\frac{W_{t+1}}{W_t^*} \right)^{\eta-1} \mathcal{F}_{t+1}^{(1)} \quad (170)$$

Doing the same for $\mathcal{F}_t^{(2)}$,

$$\mathcal{F}_t^{(2)} = d_t \varphi_t \psi \left(\frac{W_t^*}{W_t} \right)^{-\eta(1+\vartheta)} + \beta \theta_w \mathbb{E}_t \left(\frac{\Pi_t^{\chi_w}}{\Pi_{t+1}} \right)^{-\eta(1+\vartheta)} \left(\frac{W_{t+1}}{W_t^*} \right)^{\eta(1+\vartheta)} \mathcal{F}_{t+1}^{(2)} \quad (171)$$

where W_t^* denotes the optimal wage. In a symmetric equilibrium, $1 - \theta_w$ of households set W_t^* as their wage, so the real wage index evolves as a (geometric) average of past real wages and the optimal wage,

$$W_t^{1-\eta} = \theta_w \left(\frac{\Pi_{t-1}^{\chi_w}}{\Pi_t} \right)^{1-\eta} W_{t-1}^{1-\eta} + (1 - \theta_w) (W_t^*)^{1-\eta}. \quad (172)$$

The production function of the final good domestic producer is given by

$$Y_t^d = \left(\int_0^1 (Y_{i,t})^{\frac{\varepsilon-1}{\varepsilon}} di \right)^{\frac{\varepsilon}{\varepsilon-1}} \quad (173)$$

which aggregates the output of intermediate goods producers. The maximization problem for the final good producer is

$$\max_{Y_{i,t}} P_t \left(\int_0^1 (Y_{i,t})^{\frac{\varepsilon-1}{\varepsilon}} di \right)^{\frac{\varepsilon}{\varepsilon-1}} - \int_0^1 P_{i,t} Y_{i,t} di \quad (174)$$

which, as in the case of wages, results in demand functions of the form

$$Y_{i,t} = \left(\frac{P_{i,t}}{P_t} \right)^{-\varepsilon} Y_t^d \quad (175)$$

for an arbitrary i and the price index is defined as

$$P_t = \left(\int_0^1 P_{i,t}^{1-\varepsilon} di \right)^{-\frac{1}{1-\varepsilon}} \quad (176)$$

Intermediate goods are produced according to the technology

$$Y_{i,t} = A_t K_{i,t-1}^\alpha \left(L_{i,t}^d \right)^{1-\alpha} - \phi z_t \quad (177)$$

where α is capital's share of income, ϕ is a fixed cost, and A_t is a 'neutral technology shock' given by

$$A_t = A_{t-1} \exp(\Lambda_A + z_{A,t}) \quad (178)$$

$z_{A,t} = \varepsilon_{A,t}$ and $\varepsilon_{A,t} \sim \mathcal{N}(0, \sigma_A^2)$. Note that this introduces a second unit-root into the model (the first being μ_t). We can define

$$z_t = A_t^{\frac{1}{1-\alpha}} \mu_t^{\frac{\alpha}{1-\alpha}} \quad (179)$$

which implies that economic profits, in the long-run, will be approximately zero, and we can also write z_t as

$$z_t = z_{t-1} \exp(\Lambda_z + z_{z,t}) \quad (180)$$

where

$$z_{z,t} = \frac{z_{A,t} + \alpha z_{\mu,t}}{1 - \alpha} \quad (181)$$

with

$$\Lambda_z = \frac{\Lambda_A + \alpha \Lambda_\mu}{1 - \alpha} \quad (182)$$

Intermediate goods producers first minimise cost according to

$$\min_{L_{i,t}^d, K_{i,t-1}} W_t L_{i,t}^d + r_t K_{i,t-1} \quad (183)$$

such that

$$Y_{i,t} = \max \left\{ A_t K_{i,t-1}^\alpha \left(L_{i,t}^d \right)^{1-\alpha} - \phi z_t, 0 \right\} \quad (184)$$

by taking input prices as given and assuming perfectly competitive factor markets. The first-order conditions of this problem, along with real cost, $W_t L_{i,t}^d + r_t K_{i,t-1}$, and the production function, can be used to find real marginal cost, MC_t ,

$$MC_t = \left(\frac{1}{1-\alpha} \right)^{1-\alpha} \left(\frac{1}{\alpha} \right)^\alpha \frac{1}{A_t} W_t^{1-\alpha} r_t^\alpha \quad (185)$$

Then, firms solve a maximization problem by choosing an optimal price, where $1 - \theta_p$ percent of firms can reset their prices in a given period, and θ_p percent of firms that index their prices to past inflation. Indexation is controlled by $\chi \in [0, 1]$, where $\chi = 1$ is full indexation. The problem of the firm is then

$$\max_{P_{i,t}} \mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_p)^\tau \frac{\lambda_{t+\tau}}{\lambda_t} \left\{ \left(\prod_{s=1}^{\tau} \Pi_{t+s-1}^\chi \frac{P_{i,t}}{P_{t+\tau}} - MC_{t+\tau} \right) Y_{i,t+\tau} \right\} \quad (186)$$

subject to

$$Y_{i,t+\tau} = \left(\prod_{s=1}^{\tau} \Pi_{t+s-1}^\chi \frac{P_{i,t}}{P_{t+\tau}} \right)^{-\varepsilon} Y_{t+\tau}^d \quad (187)$$

The first-order condition of this problem is

$$\mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_p)^\tau \frac{\lambda_{t+\tau}}{\lambda_t} \left\{ \left((1-\varepsilon) \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^\chi P_{i,t}}{\Pi_{t+s} P_t} \right)^{1-\varepsilon} \frac{1}{P_{i,t}} + \varepsilon \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^\chi P_{i,t}}{\Pi_{t+s} P_t} \right)^{-\varepsilon} \frac{1}{P_{i,t}} MC_{t+\tau} \right) Y_{t+\tau}^d \right\} = 0$$

Setting $P_{i,t} = P_{i,t}^* = P_t^*$, where '*' stands for the optimum,

$$\mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_p)^\tau \lambda_{t+\tau} \left\{ \left((1-\varepsilon) \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^\chi}{\Pi_{t+s}} \right)^{1-\varepsilon} \frac{P_t^*}{P_t} + \varepsilon \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^\chi}{\Pi_{t+s}} \right)^{-\varepsilon} MC_{t+\tau} \right) Y_{t+\tau}^d \right\} = 0$$

As before, we can rewrite this recursively by using auxiliary variables,

$$\mathcal{G}_t^{(1)} = \mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_p)^\tau \lambda_{t+\tau} \left\{ \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^\chi}{\Pi_{t+s}} \right)^\varepsilon MC_{t+\tau} Y_{t+\tau}^d \right\}$$

and

$$\mathcal{G}_t^{(2)} = \mathbb{E}_t \sum_{\tau=0}^{\infty} (\beta \theta_p)^\tau \lambda_{t+\tau} \left\{ \left(\prod_{s=1}^{\tau} \frac{\Pi_{t+s-1}^\chi}{\Pi_{t+s}} \right)^{1-\varepsilon} \frac{P_t^*}{P_t} Y_{t+\tau}^d \right\}$$

where $\mathcal{G}_t^{(1)}$ and $\mathcal{G}_t^{(2)}$ are related by the fact that $\epsilon \mathcal{G}_t^{(1)} = (\epsilon - 1) \mathcal{G}_t^{(2)}$. Now,

$$\mathcal{G}_t^{(1)} = \lambda_t M C_t Y_t^d + \mathbb{E}_t \beta \theta_p \left\{ \left(\frac{\Pi_t^\chi}{\Pi_{t+1}} \right)^{-\varepsilon} \mathcal{G}_{t+1}^{(1)} \right\} \quad (188)$$

and

$$\mathcal{G}_t^{(2)} = \lambda_t \Pi_t^* Y_t^d + \mathbb{E}_t \beta \theta_p \left\{ \left(\frac{\Pi_t^\chi}{\Pi_{t+1}} \right)^{1-\varepsilon} \left(\frac{\Pi_t^*}{\Pi_{t+1}^*} \right) \mathcal{G}_{t+1}^{(2)} \right\} \quad (189)$$

where $\Pi_t^* = P_t^*/P_t$. Calvo pricing implies that the price index follows,

$$P_t^{1-\varepsilon} = \theta_p (\Pi_{t-1}^\chi)^{1-\varepsilon} P_{t-1}^{1-\varepsilon} + (1 - \theta_p) (P_t^*)^{1-\varepsilon}$$

and dividing both sides by $P_t^{1-\varepsilon}$, we can rewrite as

$$1 = \theta_p \left(\frac{\Pi_{t-1}^\chi}{\Pi_t} \right)^{1-\varepsilon} P_{t-1}^{1-\varepsilon} + (1 - \theta_p) (\Pi_t^*)^{1-\varepsilon}$$

A nominal interest rate, R_t , is set by government according to a Taylor rule,

$$\frac{R_t}{R_*} = \left(\frac{R_{t-1}}{R_*} \right)^{\gamma_R} \left[\left(\frac{\Pi_t}{\Pi_*} \right)^{\gamma_\Pi} \left(\frac{Y_t/Y_{t-1}}{\Lambda_{Y^d}} \right)^{\gamma_Y} \right]^{1-\gamma_R} \exp(\xi_t^R),$$

where $\xi_t^R = \varepsilon_{R,t}$, $\varepsilon_{R,t} \sim \mathcal{N}(0, \sigma_R)$, is a monetary policy shock. This is financed through lump-sum transfers T_t , such that the government runs a balanced budget,

$$T_t = \frac{\int_0^1 M_t(j) - M_{t-1}(j) dj}{P_t} + \frac{\int_0^1 B_{t+1}(j) dj}{P_t} - R_t \frac{\int_0^1 B_t(j) dj}{P_t} \quad (190)$$

The model is now closed, bar a few aggregation equations. Aggregate demand and aggregate supply are

$$Y_t^d = C_t + I_t + \frac{1}{\mu_t} \Phi[u_t] K_{t-1} \quad (191)$$

$$Y_t = \frac{1}{\nu_t^p} (A_t (u_t K_{t-1})^\alpha (L_t^d)^{1-\alpha} - \phi z_t) \quad (192)$$

respectively, where ν_t^p is an inefficiency arising from price dispersion,

$$\nu_t^p = \int_0^1 \left(\frac{P_{i,t}}{P_t} \right)^{-\varepsilon} di \quad (193)$$

Labour ‘packed’ is

$$L_t^d = \frac{L_t}{\nu_t^w} \quad (194)$$

where ν_t^w is an inefficiency arising from wage dispersion,

$$\nu_t^w = \int_0^1 \left(\frac{W_{i,t}}{W_t} \right)^{-\eta} di \quad (195)$$

By Calvo pricing, we have

$$\nu_t^p = \theta_p \left(\frac{\Pi_{t-1}^\chi}{\Pi_t} \right)^{-\varepsilon} \nu_{t-1}^p + (1 - \theta_p) (\Pi_t^*)^{-\eta} \quad (196)$$

$$\nu_t^w = \theta_w \left(\frac{\Pi_{t-1}^{\chi_w}}{\Pi_t} \right)^{-\eta} \nu_{t-1}^w + (1 - \theta_w) (\Pi_t^{w*})^{-\eta} \quad (197)$$

As there are unit roots in the model, we can make the variables stationary by using: $\tilde{c}_t = C_t/z_t$, $\tilde{\lambda}_t = \lambda_t z_t$, $\tilde{r}_t = r_t \mu_t$, $\tilde{q}_t = q_t \mu_t$, $\tilde{i}_t = I_t/z_t$, $\tilde{w}_t = W_t/z_t$, $\tilde{w}_t^* = W_t^*/z_t$, $\tilde{k}_t = K_t/(z_t \mu_t)$, and $\tilde{y}_t^d = Y_t^d/z_t$.

9.3.1. Steady State

The steady-state of this model is a little complicated. From Fernández-Villaverde and Rubio-Ramírez (2006) we have that

$$\begin{aligned}
\tilde{r} &= \frac{1 - (\beta / (\tilde{z}\tilde{\mu})) (1 - \delta)}{(\beta / (\tilde{z}\tilde{\mu}))} \\
R &= \frac{\Pi \tilde{z}}{\beta} \\
\Pi^* &= \left(\frac{1 - \theta_p \Pi^{-(1-\varepsilon)(1-\chi)}}{1 - \theta_p} \right)^{1/(1-\varepsilon)} \\
mc &= \frac{\varepsilon - 1}{\varepsilon} \frac{1 - \beta \theta_p \Pi^{\varepsilon(1-\chi)}}{1 - \beta \theta_p \Pi^{-(1-\varepsilon)(1-\chi)}} \Pi^* \\
\Pi^{w^*} &= \left(\frac{1 - \theta_w \Pi^{-(1-\eta)(1-\chi_w)} (\tilde{z})^{-(1-\eta)}}{1 - \theta_w} \right)^{1/(1-\eta)} \\
\tilde{w} &= (1 - \alpha) \left(mc \left(\frac{\alpha}{\tilde{r}} \right)^\alpha \right)^{1/(1-\alpha)} \\
\tilde{w}^* &= \tilde{w} \Pi^{w^*} \\
\tilde{v}^p &= \frac{1 - \theta_p}{1 - \theta_p \Pi^{(1-\chi)\varepsilon}} (\Pi^*)^{-\varepsilon} \\
\tilde{v}^w &= \frac{1 - \theta_w}{1 - \theta_w \Pi^{(1-\chi_w)\eta} (\tilde{z})^\eta} (\Pi^{w^*})^{-\eta} \\
l &= \tilde{v}^w l^d \\
\Omega &= \frac{\alpha}{1 - \alpha} \frac{\tilde{w}}{\tilde{r}} \tilde{z} \tilde{\mu}
\end{aligned}$$

The authors use the following non-linear equation to determine the steady-state value for labour demand l^d ,

$$\begin{aligned}
&\frac{1 - \beta \theta_w \tilde{z}^{\eta(1+\vartheta)} \Pi^{\eta(1-\chi_w)(1+\vartheta)}}{1 - \beta \theta_w \tilde{z}^{\eta-1} \Pi^{-(1-\chi_w)(1-\eta)}} \\
&= \frac{\psi(\Pi^{w^*})^{-\eta\vartheta} (l^d)^\vartheta}{\frac{\eta-1}{\eta} w^* (1 - h\beta) d \tilde{z} \left(1 - \frac{h}{\tilde{z}}\right)^{-1} \left(\left(\frac{\tilde{A}}{\tilde{z}} (\nu^p)^{-1} \Omega^\alpha - \frac{\tilde{z}\tilde{\mu} - (1-\delta)}{\tilde{z}\tilde{\mu}} \Omega \right) l^d - (\nu^p)^{-1} \phi \right)^{-1}} \quad (198)
\end{aligned}$$

To solve for l^d , the authors note the need to use a root finder, but, using an assumption from a later paper, there does exist an analytic solution for this equation, which is where my version differs with theirs. For notational convenience, define the left hand side of the above equation

as

$$\Xi = \frac{1 - \beta \theta_w \tilde{z}^{\eta(1+\vartheta)} \Pi^{\eta(1-\chi_w)(1+\vartheta)}}{1 - \beta \theta_w \tilde{z}^{\eta-1} \Pi^{-(1-\chi_w)(1-\eta)}}$$

In (Fernández-Villaverde et al., 2009, Section 3.1) the authors set $\phi = 0$, so

$$\begin{aligned} \Xi &= \frac{\psi(\Pi^{w^*})^{-\eta\vartheta}(l^d)^\vartheta}{\frac{\eta-1}{\eta} w^*(1-h\beta) d \left(\frac{\tilde{z}^2}{\tilde{z}-h} \right) \left(\left(\frac{\tilde{A}}{\tilde{z}} (\nu^p)^{-1} \Omega^\alpha - \frac{\tilde{z}\tilde{\mu}-(1-\delta)}{\tilde{z}\tilde{\mu}} \Omega \right) l^d \right)^{-1}} \\ &= \frac{\psi(\Pi^{w^*})^{-\eta\vartheta}(l^d)^\vartheta}{\frac{\eta-1}{\eta} w^*(1-h\beta) d \left(\frac{\tilde{z}^2}{\tilde{z}-h} \right) \left(\left(\frac{\tilde{A}(\nu^p)^{-1} \Omega^\alpha \tilde{\mu} - (\tilde{z}\tilde{\mu}-(1-\delta))\Omega}{\tilde{z}\tilde{\mu}} \right) l^d \right)^{-1}} \\ &= \frac{\psi(\Pi^{w^*})^{-\eta\vartheta}(l^d)^\vartheta l^d}{\frac{\eta-1}{\eta} w^*(1-h\beta) d \left(\frac{\tilde{z}^3}{\tilde{z}-h} \right) \left(\frac{\tilde{A}(\nu^p)^{-1} \Omega^\alpha \tilde{\mu} - (\tilde{z}\tilde{\mu}-(1-\delta))\Omega}{\tilde{\mu}} \right)^{-1}} \end{aligned}$$

This can be rewritten as

$$\begin{aligned} l^d &= \\ &\left[\frac{\Xi}{\psi(\Pi^{w^*})^{-\eta\vartheta}} \left(\frac{\eta-1}{\eta} w^*(1-h\beta) d \left(\frac{\tilde{z}^3}{\tilde{z}-h} \right) \left(\frac{\tilde{A}(\nu^p)^{-1} \Omega^\alpha \tilde{\mu} - (\tilde{z}\tilde{\mu}-(1-\delta))\Omega}{\tilde{\mu}} \right)^{-1} \right) \right]^{\frac{1}{1+\vartheta}} \end{aligned} \tag{199}$$

This is messy, but would be computationally simpler when estimating such a model than using numerical methods to solve for l^d .¹¹

If the reader prefers not to assume $\phi = 0$, then simple application of a Newton-Raphson algorithm will yield the desired result, *i.e.*, set

$$g(l^d) = -\Xi_1 + \frac{\psi(\Xi_2(l^d)^\vartheta)}{\Xi_3(\Xi_4 l^d - (\nu^p)^{-1} \phi)^{-1}}$$

and find $g(l^d) = 0$, where

$$\Xi_1 = \frac{1 - \beta \theta_w \tilde{z}^{\eta(1+\vartheta)} \Pi^{\eta(1-\chi_w)(1+\vartheta)}}{1 - \beta \theta_w \tilde{z}^{\eta-1} \Pi^{-(1-\chi_w)(1-\eta)}}$$

¹¹ l^d is defined on \mathbb{R}_{++} , so we focus on the principal root, and ignore any complex roots.

$\Xi_2 = \psi(\Pi^{w^*})^{-\eta\vartheta}$, $\Xi_3 = \frac{\eta-1}{\eta} w^*(1-h\beta) d\tilde{z} \left(1 - \frac{h}{\tilde{z}}\right)^{-1}$, and

$$\Xi_4 = \left(\frac{\tilde{A}}{\tilde{z}} (\nu^p)^{-1} \Omega^\alpha - \frac{\tilde{z}\tilde{\mu} - (1-\delta)}{\tilde{z}\tilde{\mu}} \Omega \right)$$

Let $i \in [1, \mathcal{I}]$ index the iterations, set $(l^d)_0 = 0.0001$, and $\epsilon = 10^{-8}$; then

$$(l^d)_{i+1} = (l^d)_i - \frac{g((l^d)_i)}{g'((l^d)_i)}$$

where $g'((l^d)_i)$ is approximated by a numerical derivative, *i.e.*,

$$g'((l^d)_i) \approx \frac{g((l^d)_i + \Delta) - g((l^d)_i)}{\Delta}$$

Δ = ‘small’. When $| (l^d)_{i+1} - (l^d)_i | < \epsilon$, stop.

Having solved for l^d , we use this for

$$\tilde{k} = \Omega l^d \tag{200}$$

$$\tilde{i} = \frac{\tilde{z}\tilde{\mu} - (1-\delta)}{\tilde{z}\tilde{\mu}} \tilde{k} \tag{201}$$

$$\tilde{y}^d = \frac{\frac{\tilde{A}}{\tilde{z}} (\tilde{k})^\alpha (l^d)^{1-\alpha}}{\nu^p} \tag{202}$$

$$\tilde{c} = \left(\frac{\tilde{A}}{\tilde{z}} (\nu^p)^{-1} \Omega^\alpha - \frac{\tilde{z}\tilde{\mu} - (1-\delta)}{\tilde{z}\tilde{\mu}} \Omega \right) l^d \tag{203}$$

And the model is complete.

9.3.2. Log-linearized Model

For notational convenience, use the following definitions

$$\begin{aligned}
\mathcal{A}_p &= \frac{A_*}{z_*} (u_* k_*)^\alpha (l_*^d)^{1-\alpha} \\
\phi_u &= \frac{\gamma_2}{\gamma_1} \\
\mathcal{A}_1 &= \frac{\theta_w \Pi^{-(1-\chi_w)(1-\eta)} z_*^{-(1-\eta)}}{1 - \theta_w} (\Pi^{w^*})^{1-\eta} \\
\mathcal{A}_2 &= \frac{\theta_p \Pi^{-(1-\epsilon)(1-\chi)}}{1 - \theta_p} (\Pi^*)^{1-\epsilon} \\
\mathcal{A}_3 &= \beta \theta_p \Pi^{\epsilon(1-\chi)} \\
\mathcal{A}_4 &= \theta_w \Pi^{\eta(1-\chi_w)} z_*^\eta \\
\mathcal{A}_5 &= \beta \theta_w z_*^{\eta-1} \Pi^{-(1-\eta)(1-\chi_w)} \\
\mathcal{A}_6 &= \beta \theta_w z_*^{\eta(1+\vartheta)} \Pi^{\eta(1+\vartheta)(1-\chi_w)} \\
\mathcal{A}_7 &= \beta \theta_p \Pi^{-(1-\epsilon)(1-\chi)}
\end{aligned}$$

Expectational block:

$$\begin{aligned}
d_t &= (1 - h\beta z_*)\lambda_t + h\beta z_* \mathbb{E}_t d_{t+1} + \frac{1 + (h^2)\beta}{1 - \frac{h}{z_*}} c_t - \frac{h}{z_*(1 - \frac{h}{z_*})} \mathbb{E}_t c_{t-1} - \frac{h\beta z_*}{1 - \frac{h}{z_*}} c_{t+1} + \frac{h}{z_*(1 - \frac{h}{z_*})} z_t \\
\lambda_t &= \mathbb{E}_t \lambda_{t+1} + R_t - \mathbb{E}_t \pi_{t+1} \\
q_t &= \mathbb{E}_t \lambda_{t+1} - \lambda_t + \frac{\beta(1-\delta)}{z_* \mu_*} \mathbb{E}_t q_{t+1} + \left(1 - \frac{\beta(1-\delta)}{z_* \mu_*}\right) \mathbb{E}_t r_{t+1} \\
q_t &= \kappa(z_*^2)(i_t - i_{t-1} + z_t) - \beta \kappa(z_*^2)(\mathbb{E}_t i_{t+1} - i_t) \\
f_t &= (1 - \mathcal{A}_5)((1 - \eta)w_t^* + \lambda + \eta w_t + l_t^d) + \mathcal{A}_5(\mathbb{E}_t f_{t+1} - (1 - \eta)(\mathbb{E}_t \pi_{t+1} - \chi_w \pi_t + \mathbb{E}_t w_{t+1}^* - w_t^*)) \\
f_t &= (1 - \mathcal{A}_6)(d_t + \varphi_t + \eta(1 + \vartheta)(w_t - w_t^*) + (1 + \vartheta)l_t^d) \\
&\quad + \mathcal{A}_6(\mathbb{E}_t f_{t+1} + \eta(1 + \vartheta)(\mathbb{E}_t \pi_{t+1} - \chi_w \pi_t + \mathbb{E}_t w_{t+1}^* - w_t^*)) \\
g_t^{(1)} &= (1 - \mathcal{A}_3)(\lambda_t + mc_t + y_t) + \mathcal{A}_3(\epsilon(\mathbb{E}_t \pi_{t+1} - \chi \pi_t) + \mathbb{E}_t g_{t+1}^{(1)}) \\
g_t^{(2)} &= (1 - \mathcal{A}_7)\lambda_t + \pi_t^* + (1 - \mathcal{A}_7)y_t + (\epsilon - 1)\mathcal{A}_7 \mathbb{E}_t \pi_{t+1} - \chi(\epsilon - 1)\mathcal{A}_7 \pi_t - \mathcal{A}_7 \mathbb{E}_t \pi_{t+1}^* + \mathcal{A}_7 \mathbb{E}_t g_{t+1}^{(2)}
\end{aligned}$$

The deterministic block of the model is

$$\begin{aligned}
w_t^* - w_t &= \mathcal{A}_1 \pi_t - \chi_w \mathcal{A}_1 \pi_{t-1} + \mathcal{A}_1 w_t - \mathcal{A}_1 w_{t-1} + \mathcal{A}_1 z_t \\
\pi_t^* &= \mathcal{A}_2 \pi_t - \mathcal{A}_2 \chi \pi_{t-1} \\
r_t &= \phi_u u_t \\
u_t &= -k_{t-1} + l_t^d + w_t - r_t + z_t + \mu_t \\
mc_t &= (1 - \alpha)w_t + \alpha r_t \\
R_t &= \gamma_R R_{t-1} + (1 - \gamma_R)\gamma_\pi \pi_t + (1 - \gamma_R)\gamma_y (y_t - y_{t-1} + z_t) + \xi_t^R \\
y_* y_t &= c_* c_t + i_* i_t + \frac{\gamma_1 k_*}{z_* \mu_*} u_t \\
(y_* v_t^p)(v_t^p + y_t) &= \mathcal{A}_p (A_t - z_t + \alpha(u_t + k_{t-1}) + (1 - \alpha)l_t^d) \\
v_t^p &= \frac{\mathcal{A}_3}{\beta} \epsilon \pi_t - \chi \epsilon \frac{\mathcal{A}_3}{\beta} \pi_{t-1} + \frac{\mathcal{A}_3}{\beta} v_{t-1}^p - \left(1 - \frac{\mathcal{A}_3}{\beta}\right) \epsilon \pi_t^* \\
v_t^w &= \mathcal{A}_4 (\eta(\pi_t - \chi_w \pi_{t-1} + w_t - w_{t-1} + z_t) + v_{t-1}^w) - (1 - \mathcal{A}_4) \eta(w_t^* - w_t) \\
l_t^s &= v_t^w + l_t^d \\
k_t &= \frac{1 - \delta}{z_* \mu_*} k_{t-1} + \frac{z_* \mu_* - (1 - \delta)}{z_* \mu_*} i_t - \frac{1 - \delta}{z_* \mu_*} (z_t + \mu_t) \\
z_t &= \frac{A_t + \alpha \mu_t}{1 - \alpha} \\
g_t^{(1)} &= g_t^{(2)}
\end{aligned}$$

And shocks

$$\begin{aligned}
\mu_t &= \varepsilon_{\mu,t} \\
d_t &= \rho_d d_{t-1} + \varepsilon_{d,t} \\
\varphi_t &= \rho_\varphi \varphi_{t-1} + \varepsilon_{\varphi,t} \\
A_t &= \varepsilon_{A,t} \\
\xi_t^R &= \varepsilon_{m,t}
\end{aligned}$$

As an exercise, the reader can attempt to solve this model using the methodology previously outlined; hint: Fernández-Villaverde and Rubio-Ramírez (2006) contains a lot of what you'll need, but $l \neq n$ in this model! The IRFs of inflation, the real wage, the interest rate, output, Tobin's Q, real marginal cost, and labour supply, for one unit investment-specific, pref-

erence, and labour shocks, is given below, based on calibrating parameter values to the median estimates found in Fernández-Villaverde (2010).

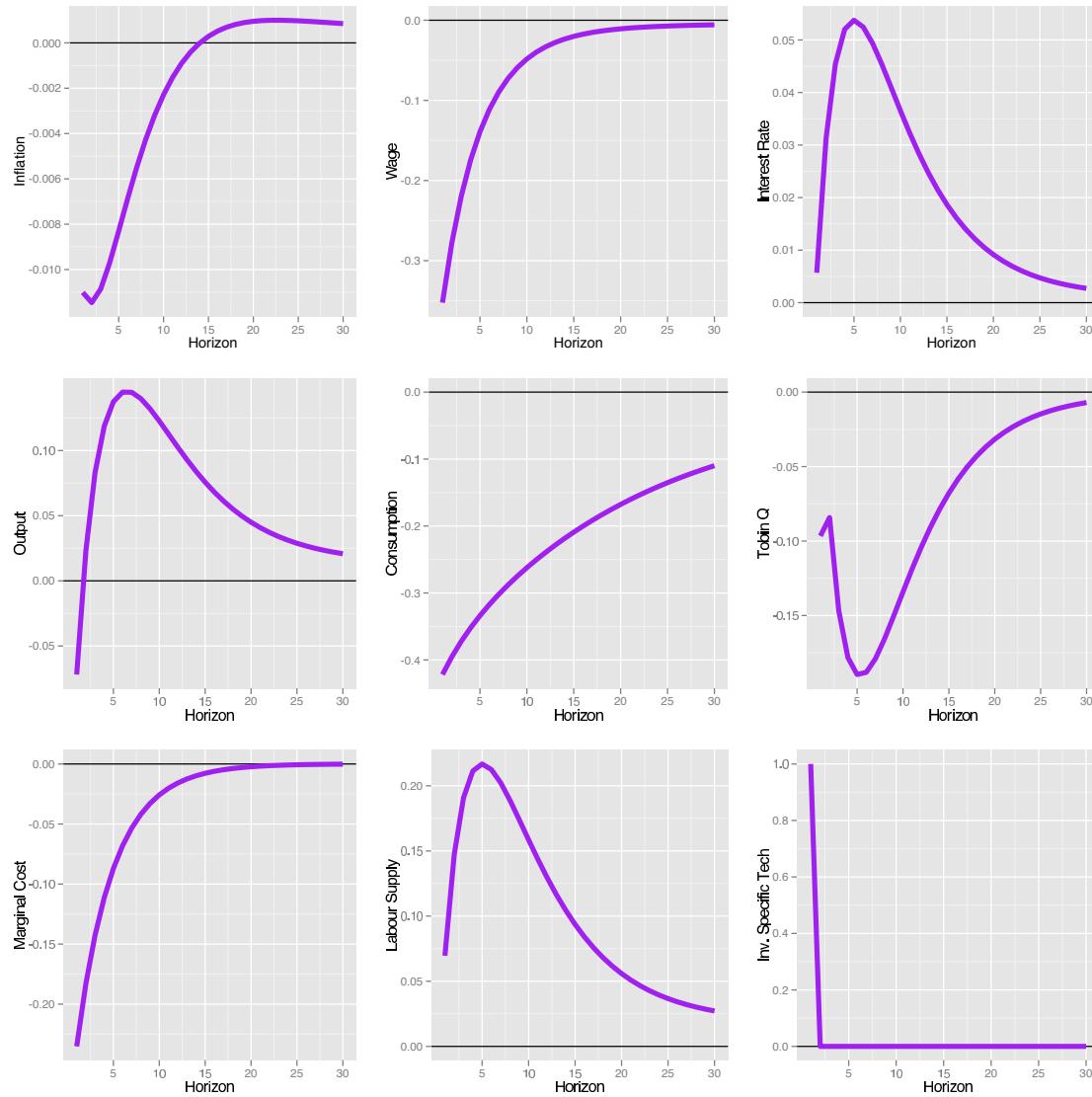


Figure 37: Investment Specific Technology Shock.

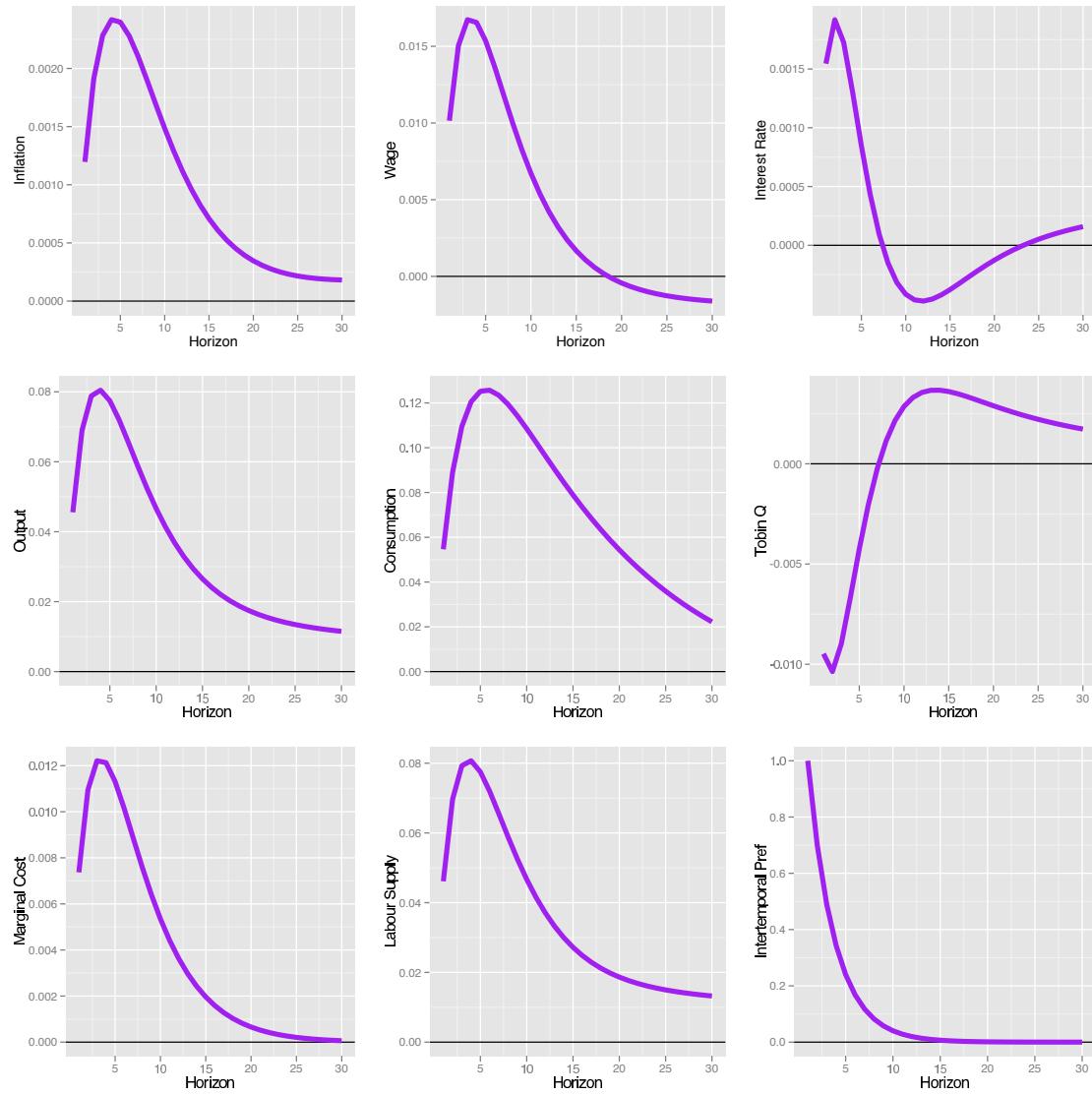


Figure 38: Intertemporal Preference Shock.

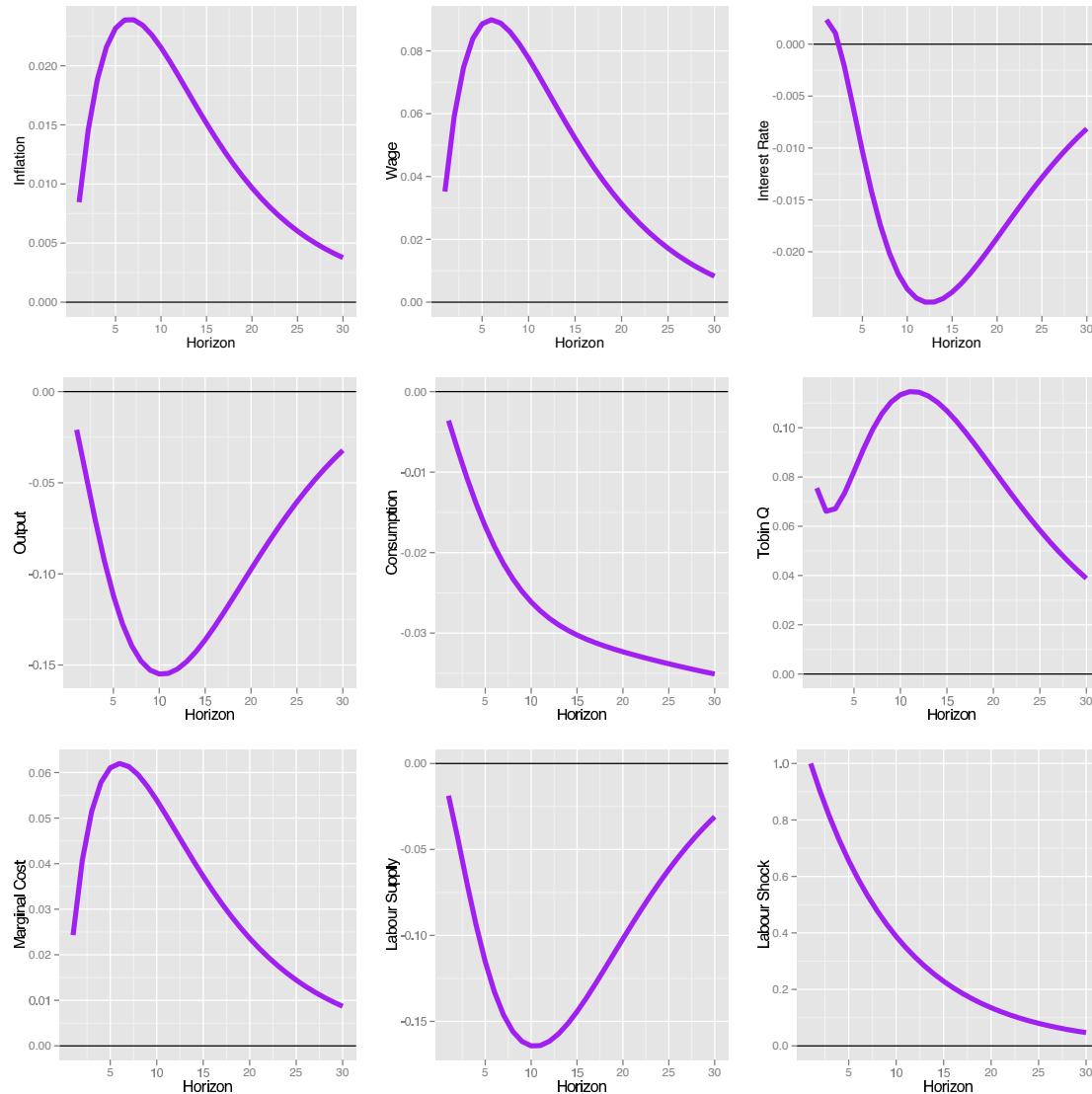


Figure 39: Labour Supply Shock.

10. DSGE-VAR

The basic idea of the DSGE-VAR(λ) is to use the implied moments of a DSGE model as the prior for a Bayesian VAR.

10.1. Details

Recall that, in stacked form, the VAR(p) model is written as

$$Y = Z\beta + \varepsilon$$

where Y and Z are $T \times n$ and $T \times pn$, and the likelihood function is proportional to

$$p(Y|\beta, \Sigma_\varepsilon) \propto |\Sigma_\varepsilon|^{-T/2} \exp \left\{ -\frac{1}{2} \text{tr} [\Sigma_\varepsilon^{-1} (Y^\top Y - \beta^\top Z^\top Y - Y^\top Z\beta + \beta^\top Z^\top Z\beta)] \right\} \quad (204)$$

The hybrid DSGE-VAR differs from BVAR models considered previously in that the prior for $(\beta, \Sigma_\varepsilon)$ will be a function of the ‘deep’ parameters of the DSGE model.

We can think of the DSGE prior as introducing λT ‘artificial’ observations from the DSGE model, where $\lambda \in (0, \infty]$, in a sense, weights how tight our prior belief is for the DSGE model compared to an unrestricted VAR. As $\lambda \nearrow \infty$, we approach the dynamics implied by the DSGE model.

The prior is expressed in terms of scaled population moments from the DSGE model. This gives a prior of the form

$$\begin{aligned} p(\beta, \Sigma_\varepsilon | \theta) &= c^{-1}(\theta) |\Sigma_\varepsilon|^{-\frac{\lambda T + n + 1}{2}} \\ &\times \exp \left\{ -\frac{1}{2} \text{tr} [\lambda T \Sigma_\varepsilon^{-1} (\Gamma_{YY}^*(\theta) - \beta^\top \Gamma_{ZY}^*(\theta) - \Gamma_{YZ}^*(\theta) \beta + \beta^\top \Gamma_{ZZ}^*(\theta) \beta)] \right\} \end{aligned}$$

where $c^{-1}(\theta)$ is a normalizing constant that ensures this density integrates to one, and $\Gamma_{YY}^*(\theta) := \mathbb{E}_\theta[Y_t Y_t^\top]$, etc, are the population moments implied by the solution to our DSGE model.

In section 8.1, we noted that the DSGE model solution can be expressed as

$$\begin{aligned} \xi_t &= \mathcal{F}(\theta) \xi_{t-1} + \mathcal{G}(\theta) \varepsilon_t \\ Y_t &= \mathcal{C} + \mathcal{H}^\top \xi_t + \mathcal{R} \end{aligned}$$

where ξ_t is the state, and \mathcal{F} & \mathcal{G} are functions of the deep parameters of the DSGE model. To form the Γ^* matrices above, we first compute the steady-state covariance matrix of the state by solving the discrete Lyapunov equation

$$\Omega_{ss} = \mathcal{F}\Omega_{ss}\mathcal{F}^\top + \mathcal{G}Q\mathcal{G}^\top$$

using a doubling algorithm. Then we compute the Γ^* matrices with

$$\begin{aligned}\Gamma_{YY}^*(\theta) &= \mathcal{H}^\top \Omega_{ss} \mathcal{H} + \mathcal{R} \\ \Gamma_{YZ_h}^*(\theta) &= \mathcal{H}^\top \mathcal{F}^h \Omega_{ss} \mathcal{H}\end{aligned}$$

where $h = 1, 2, \dots, p$.

Though an exact finite-order VAR representation doesn't exist, define the VAR approximation to the DSGE model as

$$\begin{aligned}\beta^*(\theta) &= [\Gamma_{ZZ}^*(\theta)]^{-1} \Gamma_{ZY}^*(\theta) \\ \Sigma_\varepsilon^*(\theta) &= \Gamma_{YY}^*(\theta) - \Gamma_{YZ}^*(\theta)[\Gamma_{ZZ}^*(\theta)]^{-1} \Gamma_{ZY}^*(\theta)\end{aligned}$$

Given a set of values θ , the prior distribution is of the usual inverse-Wishart-Normal form:

$$\begin{aligned}\Sigma_\varepsilon | \theta &\sim \mathcal{IW}(\lambda T \Sigma_\varepsilon^*(\theta), \lambda T - p \times n, n) \\ \beta | \Sigma_\varepsilon, \theta &\sim \mathcal{N}(\beta^*(\theta), \Sigma_\varepsilon \otimes (\lambda T \Gamma_{ZZ}^*(\theta))^{-1})\end{aligned}$$

The joint prior of the VAR parameters and DSGE parameters is then given by

$$p(\beta, \Sigma_\varepsilon, \theta) = p(\beta, \Sigma_\varepsilon | \theta)p(\theta)$$

where the prior $p(\theta)$ is the same as in section 8. The joint posterior distribution is factorized similarly:

$$p(\beta, \Sigma_\varepsilon, \theta | Y) = p(\beta, \Sigma_\varepsilon | Y, \theta)p(\theta | Y)$$

Note, however, that $p(\theta | Y)$ will involve a different likelihood function for $p(Y | \theta)$ as we've assumed the form (204). This will require integrating out the VAR parameters.

The maximum likelihood estimates are

$$\begin{aligned}\hat{\beta}(\theta) &= [\lambda T \Gamma_{ZZ}^*(\theta) + Z^\top Z]^{-1} [\lambda T \Gamma_{ZY}^* + Z^\top Y] \\ \hat{\Sigma}_\epsilon(\theta) &= \frac{1}{(\lambda+1)T} [(\lambda T \Gamma_{YY}^*(\theta) + Y^\top Y)] \\ &\quad - \frac{1}{(\lambda+1)T} [(\lambda T \Gamma_{YZ}^*(\theta) + Y^\top Z)(\lambda T \Gamma_{ZZ}^*(\theta) + Z^\top Z)^{-1} (\lambda T \Gamma_{ZY}^*(\theta) + Z^\top Y)]\end{aligned}$$

The prior and likelihood are conjugate, and so the posterior distributions are of the form

$$\begin{aligned}\Sigma_\epsilon | Y, \theta &\sim \mathcal{IW}((\lambda+1)T \hat{\Sigma}_\epsilon(\theta), (1+\lambda)T - p \times n, n) \\ \beta | Y, \Sigma_\epsilon, \theta &\sim \mathcal{N}(\hat{\beta}(\theta), \Sigma_\epsilon \otimes (\lambda T \Gamma_{ZZ}^*(\theta) + Z^\top Z)^{-1})\end{aligned}$$

Thus, for a given θ , we draw a Σ_ϵ from an inverse-Wishart distribution. Then, using the Σ_ϵ draw, we draw a vector of coefficients β using the multivariate normal distribution.

The posterior for θ is of the usual form:

$$p(\theta | Y) \propto p(Y | \theta) p(\theta)$$

that is, proportional to the likelihood of the data times the prior. However, we use the marginal likelihood

$$p(Y | \theta) = \int p(Y | \beta, \Sigma_\epsilon) p(\beta, \Sigma_\epsilon | \theta) d(\beta, \Sigma_\epsilon)$$

instead of a state space approach, where we previously used the Kalman filter to integrate out the unknown state. The closed form expression for this marginal likelihood is

$$\begin{aligned}p(Y | \theta) &= \frac{p(Y | \beta, \Sigma) p(\beta, \Sigma | \theta)}{p(\beta, \Sigma | Y)} \\ &= \frac{|\lambda T \Gamma_{ZZ}^*(\theta) + Z^\top Z|^{-\frac{n}{2}} |(\lambda+1)T \hat{\Sigma}_\epsilon(\theta)|^{-\frac{(\lambda+1)T-k}{2}}}{|\lambda T \Gamma_{ZZ}^*(\theta)|^{-\frac{n}{2}} |\lambda T \Sigma_\epsilon^*(\theta)|^{-\frac{\lambda T-k}{2}}} \\ &\times \frac{(2\pi)^{-nT/2} 2^{\frac{n(\lambda+1)T-k}{2}} \prod_{i=1}^n \Gamma[((\lambda+1)T - k + 1 - i)/2]}{2^{\frac{n(\lambda T-k)}{2}} \prod_{i=1}^n \Gamma[(\lambda T - k + 1 - i)/2]}\end{aligned}$$

where $k = n \times p$. Other than using a different likelihood function $p(Y | \theta)$, the RWM algorithm for θ remains the same.

10.2. Estimate DSGE-VAR (DSGEVAR)

```
DSGEVAR(dsgeData, chains=1, cores=1, lambda=Inf, p=2,
        constant=FALSE, ObserveMat, initialvals, partomats,
        priorform, priorpars, parbounds, parnames=NULL,
        optimMethod="Nelder-Mead",
        optimLower=NULL, optimUpper=NULL,
        optimControl=list(),
        IRFs=TRUE, irf.periods=20, scalepar=1,
        keep=50000, burnin=10000,
        tables=TRUE)
```

- **dsgeData**

A matrix or data frame of size $T \times j$ containing the data series used for estimation.

Note: in order to identify the structural shocks, **there must be the same number of observable series as there are shocks in the DSGE model.**

- **chains**

A positive integer value indicating the number of MCMC chains to run.

- **cores**

A positive integer value indicating the number of CPU cores that should be used for estimation. This number should be less than or equal to the number of chains. **Do not allocate more cores than your computer can safely handle!**

- **lambda**

The proportion of DSGE dummy data to actual data. Acceptable values lie in the interval $[j \times (p + 1)/T, \infty]$.

- **p**

The number of lags to include of each variable. The default value is 2.

- **constant**

A logical statement on whether to include a constant vector in the model. The default is ‘FALSE’, and the alternative is ‘TRUE’.

- **ObserveMat**

The $(m + n + k) \times j$ observable matrix \mathcal{H} that maps the state variables to the observable series in the measurement equation.

- **initialvals**

Initial values to begin the optimization routine.

- **partomats**

This is perhaps the most important function input.

‘partomats’ should be a function that maps the deep parameters of the DSGE model to the matrices of a solution method, and contain: a $k \times k$ matrix labelled ‘shocks’ containing the variances of the structural shocks; a $j \times 1$ matrix labelled ‘MeasCons’ containing any constant terms in the measurement equation; and a $j \times j$ matrix labelled ‘MeasErrs’ containing the variances of the measurement errors..

- **priorform**

The prior distribution of each parameter.

- **priorpars**

The parameters of the prior densities.

For example, if the user selects a Gaussian prior for a parameter, then the first entry will be the mean and the second its variance.

- **parbounds**

The lower- and (where relevant) upper-bounds on the parameter values. ‘NA’ values are permitted.

- **parnames**

A character vector containing labels for the parameters.

- **optimMethod**

The optimization algorithm used to find the posterior mode. The user may select: the “Nelder-Mead” simplex method, which is the default; “BFGS”, a quasi-Newton method; “CG” for a conjugate gradient method; “L-BFGS-B”, a limited-memory BFGS algorithm with box constraints; or “SANN”, a simulated-annealing algorithm.

See `?optim` for more details.

If more than one method is entered, e.g., `c(Nelder-Mead, CG)`, optimization will proceed in a sequential manner, updating the initial values with the result of the previous optimization routine.

- `optimLower`

If `optimMethod="L-BFGS-B"`, this is the lower bound for optimization.

- `optimUpper`

If `optimMethod="L-BFGS-B"`, this is the upper bound for optimization.

- `optimControl`

A control list for optimization. See `?optim` in **R** for more details.

- `IRFs`

Whether to calculate impulse response functions.

- `irf.periods`

If `IRFs=TRUE`, then use this option to set the IRF horizon.

- `scalepar`

The scaling parameter, c , for the MCMC run.

- `keep`

The number of replications to keep. If `keep` is set to zero, the function will end with a normal approximation at the posterior mode.

- `burnin`

The number of sample burn-in points.

- `tables`

Whether to print results of the posterior mode estimation and summary statistics of the MCMC run.

The function returns an object of class 'DSGEVAR', which contains:

- `Parameters`

A matrix with ‘keep × chains’ number of rows that contains the estimated, post sample burn-in parameter draws.

- **Beta**

An array of size $j \cdot p \times m \times (\text{keep} \cdot \text{chains})$ which contains the post burn-in draws of β .

- **Sigma**

An array of size $j \times j \times (\text{keep} \cdot \text{chains})$ which contains the post burn-in draws of Σ_ϵ .

- **DSGEIRFs**

A four-dimensional object of size $\text{irf.periods} \times (m+n+k) \times n \times (\text{keep} \cdot \text{chains})$ containing the impulse response calculations values for the DSGE model. The first m refers to responses to the last m shock.

- **DSGEVARIRFs**

A four-dimensional object of size $\text{irf.periods} \times j \times n \times (\text{keep} \cdot \text{chains})$ containing the impulse response function calculations for the VAR. The last m refers to the structural shock.

- **parMode**

Estimated posterior mode parameter values.

- **ModeHessian**

The Hessian computed at the posterior mode for the transformed parameters.

- **logMargLikelihood**

The log marginal likelihood from a Laplacian approximation at the posterior mode.

- **AcceptanceRate**

The acceptance rate of the chain(s).

- **RootRConvStats**

Gelman’s \sqrt{R} -between-chain convergence statistics for each parameter. A value close 1 would signal convergence.

- **ObserveMat**

The user-supplied \mathcal{H} matrix from the Kalman filter recursion.

- **data**

The data used for estimation.

10.3. Example

To illustrate the DSGE-VAR in practice, we continue with the basic New Keynesian model discussed in section 9.2. We set λ , the relative number of DSGE dummy observations to actual data, to 1, and set the number of lags p to 4. The other inputs remain the same as in the estimation example in section 9.2. Again, the two chains are set to run in parallel by setting cores=2.

```
NKMVAR <- DSGEVAR(dsgedata,chains=2,cores=2,lambda=1,p=4,
                     FALSE,ObserveMat,initialvals,partomats,
                     priorform,priorpars,parbounds,parnames,
                     optimMethod=c("Nelder-Mead","CG"),
                     optimLower=NULL,optimUpper=NULL,
                     optimControl=list(maxit=20000,reltol=(10^(-12))),
                     IRFs=TRUE,irf.periods=5,
                     scalepar=0.28,keep=25000,burnin=25000)
```

Trying to solve the model with your initial values... Done.

Beginning optimization, Tue Jul 8 12:10:28 2014.

Using Optimization Method: Nelder-Mead.

Using Optimization Method: CG. Change in the log posterior: 4.549311.

Optimization over, Tue Jul 8 12:11:00 2014.

Optimizer Convergence Code: 0; successful completion.

Optimizer Iterations:

function gradient

1036 377

Log Marginal Likelihood: -534.7842.

Parameter Estimates and Standard Errors (SE) at the Posterior Mode:

	Estimate	SE
Alpha	0.3315737	0.049878827
Beta	0.9954120	0.004150994
Vartheta	5.0516123	0.993553685
Theta	0.7259380	0.074283768
Eta	1.0012686	0.099407198
Phi	0.9835030	0.300263714
Phi.Pi	1.5028955	0.099196808
Phi.Y	0.6962982	0.099599178
Rho.A	0.9432951	0.032108263
Rho.V	0.7098416	0.083266913
Sigma.T	2.1122325	0.639841680
Sigma.M	0.8941240	0.170289023

Trying to Compute DSGE-VAR Prior at the Posterior Mode... Done.

Beginning DSGE-VAR MCMC run, Tue Jul 8 12:11:00 2014.

MCMC run finished, Tue Jul 8 12:13:14 2014.

Acceptance Rate: Chain 1: 0.37124; Chain 2: 0.36968.

Root-R Chain-Convergence Statistics:

Alpha	Beta	Vartheta	Theta	Eta	Phi	Phi.Pi	Phi.Y	Rho.A
Stat: 0.99999	1.000055	1.000895	0.9999905	0.99999	1.000623	0.99999	1.000055	1.000895
Rho.V	Sigma.T	Sigma.M						
Stat: 0.9999905	0.99999	1.000623						

Parameter Estimates and Standard Errors:

	Posterior.Mode	SE.Mode	Posterior.Mean	SE.Posterior
Alpha	0.3315737	0.049878827	0.3313195	0.050442362
Beta	0.9954120	0.004150994	0.9953925	0.003083174

Vartheta	5.0516123	0.993553685	5.0056614	1.001814050
Theta	0.7259380	0.074283768	0.7190511	0.075485626
Eta	1.0012686	0.099407198	0.9942086	0.100569641
Phi	0.9835030	0.300263714	0.9820292	0.305940901
Phi.Pi	1.5028955	0.099196808	1.5036051	0.099788238
Phi.Y	0.6962982	0.099599178	0.6935541	0.101125593
Rho.A	0.9432951	0.032108263	0.9420066	0.027144457
Rho.V	0.7098416	0.083266913	0.7014396	0.084659980
Sigma.T	2.1122325	0.639841680	2.4533029	1.029426769
Sigma.M	0.8941240	0.170289023	0.9182360	0.223114129

Computing DSGE IRFs now... Done.

Starting DSGE-VAR IRFs, Tue Jul 8 12:13:27 2014.

DSGEVAR IRFs finished, Tue Jul 8 12:13:51 2014.

A check of the parameter values at the posterior mode is given by

```
modecheck(NKMVAR,1000,1,plottransform=FALSE,save=TRUE)
```

which is illustrated in figure 40. The marginal posterior distributions of the DSGE parameters are illustrated in figure 41, with input syntax:

```
plot(NKMVAR,save=TRUE)
```

Impulse response functions are called in a similar way to estimated DSGE models. For comparison, the function will plot IRFs for both the DSGE model and the DSGE-VAR (purple and solid lines for the former, green and dashed for the latter).

```
varnames <- names(USMacroData)[c(2,4)]
IRF(NKMVAR,varnames=varnames,save=TRUE)
```

(See figures 42 & 43.) Finally, forecasts of the observable series can be generated using

```
forecast(NKMVAR,5,backdata=5,save=TRUE)
```

which is illustrated in figure 44.

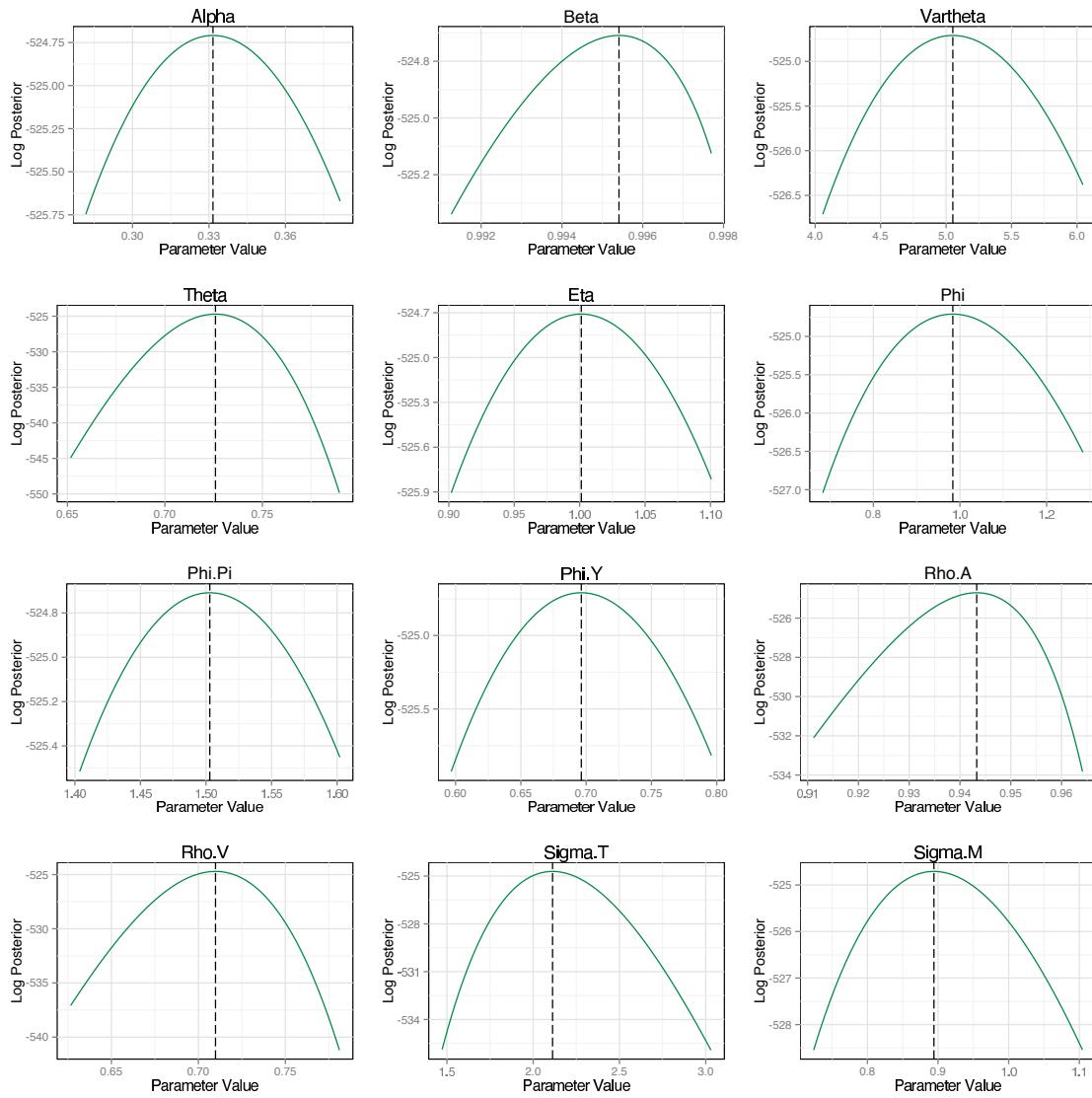
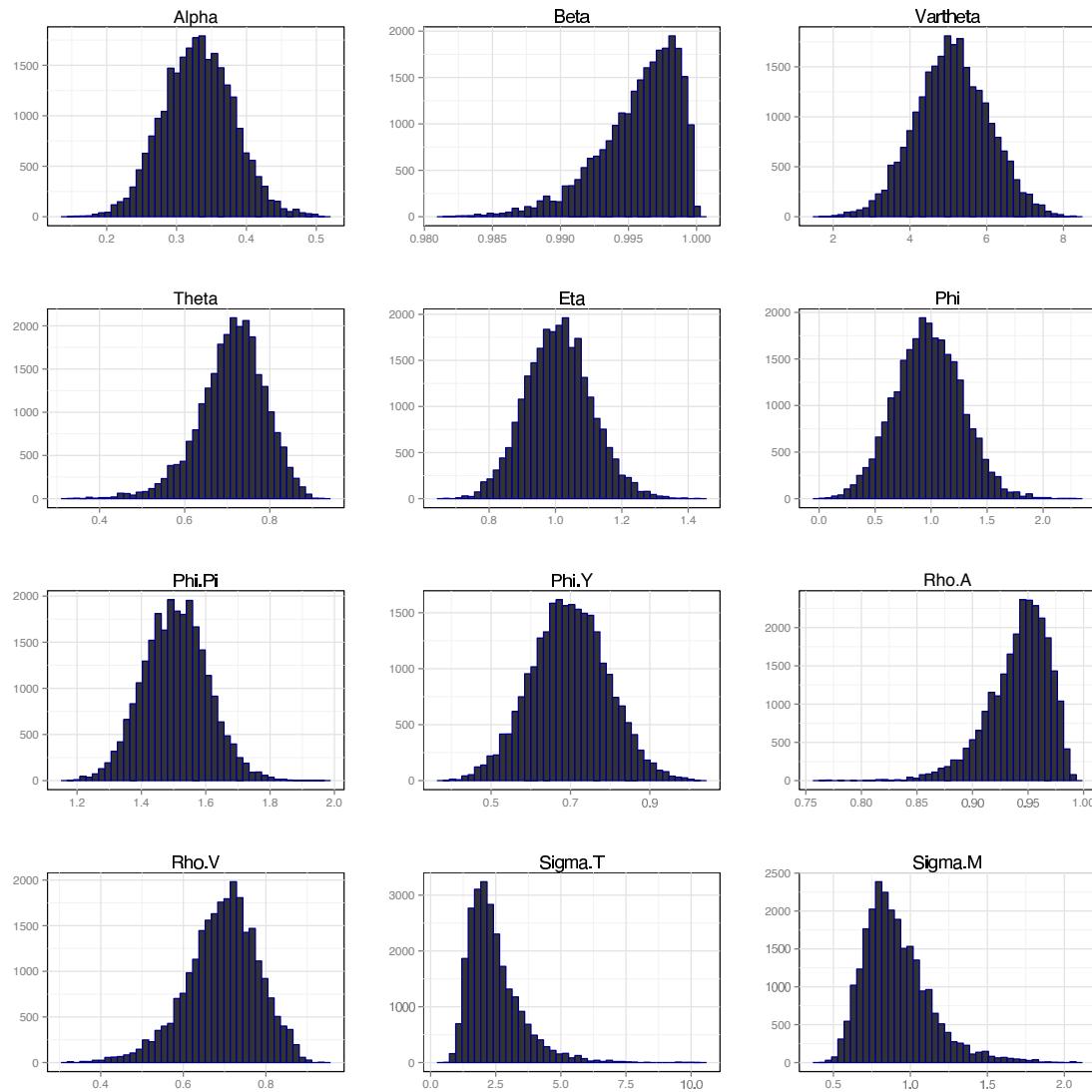


Figure 40: Plot of the Log-Posterior at the Posterior Mode.

Figure 41: Posterior distributions of θ .

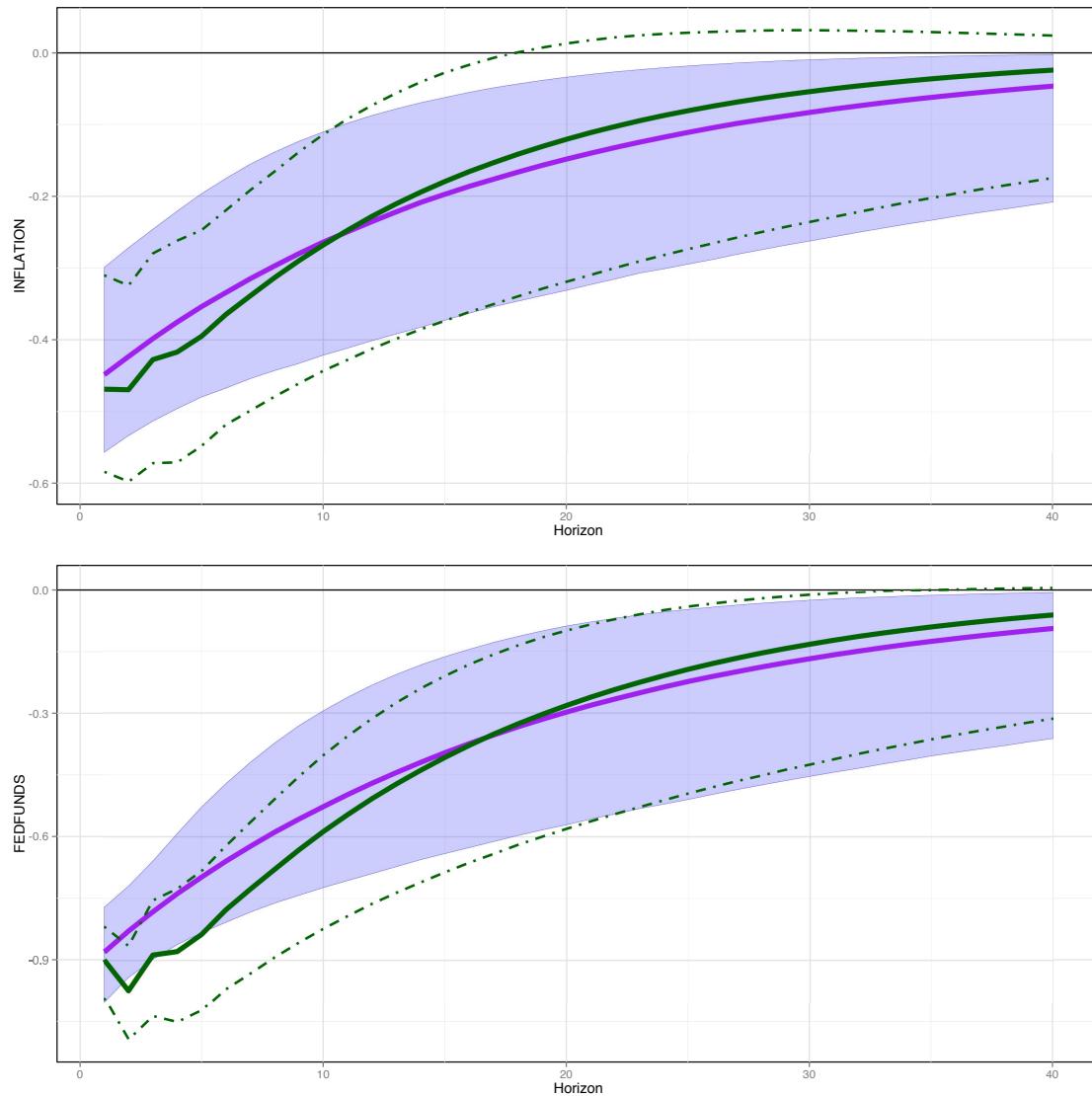


Figure 42: Shock to Technology.

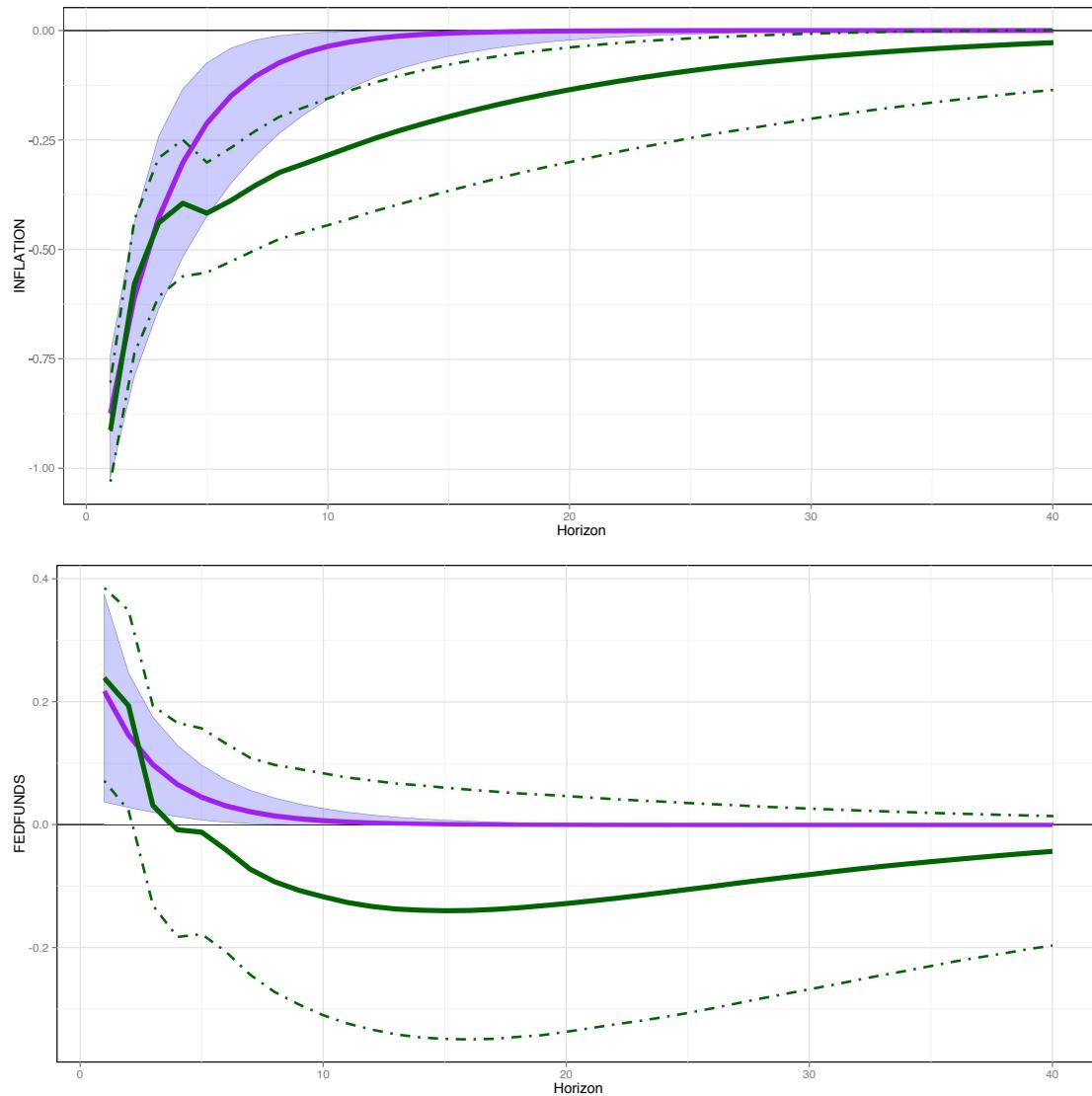


Figure 43: Shock to Monetary Policy.

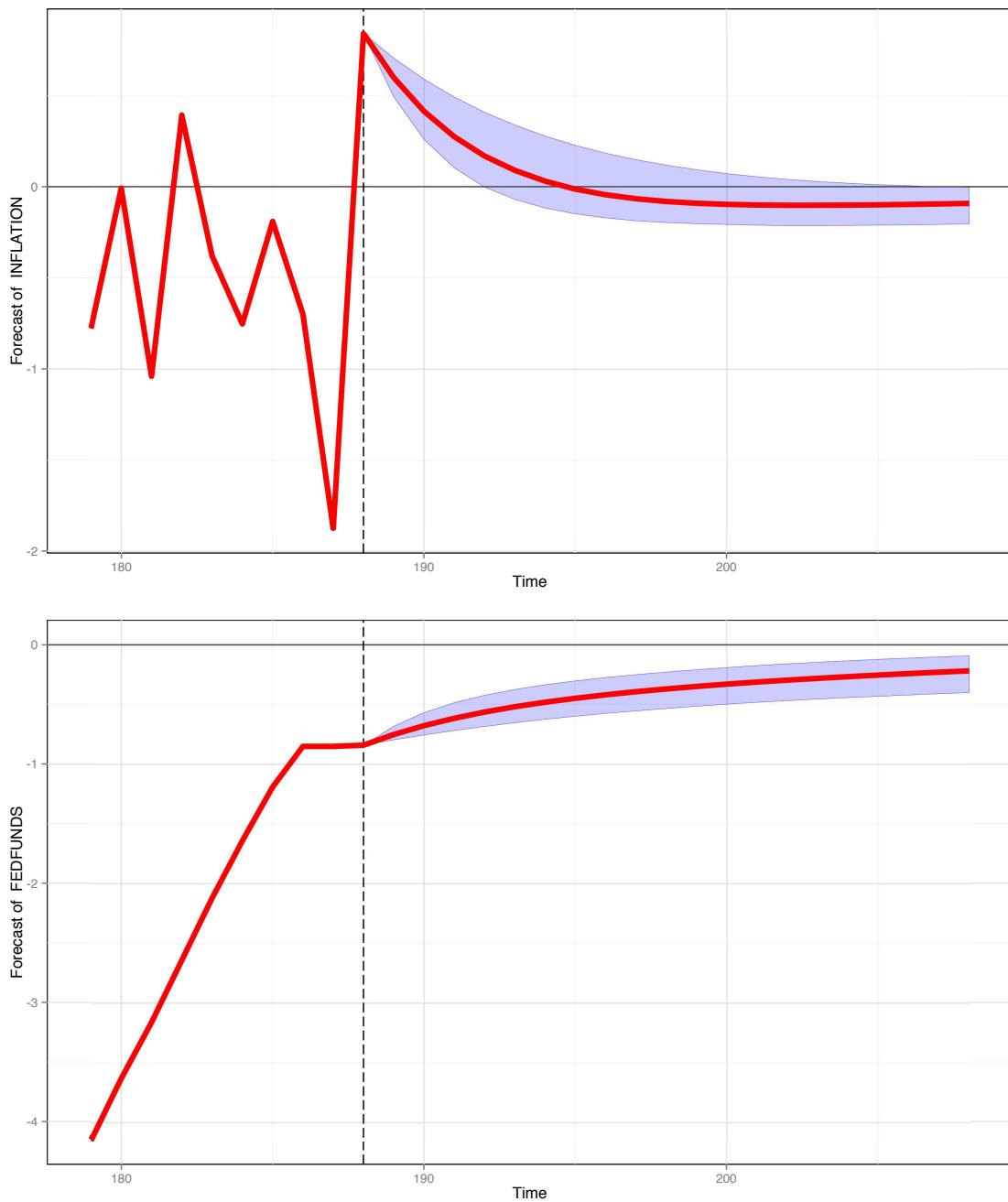


Figure 44: Forecast.

References

- Adjemian, Stéphane, Houtan Bastani, Michel Juillard, Ferhat Mihoubi, George Perendia, Marco Ratto, and Sébastien Villemot**, “Dynare: Reference Manual, Version 4,” *Dynare Working Papers, CEPREMAP*, 2011, 1.
- An, Sungbae and Frank Schorfheide**, “Bayesian Analysis of DSGE Models,” *Econometric Reviews*, 2007, 26 (2–4), 113–172.
- Anderson, Gary S.**, “Solving Linear Rational Expectations Models: A Horse Race,” *Computational Economics*, 2008, 31 (2), 95–113.
- Arulampalam, M. Sanjeev, Simon Maskell, Neil Gordon, and Tim Clapp**, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, 2002, 50 (2), 174–188.
- Aruoba, S.B., Jesús Fernández-Villaverde, and Juan Rubio-Ramírez**, “Comparing Solution Methods for Dynamic Equilibrium Economies,” *Journal of Economic Dynamics and Control*, 2006, 30 (12), 2477–2508.
- Calvo, Guillermo A.**, “Staggered Prices in a Utility-Maximising Framework,” *Journal of Monetary Economics*, 1983, 12 (3), 383–398.
- Canova, Fabio**, *Methods for Applied Macroeconomic Research*, Princeton, New Jersey: Princeton University Press, 2007.
- Carter, Chris K. and Robert Kohn**, “On Gibbs Sampling for State Space Models,” *Biometrika*, 1994, 81 (3), 541–553.
- Chib, Siddhartha and Edward Greenberg**, “Understanding the Metropolis-Hastings Algorithm,” *The American Statistician*, 1995, 49 (4), 327–335.
- Christiano, Lawrence J., Charles L. Evans, and Martin Eichenbaum**, “Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy,” *Journal of Political Economy*, 2005, 113 (1), 1–45.
- Christoffel, Kai, Günter Coenen, and Anders Warne**, “The New Area-Wide Model of the Euro Area: a Micro-founded Open-economy Model for Forecasting and Policy Analysis,” *ECB Working Paper*, 2008, (994).

- , —, and —, “Forecasting with DSGE Models,” *ECB Working Paper*, 2010, (1185).
- Cogley, Timothy and Thomas J. Sargent**, “Drift and Volatilities: Monetary Policies and Outcomes in the Post WWII U.S.” *Review of Economic Dynamics*, 2005, 8 (2), 262–302.
- Dave, Chetan and David N. DeJong**, *Structural Macroeconometrics*, Princeton, New Jersey: Princeton University Press, 2007.
- Del-Negro, Marco and Frank Schorfheide**, “Priors from General Equilibrium Models for VARs,” *International Economic Review*, 2004, 45 (2), 643–673.
- and —, “Bayesian Macroeconomics,” in “Handbook of Bayesian Econometrics,” Vol. 1 of *Handbook of Bayesian Econometrics*, Oxford University Press, December 2011, chapter 7.
- Doan, Tom, Robert Litterman, and Christopher Sims**, “Forecasting and Conditional Projection Using Realistic Prior Distributions,” *NBER Working Paper*, 1983, (1202).
- Durbin, James and Siem Jan Koopman**, “A Simple and Efficient Simulation Smoother for State Space Time Series Analysis,” *Biometrika*, 2002, 89 (3), 603–615.
- Edge, Rochelle M., Michael T. Kiley, and Jean-Philippe Laforte**, “A Comparison of Forecast Performance Between Federal Reserve Staff Forecasts, Simple Reduced-Form Models, and a DSGE Model,” *Journal OF Applied Econometrics*, 2010, 25 (4), 720–754.
- Fernández-Villaverde, Jesús**, “The Econometrics of DSGE Models,” *SERIES*, 2010, 1 (1), 3–49.
- and **Juan Rubio-Ramírez**, “A Baseline DSGE Model,” *Mimeo*, 2006.
- , **Pablo Guerrón-Quintana, and Juan Rubio-Ramírez**, “The New Macroeconomics: A Bayesian Approach,” *The Oxford Handbook of Applied Bayesian Analysis*, 2009.
- Francois, Romain, Dirk Eddelbuettel, and Doug Bates**, *RcppArmadillo: Rcpp integration for Armadillo templated linear algebra library* 2012.
- Galí, Jordi**, *Monetary Policy, Inflation, and the Business Cycle*, Princeton, New Jersey: Princeton University Press, 2008.
- and **Tommaso Monacelli**, “Monetary Policy and Exchange Rate Volatility in a Small Open Economy,” *Review of Economic Studies*, 2005, 72 (3), 707–734.

Geweke, John, *Contemporary Bayesian Econometrics and Statistics*, Hoboken, New Jersey: John Wiley and Sons, 2005.

Hamilton, James D., *Time Series Analysis*, Princeton, New Jersey: Princeton University Press, 1994.

Herbst, Edward, “Using the “Chandrasekhar Recursions” for Likelihood Evaluation of DSGE Models,” *Computational Economics*, 2014, April, 1–13.

Hoy, Michael, John Livernois, Chris McKenna, Ray Rees, and Thanasis Stengos, *Mathematics for Economics*, second ed., Cambridge, Massachusetts: MIT Press, 2001.

Ireland, Peter, “Technology Shocks in the New Keynesian Model,” *Review of Economics and Statistics*, 2004, 86 (4), 923–936.

Klein, Paul, “Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model,” *Journal of Economic Dynamics and Control*, 2000, 24 (10), 1405–1423.

Kolasa, Marcin, Michal Rubaszek, and Paweł Skrzypczynski, “Putting the New Keynesian DSGE Model to the Real-Time Forecasting Test,” *ECB Working Paper*, 2009, (1110).

Koop, Gary and Dimitris Korobilis, “Bayesian Multivariate Time Series Methods for Empirical Macroeconomics,” *Mimeo*, 2010.

Kwiatkowski, Denis, Peter C.B. Phillips, Peter Schmidt, and Yongcheol Shin, “Testing the Null Hypothesis of Stationarity Against the Alternative of a Unit Root,” *Journal of Econometrics*, 1992, 54 (1), 159–178.

Lawless, Martina and Karl Whelan, “Understanding the Dynamics of Labor Shares and Inflation,” *Journal of Macroeconomics*, 2011, 33 (2), 121–136.

Lees, Kirdan, Troy Matheson, and Christie Smith, “Open Economy Forecasting with a DSGE-VAR: Head to Head with the RBNZ Published Forecasts,” *International Journal of Forecasting*, 2011, 27 (2), 512–528.

Litterman, Robert, “Forecasting with Bayesian Vector Autoregressions: Five Years of Experience,” *Journal of Business & Economic Statistics*, 1986, 4 (1), 25–38.

- Lubik, Thomas and Frank Schorfheide**, “Do central banks respond to exchange rate movements? A structural investigation,” *Journal of Monetary Economics*, 2007, 54 (4), 1069–1087.
- McCandless, George**, *The ABCs of RBCs*, Cambridge, Massachusetts: Harvard University Press, 2008.
- Papoulis, Athanasios and S. Unnikrishna Pillai**, *Probability, Random Variables and Stochastic Processes*, fourth ed., New York, N.Y. 10020: McGraw-Hill, 2002.
- Primiceri, Giorgio E.**, “Time Varying Structural Vector Autoregressions and Monetary Policy,” *The Review of Economic Studies*, 2005, 72 (3), 821–851.
- Rotemberg, Julio and Michael Woodford**, “An Optimisation-based Econometric Framework for the Evaluation of Monetary Policy,” *NBER Macroeconomics Annual*, 1997, 12.
- Schorfheide, Frank**, “Estimation and Evaluation of DSGE Models: Progress and Challenges,” *Federal Reserve Bank of Philadelphia, Working Paper No. 11-7*, 2011.
- Smets, Frank and Rafael Wouters**, “An Estimated Stochastic Dynamic General Equilibrium Model of the Euro Area,” *ECB Working Paper*, 2002, (171).
- and —, “Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach,” *American Economic Review*, 2007, 97 (3), 586–606.
- Stock, James H. and Mark W. Watson**, “Vector Autoregressions,” *Journal of Economic Perspectives*, 2001, 15 (4), 101–115.
- Uhlig, Harald**, *A Toolkit for Analysing Nonlinear Dynamic Stochastic Models Easily* Computational Methods for the Study of Dynamic Economics, Oxford University Press, 1999.
- Villani, Mattias**, “Steady-State Priors for Vector Autoregressions,” *Journal of Applied Econometrics*, 2009, 24 (4), 630–650.
- Warne, Anders**, “YADA: Computational Details,” *Mimeo*, 2012.
- Wickham, Hadley**, *ggplot2: elegant graphics for data analysis*, Springer New York, 2009.
- Woodford, Michael**, *Interest and Prices: Foundations of a Theory of Monetary Policy*, Princeton, New Jersey: Princeton University Press, 2003.

A Calculating Impulse Response Functions for BVARs

BVAR-related Impulse response calculations in BMR use a Choleski decomposition to identify the contemporaneous nature of structural shocks. The Choleski decomposition for a symmetric positive-definite matrix $\Sigma \in \mathbb{R}^{d \times d}$ is $\Sigma = LL^\top$, where L is lower-triangular, a particular case of a LU decomposition.¹² The ordering of variables is important for identification, so take necessary steps to justify your ordering, and try different combinations! One exception to note is the Minnesota prior, where, by construction, Σ is diagonal.

For illustrative purposes, let's use a bi-variate VAR(2) model, with coefficient matrices as given in the examples section of the BVAR models; I rewrite them here for convenience:

$$\beta = \begin{bmatrix} 0.5 & 0.2 \\ 0.28 & 0.7 \\ -0.39 & -0.1 \\ 0.1 & 0.05 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

When Σ is diagonal, $L_{i,i} = \sqrt{\Sigma_{i,i}}$, and as $\Sigma = \mathbb{I}_2$, we have $\Sigma = L = L^\top$. Define $\Psi(\ell)$ as a $m \times m$ matrix containing the response of our variables to shocks in Σ . Let $E(\ell)$ be a $p \cdot m \times m$ matrix containing the last p IRF matrices, $\Psi(\ell-1), \Psi(\ell-2), \dots, \Psi(\ell-p)$, stacked with the $\Psi(\ell-1)$ being the first block and $\Psi(\ell-p)$ being last.

To begin, $\Psi(1) = L$ and $E(1) = \mathbf{0}_{4 \times 2}$. To find $\Psi(2)$, first build $E(2)$, which is

$$E(2) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

¹²For a non-singular Hermitian matrix $\Sigma \in \mathbb{C}^{d \times d}$, we can factorise as $\Sigma = LU$, where L and U are lower- and upper-triangular $d \times d$ matrices, respectively

To find $\Psi(2)$, we premultiply $E(2)$ by β^\top ,

$$\Psi(2) = \begin{bmatrix} 0.5 & 0.28 & -0.39 & 0.1 \\ 0.2 & 0.7 & -0.1 & 0.5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.28 \\ 0.2 & 0.7 \end{bmatrix}$$

$E(3)$ is now

$$E(3) = \begin{bmatrix} 0.5 & 0.28 \\ 0.2 & 0.7 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

To find $\Psi(3)$, we premultiply $E(3)$ by β^\top , just as before,

$$\Psi(3) = \begin{bmatrix} 0.5 & 0.28 & -0.39 & 0.1 \\ 0.2 & 0.7 & -0.1 & 0.5 \end{bmatrix} \begin{bmatrix} 0.5 & 0.28 \\ 0.2 & 0.7 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.084 & 0.436 \\ 0.140 & 0.596 \end{bmatrix}$$

And so on, for as many points as the user desires. We interpret the elements of $\Psi(\ell)$ as follows: the columns are the ‘shocks from’ and the rows make up the ‘responses to’ part of IRFs.

For example, if our data are quarterly, and we are looking at the $\Psi(3)$ matrix: the element $\Psi_{1,1} = -0.084$ gives the response of the first variable to a one standard deviation shock to the first variable after 2 quarters; the element $\Psi_{1,2} = 0.436$ is the response of the first variable to a one standard deviation shock to the second variable after 2 quarters; the element $\Psi_{2,1} = 0.140$ is the response of the second variable to a one standard deviation shock to the first variable after 2 quarters; and the element $\Psi_{2,2} = 0.596$ is the response of the second variable to a one

standard deviation shock to the second variable after 2 quarters.

B Specifying Full Matrix Priors in BMR

This section details how to specify full matrix priors on both the mean and covariance matrices.

B1. Prior Mean of the Coefficients

For the BVARM, BVARS, and BVARW functions, we can simplify the process of selecting a prior mean of each coefficient by only looking at the first own-lags, and setting all other coefficients to zero. However, the user may prefer to, say, place a non-zero prior on the intercepts.

For a bi-variate VAR(2) model with a constant Φ , the β matrix is formed as follows,

$$\beta = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \beta_1^{(1,1)} & \beta_1^{(1,2)} \\ \beta_1^{(2,1)} & \beta_1^{(2,2)} \\ \beta_2^{(1,1)} & \beta_2^{(1,2)} \\ \beta_2^{(2,1)} & \beta_2^{(2,2)} \end{bmatrix}$$

where $\beta_p^{(i,j)}$ is the coefficient corresponding to the effect of variable i on variable j for the p th lag, e.g.,

$$\begin{aligned} Y_{1,t} &= \Phi_1 + \beta_1^{(1,1)} \cdot Y_{1,t-1} + \beta_1^{(2,1)} \cdot Y_{2,t-1} + \beta_2^{(1,1)} \cdot Y_{1,t-2} + \beta_2^{(2,1)} \cdot Y_{2,t-2} + \epsilon_{1,t} \\ Y_{2,t} &= \Phi_2 + \beta_1^{(1,2)} \cdot Y_{1,t-1} + \beta_1^{(2,2)} \cdot Y_{2,t-1} + \beta_2^{(1,2)} \cdot Y_{1,t-2} + \beta_2^{(2,2)} \cdot Y_{2,t-2} + \epsilon_{2,t} \end{aligned}$$

For example, if we wished to specify a mean of 3 on both the constant terms Φ , a coefficient of 0.7 on the first own-lag of Y_1 and a coefficient of 0.4 on the first own-lag of Y_2 with everything else as zero, the code is

```
coefprior <- rbind(c( 3,  3),
                     c(0.7,  0),
                     c( 0,0.4),
                     c( 0,  0),
                     c( 0,  0))
```

Then we can use this for estimation with, say, a Minnesota prior:

```
testbvarm <- BVARM(bvardata,coefprior,p=2,constant=TRUE,irf.periods=20,  
keep=10000,burnin=5000,VType=1,  
HP1=0.5,HP2=0.5,HP3=4)
```

Remember, if you're doing something similar with BVARS (steady-state prior), there are no constants Φ , as we're modelling the unconditional mean (Ψ) instead.

B2. Prior Covariance Matrix

For those who wish to specify a full prior on the covariance matrix of β , or, for example, the location matrix of Σ , BMR allows the user to do so. Ξ_β is based on the vectorised version of β , where, using our previous VAR(2) example,

$$\text{vec}(\beta) = \begin{bmatrix} \Phi_1 & \beta_1^{(1,1)} & \beta_1^{(2,1)} & \beta_2^{(1,1)} & \beta_2^{(2,1)} & \Phi_2 & \beta_1^{(1,2)} & \beta_1^{(2,2)} & \beta_2^{(1,2)} & \beta_2^{(2,2)} \end{bmatrix}^\top$$

and, for simplicity, a diagonal Ξ_β is

$$\begin{bmatrix} V(\Phi_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & V(\beta_1^{(1,1)}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & V(\beta_1^{(2,1)}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & V(\beta_2^{(1,1)}) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & V(\beta_2^{(2,1)}) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & V(\Phi_2) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & V(\beta_1^{(1,2)}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & V(\beta_1^{(2,2)}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & V(\beta_2^{(1,2)}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & V(\beta_2^{(2,2)}) \end{bmatrix}$$

This is the required format for specifying a full Ξ prior. If the user also opts to specify covariance terms, ensure that the matrix is symmetric.

C Class-related Functions

Several functions in BMR are class-specific, including the forecast, IRF and plot commands.

C1. Forecast

C1.1. BVARM, BVARS, BVARW, DSGEVAR

```
forecast(obj, periods=20, shocks=TRUE, plot=TRUE,
         percentiles=c(.05, .50, .95), useMean=FALSE,
         backdata=0, save=FALSE, height=13, width=11)
```

- **obj**

An object of class ‘BVARM’, ‘BVARS’, or ‘BVARW’.

- **periods**

The forecast horizon.

- **shocks**

Whether to include uncertainty about future shocks when calculating the distribution of forecasts.

- **plot**

Whether to plot the forecasts.

- **percentiles**

The percentiles of the distribution the user wants to use.

- **useMean**

Whether the user would prefer to use the mean of the forecast distribution rather than the middle value in ‘percentiles’.

- **backdata**

How many ‘real’ data points to plot before plotting the forecast. A broken line will indicate whether the ‘real’ data ends and the forecast begins.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

The function returns a list containing

- **MeanForecast**

The mean values of the forecast.

- **PointForecast**

The values of the point forecast.

- **Forecasts**

An array containing all of the calculated forecasts.

C1.2. CVAR

```
forecast(obj, periods=20, plot=TRUE, confint=0.95,
         backdata=0, save=FALSE, height=13, width=11)
```

- **obj**

An object of class ‘CVAR’.

- **periods**

The forecast horizon.

- **plot**

Whether to plot the forecasts.

- **confint**

The confidence interval to use.

- **backdata**

How many ‘real’ data points to plot before plotting the forecast. A broken line will indicate whether the ‘real’ data ends and the forecast begins.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

The function returns a list containing

- **PointForecast**

The values of the point forecast.

- **Forecasts**

The values of the point forecast and user-selected percentiles.

C1.3. EDSGE

```
forecast(obj, periods=20, shocks=TRUE, plot=TRUE,
         percentiles=c(.05, .50, .95), useMean=FALSE,
         backdata=0, save=FALSE, height=13, width=11)
```

- **obj**

An object of class ‘EDSGE’.

- **periods**

The forecast horizon.

- **plot**

Whether to plot the forecasts.

- **percentiles**

The percentiles of the distribution the user wants to use.

- **useMean**

Whether the user would prefer to use the mean of the forecast distribution rather than the middle value in ‘percentiles’.

- **backdata**

How many 'real' data points to plot before plotting the forecast. A broken line will indicate whether the 'real' data ends and the forecast begins.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

The function returns a list containing

- **MeanForecast**

The mean values of the forecast.

- **PointForecast**

The values of the point forecast.

- **Forecasts**

An array containing all of the calculated forecasts.

C2. IRF

C2.1. BVARM, BVARS, BVARW, CVAR

```
IRF(obj, percentiles=c(.05,.50,.95), save=TRUE, height=13, width=13)
```

- **obj**

An object of class 'BVARM', 'BVARS', 'BVARW', or 'CVAR'.

- **percentiles**

The percentiles of the distribution the user wants to use.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

C2.2. BVARTVP

```
IRF(obj,whichirfs=NULL,percentiles=c(.05,.50,.95),save=TRUE,height=13,width=13)
```

- **obj**

An object of class ‘BVARTVP’.

- **percentiles**

The percentiles of the distribution the user wants to use.

- **whichirfs**

Which IRFs to plot. The default is all of those which the user chose to calculate during estimation.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

C2.3. DSGEVAR

```
IRF(obj,varnames=NULL,percentiles=c(.05,.50,.95),
    save=TRUE,height=13,width=13)
```

C2.4. EDSGE

```
IRF(obj,ObservableIRFs=TRUE,varnames=NULL,percentiles=c(.05,.50,.95),
    save=TRUE,height=13,width=13)
```

- **obj**

An object of class ‘EDSGE’.

- **ObservableIRFs**

Whether to plot the IRFs relating to the state, or the implied IRFs of the observable series. Remember, we compute the ℓ -step ahead IRF for the j -th shock as

$$\mathcal{F}^\ell \mathcal{G} [\mathbf{0}_{1 \times (m+n+j-1)} \ \sigma_j \ \mathbf{0}_{1 \times (j-1)}]^\top$$

where $\mathcal{F}^0 = \mathbb{I}_{(m+n+k)}$. The observable IRFs are then

$$\mathcal{H}^\top \mathcal{F}^\ell \mathcal{G} [\mathbf{0}_{1 \times (m+n+j-1)} \ \sigma_j \ \mathbf{0}_{1 \times (j-1)}]^\top$$

- **varnames**

A character vector with the names of the variables.

- **percentiles**

The percentiles of the distribution the user wants to use.

- **save**

Whether to save the plots.

- **height**

If `save=TRUE`, use this to set the height of the plot.

- **width**

If `save=TRUE`, use this to set the width of the plot.

C2.5. SDSGE

```
IRF(obj, shocks, irf.periods=20, varnames=NULL, plot=TRUE,
  save=FALSE, height=13, width=13)
```

- **obj**

An object of class ‘SDSGE’.

- **shocks**

A numeric vector containing the standard deviations of the shocks.

- **irf.periods**

The horizon of the IRFs.

- **varnames**

A character vector with the names of the variables.

- **plot**

Whether to plot the IRFs.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

C3. Mode Check

C3.1. DSGEVAR, EDSGE

```
modecheck(obj, gridsize=1000, scalepar=NULL, plottransform=FALSE,
          save=FALSE, height=13, width=13)
```

- **obj**

An object of class ‘EDSGE’ or ‘DSGEVAR’.

- **gridsize**

The number of grid points to use when calculating the log posterior around the mode values.

- **scalepar**

A value to replace the scale parameter from estimation (“c”) when plotting the log posterior.

- **plottransform**

Whether to plot the transformed values (*i.e.*, such that the support is unbounded), or to plot the untransformed values.

- **save**

Whether to save the plots.

- **height**

If `save=TRUE`, use this to set the height of the plot.

- **width**

If `save=TRUE`, use this to set the width of the plot.

C4. Plot

C4.1. BVARM, BVARS, BVARW

```
plot(obj, type=1, plotSigma=TRUE, save=TRUE, height=13, width=13)
```

- **obj**

An object of class ‘BVARM’, ‘BVARS’, or ‘BVARW’.

- **type**

An integer value indicating the plot style; `type=1` will produce a histogram, while `type=2` will use smoothed densities.

- **plotSigma**

Whether to plot the covariance terms.

- **save**

Whether to save the plots.

- **height**

If `save=TRUE`, use this to set the height of the plot.

- **width**

If `save=TRUE`, use this to set the width of the plot.

C4.2. BVARTVP

```
plot(obj,percentiles=c(.05,.50,.95),save=FALSE,height=13,width=13)
```

- **obj**

An object of class ‘BVARTVP’.

- **percentiles**

The percentiles of the distribution the user wants to use.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

C4.3. DSGEVAR, EDSGE

```
plot(obj,BinDenom=40,save=FALSE,height=13,width=13)
```

- **obj**

An object of class ‘EDSGE’.

- **binDenom**

The ‘bin’ size of the histograms used to plot the estimated parameters is calculated by dividing the range of values by ‘binDenom’. ggplot2 will use 30 as its default, but BMR sets the default to 40.

- **save**

Whether to save the plots.

- **height**

If save=TRUE, use this to set the height of the plot.

- **width**

If save=TRUE, use this to set the width of the plot.

D Example of a Parameter to Matrices Function (partomats)

The code below, used when estimating the basic New-Keynesian model, illustrates the required format of a ‘partomats’ function the user provides to the EDSGE function in BMR. The user-written function maps the deep parameters of the DSGE model to the 12 matrices of Uhlig’s method, along with a ‘shocks’ matrix that defines the covariance matrix of the exogenous shocks, a vector of intercept terms in the measurement equation, and the covariance matrix of the measurement errors.

```
partomats <- function(parameters){
  alpha      <- parameters[1]
  beta       <- parameters[2]
  vartheta   <- parameters[3]
  theta      <- parameters[4]
  #
  eta        <- parameters[5]
  phi        <- parameters[6]
  phi_pi    <- parameters[7]
  phi_y     <- parameters[8]
  rho_a     <- parameters[9]
  rho_v     <- parameters[10]
  #
  sigmaT    <- (parameters[11])^2
  sigmaM    <- (parameters[12])^2
  #
  BigTheta <- (1-alpha)/(1-alpha+alpha*vartheta)
  kappa    <- (((1-theta)*(1-beta*theta))/(theta))*BigTheta*((1/eta)+((phi+alpha)/(1-alpha)))
  #
  psi      <- (eta*(1+phi))/(1-alpha+eta*(phi + alpha))
  #
  A <- matrix(0,nrow=0,ncol=0)
  B <- matrix(0,nrow=0,ncol=0)
  C <- matrix(0,nrow=0,ncol=0)
```

```

D <- matrix(0,nrow=0,ncol=0)
#
#Order:      yg      y,      pi,      rn,      i,      n
F <- rbind(c(-1, 0, -eta, 0, 0, 0),
           c(0, 0, -beta, 0, 0, 0),
           c(0, 0, 0, 0, 0, 0))
#
G <- rbind(c(1, 0, 0, -1*eta, eta, 0),
           c(-kappa, 0, 1, 0, 0, 0),
           c(-phi_y, 0, -phi_pi, 0, 1, 0),
           c(0, 0, 0, 1, 0, 0),
           c(0, 1, 0, 0, 0, -(1-alpha)),
           c(1, -1, 0, 0, 0, 0))
#
H <- rbind(c(0, 0, 0, 0, 0, 0),
           c(0, 0, 0, 0, 0, 0))
#
J <- matrix(0,nrow=0,ncol=0)
K <- matrix(0,nrow=0,ncol=0)
#
L <- matrix(c( 0, 0,
              0, 0,
              0, 0,
              0, 0,
              0, 0,

```

```

0, 0 ), ncol=2, byrow=T);

#
M41 <- -(1/eta)*psi*(rho_a - 1)

M<- matrix(c( 0, 0,
              0, 0,
              0, -1,
              M41, 0,
              -1, 0,
              psi, 0), ncol=2, byrow=T)

#
N = matrix(c( rho_a, 0,
              0, rho_v), nrow=2)

#
shocks <- matrix(c(sigmaT, 0,
                     0, sigmaM), nrow=2)

#
ObsCons <- matrix(0, nrow=2, ncol=1)
MeasErrs <- matrix(0, nrow=2, ncol=2)

#
return=list(A=A, B=B, C=C, D=D, F=F, G=G, H=H, J=J, K=K, L=L, M=M, N=N, shocks=shocks, ObsCons=ObsCons)
}

```