Bayesian Inference for DSGE Models

Edward Herbst
Frank Schorfheide

September 24, 2014

Contents

Abbreviations

CLT: central limit theorem

DSGE: dynamic stochastic general equilibrium

HAC: heteroskedasticity and autocorrelation consistent covariance

LLN: law of large numbers

MAL: Metropolis-adjusted Langevin

MCMC: Markov chain Monte Carlo

MC: Monte Carlo

MH: Metropolis-Hastings

RWMH: random walk Metropolis-Hastings

SLLN: strong law of large numbers

VAR: vector autoregression

Additional Terminology and Hyphenation

iid-equivalent

Mathematical Notation

- $\mathbb{E}_\pi[h]$ is posterior mean
- $\mathbb{V}_\pi[h]$ is posterior variance
- $\bar{V}(\bar{h})$ is asymptotic variance of $\bar{h}_N$
- $HAC(\bar{h})$ is HAC estimator of $\bar{V}(\bar{h})/N$.
- $\hat{V}(\bar{h})$ is small sample variance of $\bar{h}$. Note that $\hat{V}(\bar{h}) \approx \bar{V}(\bar{h})/N$.
- Also, we defined $ESS = N\bar{V}(\bar{h})/\mathbb{V}_\pi[h]$.

Preface

The first papers that used Bayesian techniques to estimate dynamic stochastic general equilibrium (DSGE) models were published about fifteen years ago: DeJong, Ingram, and Whiteman (2000), Schorfheide (2000), and Otrok (2001). The DSGE models at the time were relatively small in terms of the number of parameters and hidden states, and were estimated with, by today's standards, fairly simple versions of Metropolis-Hastings (MH) or importance sampling algorithms. Since then, DSGE models have grown in size, in particular the ones that are used by central banks for prediction and policy analysis. The celebrated Smets and Wouters (2003, 2007) has more than a dozen hidden states and thirty-six estimated parameters. The Smets-Wouters model forms the core of the latest vintage of DSGE models which may add a housing sector, search frictions in the labor market, or a banking sector and financial frictions to the basic set of equations. Each of these mechanisms increases the state space and the parameter space of the DSGE model.

The goal of this book is to assess the accuracy of the "standard" Bayesian computational techniques that have been applied in the DSGE model literature over the past fifteen years and to introduce and explore "new" computational tools that improve the accuracy of Monte Carlo approximations of posterior distributions associated with DSGE models. The reader will quickly notice that the tools are not really new (which is why we used quotation marks): they are imported from the engineering and statistical literature and tailored toward DSGE model applications. The book is based on a series of lectures on *Recent Theory and Applications of DSGE Models* which were presented as the Tinbergen Institute Econometrics Lectures at the Erasmus University Rotterdam in June 2012, but the material has evolved significantly since then.

The book is consists of three parts. The first part consists of an introduction to DSGE modeling and Bayesian inference. We present a small-scale New Keynesian model, show how it can be solved and turned into a state-space model that is amenable to Bayesian estimation. We also provide a primer on Bayesian inference for readers unfamiliar with Bayesian

econometrics. While this primer is not a substitute for a thorough textbook treatment, it tries to explain the key ideas in the context of a linear Gaussian regression model. Moreover, we provide an introduction to important computational techniques: direct sampling, importance sampling, and Metropolis-Hastings algorithms.

The second part of the book is devoted to Bayesian computations for linearized DSGE models with Gaussian shocks. Thus, we focus on models for which the likelihood function can be evaluated with the Kalman filter. Starting point is the Random-Walk MH algorithm, which is the most widely-used algorithm for Bayesian estimation of DSGE models in the literature. We discuss several refinements to this algorithm before proceeding with Sequential Monte Carlo (SMC) methods. While popular in the statistical literature, there are hardly any applications to the estimation of DSGE models. We provide a detailed discussion of how to tune these algorithms for DSGE model applications and examine their accuracy. The performance of MH and SMC algorithms is compared in three empirical applications.

The last part of the book focuses on computations for DSGE models solved with nonlinear techniques. The main difference is that the likelihood function can no longer be evaluated with the Kalman filter. It requires a sequential Monte Carlo filter (also called particle filter), instead. To avoid any disappointments, we hasten to point out that we are actually not estimating any nonlinear DSGE models in this book. Instead, we are using various versions of particle filters to evaluate the likelihood function of linear DSGE models. This has the advantage that we can easily compare results from procedures that utilize particle filters to procedures that use the exact Kalman filter and thereby assess the accuracy of the particle filter approximation. We begin with likelihood evaluations conditional on a fixed parameter vector and subsequently embed the particle filter approximations of the likelihood function into MH and SMC algorithms to conduct posterior inference for the DSGE model parameters.

Edward Herbst
Frank Schorfheide
Whenever, 2014

Part I

Introduction to DSGE Modeling and Bayesian Inference

Chapter 1

DSGE Modeling

Estimated dynamic stochastic general equilibrium (DSGE) models are now widely-used by academics to conduct empirical research macroeconomics as well as by central banks to interpret the current state of the economy, analyze the impact of changes in monetary or fiscal policy, and to generate predictions for key macroeconomic aggregates. The term DSGE model encompasses a broad class of macroeconomic models that span the real business cycle models of Kydland and Prescott (1982) and King, Plosser, and Rebelo (1988) as well as the New Keynesian models of Rotemberg and Woodford (1997) or Christiano, Eichenbaum, and Evans (2005), which feature nominal price and wage rigidities and a role for central banks to adjust interest rates in response to inflation and output fluctuations. A common feature of these models is that decision rules of economic agents are derived from assumptions about preferences and technologies by solving intertemporal optimization problems. Moreover, agents potentially face uncertainty with respect to total factor productivity, for instance, or the nominal interest rate set by a central bank. This uncertainty is generated by exogenous stochastic processes that shift technology, for example, or generate unanticipated deviations from a central bank's interest-rate feedback rule.

The focus of this book is the Bayesian estimation of DSGE models. Conditional on distributional assumptions for the exogenous shocks, the DSGE model generates a joint probability distribution for the endogenous model variables such as output, consumption, investment, and inflation. In a Bayesian framework, this likelihood function can be used to transform a prior distribution for the structural parameters of the DSGE model into a posterior distribution. This posterior is the basis for substantive inference and decision making. Unfortunately, it is not feasible to characterize moments and quantiles of the posterior distribution analyti-

cally. Instead, we have to use computational techniques to generate draws from the posterior and then approximate posterior expectations by Monte Carlo averages.

In Section 1.1 we will present a small-scale New Keynesian DSGE model and describe the decision problems of firms and households and the behavior of the monetary and fiscal authorities. We then characterize the resulting equilibrium conditions. This model is subsequently used in many of the numerical illustrations. Section 1.2 briefly sketches two other DSGE models that will be estimated in subsequent chapters.

1.1 A Small-Scale New Keynesian DSGE Model

We begin with a small-scale New Keynesian DSGE model that has been widely studied in the literature (see Woodford (2003) or Gali (2008) for textbook treatments). The particular specification presented below is based on An and Schorfheide (2007a). The likelihood function for a linearized version of this model can be quickly evaluated, which makes the model an excellent showcase for the computational algorithms studied below.

1.1.1 Firms

The perfectly competitive, representative, final good producing firm combines a continuum of intermediate goods indexed by $j \in [0,1]$ using the technology

$$Y_t = \left(\int_0^1 Y_t(j)^{1-\nu} dj \right)^{\frac{1}{1-\nu}}. \tag{1.1}$$

Here $1/\nu > 1$ represents the elasticity of demand for each intermediate good. The firm takes input prices $P_t(j)$ and output prices P_t as given. Profit maximization implies that the demand for intermediate goods is

$$Y_t(j) = \left(\frac{P_t(j)}{P_t} \right)^{-1/\nu} Y_t. \tag{1.2}$$

The relationship between intermediate goods prices and the price of the final good is

$$P_t = \left(\int_0^1 P_t(j)^{\frac{\nu-1}{\nu}} dj \right)^{\frac{\nu}{\nu-1}}. \tag{1.3}$$

Intermediate good j is produced by a monopolist who has access to the following linear production technology:

$$Y_t(j) = A_t N_t(j), \tag{1.4}$$

where A_t is an exogenous productivity process that is common to all firms and $N_t(j)$ is the labor input of firm j. Labor is hired in a perfectly competitive factor market at the real wage W_t. Firms face nominal rigidities in terms of quadratic price adjustment costs

$$AC_t(j) = \frac{\phi}{2}\left(\frac{P_t(j)}{P_{t-1}(j)} - \pi\right)^2 Y_t(j), \tag{1.5}$$

where ϕ governs the price stickiness in the economy and π is the steady state inflation rate associated with the final good. Firm j chooses its labor input $N_t(j)$ and the price $P_t(j)$ to maximize the present value of future profits

$$\mathbb{E}_t\left[\sum_{s=0}^{\infty}\beta^s Q_{t+s|t}\left(\frac{P_{t+s}(j)}{P_{t+s}}Y_{t+s}(j) - W_{t+s}N_{t+s}(j) - AC_{t+s}(j)\right)\right]. \tag{1.6}$$

Here, $Q_{t+s|t}$ is the time t value of a unit of the consumption good in period $t+s$ to the household, which is treated as exogenous by the firm.

1.1.2 Households

The representative household derives utility from real money balances M_t/P_t and consumption C_t relative to a habit stock. We assume that the habit stock is given by the level of technology A_t. This assumption ensures that the economy evolves along a balanced growth path even if the utility function is additively separable in consumption, real money balances, and leisure. The household derives disutility from hours worked H_t and maximizes

$$\mathbb{E}_t\left[\sum_{s=0}^{\infty}\beta^s\left(\frac{(C_{t+s}/A_{t+s})^{1-\tau}-1}{1-\tau} + \chi_M \ln\left(\frac{M_{t+s}}{P_{t+s}}\right) - \chi_H H_{t+s}\right)\right], \tag{1.7}$$

where β is the discount factor, $1/\tau$ is the intertemporal elasticity of substitution, and χ_M and χ_H are scale factors that determine steady state real money balances and hours worked. We will set $\chi_H = 1$. The household supplies perfectly elastic labor services to the firms taking the real wage W_t as given. The household has access to a domestic bond market where nominal government bonds B_t are traded that pay (gross) interest R_t. Furthermore, it receives aggregate residual real profits D_t from the firms and has to pay lump-sum taxes T_t. Thus, the household's budget constraint is of the form

$$P_tC_t + B_t + M_t - M_{t-1} + T_t = P_tW_tH_t + R_{t-1}B_{t-1} + P_tD_t + P_tSC_t, \tag{1.8}$$

where SC_t is the net cash inflow from trading a full set of state-contingent securities. The usual transversality condition on asset accumulation applies, which rules out Ponzi schemes.

1.1.3 Monetary and Fiscal Policy

Monetary policy is described by an interest rate feedback rule of the form

$$R_t = R_t^{*\,1-\rho_R} R_{t-1}^{\rho_R} e^{\epsilon_{R,t}}, \tag{1.9}$$

where $\epsilon_{R,t}$ is a monetary policy shock and R_t^* is the (nominal) target rate:

$$R_t^* = r\pi^* \left(\frac{\pi_t}{\pi^*}\right)^{\psi_1} \left(\frac{Y_t}{Y_t^*}\right)^{\psi_2}. \tag{1.10}$$

Here r is the steady state real interest rate, π_t is the gross inflation rate defined as $\pi_t = P_t/P_{t-1}$, and π^* is the target inflation rate, which in equilibrium coincides with the steady state inflation rate. Y_t^* in (1.10) is the level of output that would prevail in the absence of nominal rigidities.

The fiscal authority consumes a fraction ζ_t of aggregate output Y_t, where $\zeta_t \in [0, 1]$ follows an exogenous process. The government levies a lump-sum tax (subsidy) to finance any shortfalls in government revenues (or to rebate any surplus). The government's budget constraint is given by

$$P_t G_t + R_{t-1} B_{t-1} = T_t + B_t + M_t - M_{t-1}, \tag{1.11}$$

where $G_t = \zeta_t Y_t$.

1.1.4 Exogenous Processes

The model economy is perturbed by three exogenous processes. Aggregate productivity evolves according to

$$\ln A_t = \ln \gamma + \ln A_{t-1} + \ln z_t, \quad \text{where} \quad \ln z_t = \rho_z \ln z_{t-1} + \epsilon_{z,t}. \tag{1.12}$$

Thus, on average technology grows at the rate γ and z_t captures exogenous fluctuations of the technology growth rate. Define $g_t = 1/(1 - \zeta_t)$. We assume that

$$\ln g_t = (1 - \rho_g) \ln g + \rho_g \ln g_{t-1} + \epsilon_{g,t}. \tag{1.13}$$

Finally, the monetary policy shock $\epsilon_{R,t}$ is assumed to be serially uncorrelated. The three innovations are independent of each other at all leads and lags and are normally distributed with means zero and standard deviations σ_z, σ_g, and σ_R, respectively.

1.1.5 Equilibrium Relationships

We consider the symmetric equilibrium in which all intermediate goods producing firms make identical choices so that the j subscript can be omitted. The market clearing conditions are given by

$$Y_t = C_t + G_t + AC_t \quad \text{and} \quad H_t = N_t. \tag{1.14}$$

Since the households have access to a full set of state-contingent claims, $Q_{t+s|t}$ in (1.6) is

$$Q_{t+s|t} = (C_{t+s}/C_t)^{-\tau}(A_t/A_{t+s})^{1-\tau}. \tag{1.15}$$

It can be shown that output, consumption, interest rates, and inflation have to satisfy the following optimality conditions

$$1 = \beta \mathbb{E}_t \left[\left(\frac{C_{t+1}/A_{t+1}}{C_t/A_t} \right)^{-\tau} \frac{A_t}{A_{t+1}} \frac{R_t}{\pi_{t+1}} \right] \tag{1.16}$$

$$\begin{aligned} 1 = {} & \frac{1}{\nu}\left[1 - \left(\frac{C_t}{A_t}\right)^{\tau}\right] + \phi(\pi_t - \pi)\left[\left(1 - \frac{1}{2\nu}\right)\pi_t + \frac{\pi}{2\nu}\right] \\ & - \phi\beta \mathbb{E}_t \left[\left(\frac{C_{t+1}/A_{t+1}}{C_t/A_t} \right)^{-\tau} \frac{Y_{t+1}/A_{t+1}}{Y_t/A_t} (\pi_{t+1} - \pi)\pi_{t+1} \right]. \end{aligned} \tag{1.17}$$

In the absence of nominal rigidities ($\phi = 0$) aggregate output is given by

$$Y_t^* = (1-\nu)^{1/\tau} A_t g_t, \tag{1.18}$$

which is the target level of output that appears in the output gap rule specification.

Since the non-stationary technology process A_t induces a stochastic trend in output and consumption, it is convenient to express the model in terms of detrended variables $c_t = C_t/A_t$ and $y_t = Y_t/A_t$. The model economy has a unique steady state in terms of the detrended variables that is attained if the innovations $\epsilon_{R,t}$, $\epsilon_{g,t}$, and $\epsilon_{z,t}$ are zero at all times. The steady state inflation π equals the target rate π^* and

$$r = \frac{\gamma}{\beta}, \quad R = r\pi^*, \quad c = (1-\nu)^{1/\tau}, \quad \text{and} \quad y = g(1-\nu)^{1/\tau}. \tag{1.19}$$

Let $\hat{x}_t = \ln(x_t/x)$ denote the percentage deviation of a variable x_t from its steady state x.

Then the model can be expressed as

$$
\begin{aligned}
1 &= \beta \mathbb{E}_t \left[e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{R}_t - \hat{z}_{t+1} - \hat{\pi}_{t+1}} \right] && (1.20) \\
\frac{1-\nu}{\nu \phi \pi^2} \left(e^{\tau \hat{c}_t} - 1 \right) &= \left(e^{\hat{\pi}_t} - 1 \right) \left[\left(1 - \frac{1}{2\nu} \right) e^{\hat{\pi}_t} + \frac{1}{2\nu} \right] && (1.21) \\
&\quad - \beta \, \mathbb{E}_t \left[\left(e^{\hat{\pi}_{t+1}} - 1 \right) e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{y}_{t+1} - \hat{y}_t + \hat{\pi}_{t+1}} \right] \\
e^{\hat{c}_t - \hat{y}_t} &= e^{-\hat{g}_t} - \frac{\phi \pi^2 g}{2} \left(e^{\hat{\pi}_t} - 1 \right)^2 && (1.22) \\
\hat{R}_t &= \rho_R \hat{R}_{t-1} + (1-\rho_R) \psi_1 \hat{\pi}_t + (1-\rho_R) \psi_2 \left(\hat{y}_t - \hat{g}_t \right) + \epsilon_{R,t} && (1.23) \\
\hat{g}_t &= \rho_g \hat{g}_{t-1} + \epsilon_{g,t} && (1.24) \\
\hat{z}_t &= \rho_z \hat{z}_{t-1} + \epsilon_{z,t}. && (1.25)
\end{aligned}
$$

1.2 Other DSGE Models

In addition to the small-scale New Keynesian DSGE model, we consider two other models: the widely-used Smets-Wouters (SW) model, which is a more elaborate version of the small-scale DSGE model that includes capital accumulation as well as wage rigidities, and a real business cycle model with a detailed characterization of fiscal policy.

1.2.1 The Smets-Wouters Model

The second DSGE model considered in this book is the Smets and Wouters (2007) model. The SW model is a more elaborate version of the small-scale DSGE model presented in the previous section. Capital is a factor of intermediate goods production, and in addition to price stickiness the model also features wage stickiness. In order to generate a richer autocorrelation structure, the model also includes investment adjustment costs, habit formation in consumption, and partial dynamic indexation of prices and wages to lagged values. The model is based on work by Christiano, Eichenbaum, and Evans (2005), who added various forms of frictions to a basic New Keynesian DSGE model in order to capture the dynamic response to a monetary policy shock as measured by a structural vector autoregression (VAR). In turn, Smets and Wouters (2003) augmented the Christiano-Eichenbaum-Evans model by additional shocks to be able to capture the joint dynamics of Euro Area output, consumption, investment, hours, wages, inflation, and interest rates. The Smets and Wouters (2003) paper was highly influential, not just in academic circles but also in central banks because it

demonstrated that a modern DSGE model that is useable for monetary policy analysis can achieve a time series fit that is comparable to a less restrictive vector autoregression (VAR). The 2007 version of the SW model contains a number of minor modifications of the 2003 model in order to optimize its fit on U.S. data. We will use the 2007 model exactly as it is presented in Smets and Wouters (2007) and refer the reader to that article for details. The log-linearized equilibrium conditions are reproduced in Appendix A.1.

1.2.2 A DSGE Model For the Analysis of Fiscal Policy

In the small-scale New Keynesian DSGE DSGE model and in the SW model fiscal policy is passive and non-distortionary. The government raises lump-sum taxes (or distributes lump-sum transfers) to ensure that the budget constraint is satisfied in every period. The level of government spending as a fraction of GDP evolves exogenously, an implicit money demand equation determines the amount of seignorage generated by the interest rate feedback rule, and the quantity of government bonds is not uniquely determined. These models were explicitly designed for the analysis of monetary policy and abstract from a realistic representation of fiscal policy.

In order to study the effects of exogenous changes in government spending and tax rates a more detailed representation of the fiscal sector is necessary. An example of such a model is the one studied by Leeper, Plante, and Traum (2010). While the authors abstract from monetary policy, they allow for capital, labor, and consumption tax rate that react to the state of the economy, in particular the level of output and debt, and are subject to exogenous shocks, which reflect unanticipated changes in fiscal policy. In addition to consumption, investment, and hours worked, the model is also estimated based on data on tax revenues, government spending, and government debt to identify the parameters of the fiscal policy rules. The estimated model can be used to assess the effect of counterfactual fiscal policies.

Chapter 2

Turning a DSGE Model into a Bayesian Model

Formally, a Bayesian model consists of a joint distribution of data Y and parameters θ. Throughout this book we will represent distributions by densities and denote the joint distribution by $p(Y, \theta)$. The joint distribution can be factored into a distribution of the data given the parameters, $p(Y|\theta)$, and a prior distribution $p(\theta)$. The density $p(Y|\theta)$, interpreted as function of θ is called likelihood function. It plays a central role in both Bayesian and frequentist inference. In order to turn the DSGE models of Chapter 1 into Bayesian models, we need to specify a probability distribution for the innovations of the exogenous shock processes, solve for the equilibrium law of motion, develop an algorithm that evaluates the likelihood function, and specify a prior distribution. We will illustrate these steps in the context of the small-scale New Keynesian DSGE model introduced in Section 1.1.

The solution of the DSGE model is sketched in Section 2.1. In this chapter, we will assume that the shock innovations are normally distributed and use a log-linearization (or first-order perturbation) to construct an approximate model solution. We will use the DSGE model solution as state-transition equations in a state-space representation of our empirical model. The measurement equations simply relate the (potentially unobserved) state variables of the DSGE model to observations on macroeconomic and financial time series. The evaluation of the likelihood function associated with the state-space representation of the DSGE model requires a filter that integrates out the hidden state variables of the DSGE model. We present a general characterization of the filtering algorithm. If the DSGE model is solved by a linear approximation technique and the innovations to the exogenous shock processes are Gaussian,

the filtering problem simplifies considerably. The likelihood function can be evaluated with the Kalman filter. The Kalman filter recursions are summarized for linear Gaussian state-space models are summarized in Section 2.2. Finally, we discuss the specification of prior distributions $p(\theta)$ in Section 2.3.

2.1 Solving a (Linearized) DSGE Model

Linearization and straightforward manipulation of Equations (1.20) to (1.22) yield

$$\begin{aligned}\hat{y}_t &= \mathbb{E}_t[\hat{y}_{t+1}] + \hat{g}_t - \mathbb{E}_t[\hat{g}_{t+1}] - \frac{1}{\tau}\left(\hat{R}_t - \mathbb{E}_t[\hat{\pi}_{t+1}] - \mathbb{E}_t[\hat{z}_{t+1}]\right) \\ \hat{\pi}_t &= \beta\mathbb{E}_t[\hat{\pi}_{t+1}] + \kappa(\hat{y}_t - \hat{g}_t) \\ \hat{R}_t &= \rho_R\hat{R}_{t-1} + (1-\rho_R)\psi_1\hat{\pi}_t + (1-\rho_R)\psi_2\left(\hat{y}_t - \hat{g}_t\right) + \epsilon_{R,t}\end{aligned} \tag{2.1}$$

where

$$\kappa = \tau\frac{1-\nu}{\nu\pi^2\phi}. \tag{2.2}$$

Equations (2.1) combined with the law of motion of the exogenous shocks in (1.24) and (1.25) form a linear rational expectations system in

$$x_t = [\hat{y}_t, \hat{\pi}_t, \hat{R}_t, \epsilon_{R,t}, \hat{g}_t, \hat{z}_t]'.$$

The linear rational expectations system can be cast in the canonical form used in Sims (2002):

$$\Gamma_0 s_t = \Gamma_1 s_{t-1} + \Psi\epsilon_t + \Pi\eta_t, \tag{2.3}$$

where $\epsilon_t = [\epsilon_{z,t}, \epsilon_{g,t}, \epsilon_{R,t}]'$. The vector η_t captures one-step-ahead rational expectations forecast errors. To write the equilibrium conditions of the small-scale New Keynesian model in the form of (2.3), we begin by replacing $\mathbb{E}_t[\hat{g}_{t+1}]$ and $\mathbb{E}_t[\hat{z}_{t+1}]$ in the first eqation of (2.1) with $\rho_g\hat{g}_t$ and $\rho_z\hat{z}_t$, respectively. We then introduce forecast errors for inflation and output. Let

$$\eta_{y,t} = y_t - \mathbb{E}_{t-1}[\hat{y}_t], \quad \eta_{\pi,t} = \pi_t - \mathbb{E}_{t-1}[\hat{\pi}_t], \tag{2.4}$$

and define $\eta_t = [\eta_{y,t}, \eta_{\pi,t}]$. Finally, define the expectation augmented $n \times 1$ state vector

$$s_t = \left[x_t', \mathbb{E}_t[\hat{y}_{t+1}], \mathbb{E}_t[\hat{\pi}_{t+1}]\right]'.$$

Using these definitions, the set of equations (2.1), (1.24), (1.25), and (2.4) can be written as (2.3). The system matrices Γ_0, Γ_1, Ψ, and Π are functions of the DSGE model parameters θ.

For the linearized equilibrium conditions (2.1) to characterize a solution to the underlying dynamic programming problems of the households and firms in the DSGE model, a set of transversality conditions needs to be satisfied. These conditions are satisfied, if the law of motion is non-explosive. This stability requirement restricts the set of solutions to (2.3). Depending on the system matrices Γ_0, Γ_1, Ψ, and Π the system may have no non-explosive solution (non-existence), exactly one stable solution (uniqueness), or many stable solutions (indeterminacy). Sims (2002) provides a general method to construct stable solutions for the canonical system (2.3).[1] The system can be transformed through a generalized complex Schur decomposition (QZ) of Γ_0 and Γ_1. There exist $n \times n$ matrices Q, Z, Λ, and Ω, such that $Q'\Lambda Z' = \Gamma_0$, $Q'\Omega Z' = \Gamma_1$, $QQ' = ZZ' = I$, and Λ and Ω are upper-triangular. Let $w_t = Z's_t$ and pre-multiply (2.3) by Q to obtain:

$$\begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ 0 & \Lambda_{22} \end{bmatrix} \begin{bmatrix} w_{1,t} \\ w_{2,t} \end{bmatrix} = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ 0 & \Omega_{22} \end{bmatrix} \begin{bmatrix} w_{1,t-1} \\ w_{2,t-1} \end{bmatrix} + \begin{bmatrix} Q_{1.} \\ Q_{2.} \end{bmatrix} (\Psi\epsilon_t + \Pi\eta_t). \tag{2.5}$$

The second set of equations can be rewritten as:

$$w_{2,t} = \Lambda_{22}^{-1}\Omega_{22}w_{2,t-1} + \Lambda_{22}^{-1}Q_{2.}(\Psi\epsilon_t + \Pi\eta_t) \tag{2.6}$$

Without loss of generality, we assume that the system is ordered and partitioned such that the $m \times 1$ vector $w_{2,t}$ is purely explosive, where $0 \leq m \leq n$.

A non-explosive solution of the LRE model (2.3) for s_t exists if $w_{2,0} = 0$ and for every $l \times 1$ vector of structural shock innovations ϵ_t, one can find a $k \times 1$ vector of rational expectations errors η_t that offsets the impact of ϵ_t on $w_{2,t}$:

$$\underbrace{Q_{2.}\Psi}_{m\times l}\underbrace{\epsilon_t}_{l\times 1} + \underbrace{Q_{2.}\Pi}_{m\times k}\underbrace{\eta_t}_{k\times 1} = \underbrace{0}_{m\times 1}. \tag{2.7}$$

If $m = k$ and the matrix $Q_{2.}\Pi$ is invertible, then the unique set of expectational errors that ensure the stability of the system is given by

$$\eta_t = -\big(Q_{2.}\Pi\big)^{-1}Q_{2.}\Psi\epsilon_t.$$

However, in general, the vector η_t, however, need not be unique. For instance, if the number of expectation errors k exceeds the number of explosive components m, Eq. (2.7) does not provide enough restrictions to uniquely determine the elements of η_t. Hence, it is possible

[1] There exist many alternative solution methods for linear rational expectations systems, e.g., Blanchard and Kahn (1980), Binder and Pesaran (1997) Anderson (2000), Klein (2000), Christiano (2002), and King and Watson (1998).

to introduce expectation errors (martingale difference sequences) ζ_t that are unrelated to the fundamental uncertainty ϵ_t without destabilizing the system. Using a singular value decomposition of $Q_{2.}\Pi$ of the form:

$$Q_{2.}\Pi = \underbrace{U_{.1}}_{m\times r}\underbrace{D_{11}}_{r\times r}\underbrace{V'_{.1}}_{r\times k},$$

we can express

$$\eta_t = (-V_{.1}D_{11}^{-1}U'_{.1}Q_{2.}\Psi + V_{.2}M_1)\epsilon_t + V_{.2}M_2\zeta_t, \tag{2.8}$$

where $V_{.2}$ is a matrix composed of orthonormal columns that are orthogonal to $V_{.1}$ (this matrix is a by-product of the singular value decomposition of $Q_{2.}\Pi$), M_1 is an arbitrary $(k-r)\times l$ matrix and M_2 is an arbitrary $(k-r)\times p$ matrix. The matrices M_1 and M_2 and the vector of so-called sunspot shocks ζ_t capture the potential multiplicity of non-explosive solutions (indeterminacy) of (2.7). A derivation of (2.8) is provided in Lubik and Schorfheide (2003).

The overall set of non-explosive solutions (if it is non-empty) to the linear rational expectations system (2.3) can be obtained from $s_t = Zw_t$, (2.5), and (2.8). If the system has a unique stable solution, then it can be written as a VAR in s_t:

$$s_t = \Phi_1(\theta)s_{t-1} + \Phi_\epsilon(\theta)\epsilon_t. \tag{2.9}$$

Here the coefficient matrices $\Phi^{(s)}$ and $\Phi^{(\epsilon)}$ are functions of the structural parameters of the DSGE model. The vector autoregressive representation in (2.9) forms the basis for our empirical model.

2.2 The Likelihood Function

In order to construct a likelihood function, we have to relate the model variables s_t to a set of observables y_t. Thus, the specification of the empirical model is completed by a set of measurement equations. For our small-scale New Keynesian model, We assume that the time period t in the model corresponds to one quarter and that the following observations are available for estimation: quarter-to-quarter per capita GDP growth rates (YGR), annualized quarter-to-quarter inflation rates (INFL), and annualized nominal interest rates (INT). The three series are measured in percentages and their relationship to the model variables is given

by the following set of equations:

$$\begin{aligned} YGR_t &= \gamma^{(Q)} + 100(\hat{y}_t - \hat{y}_{t-1} + \hat{z}_t) \\ INFL_t &= \pi^{(A)} + 400\hat{\pi}_t \\ INT_t &= \pi^{(A)} + r^{(A)} + 4\gamma^{(Q)} + 400\hat{R}_t. \end{aligned} \tag{2.10}$$

The parameters $\gamma^{(Q)}$, $\pi^{(A)}$, and $r^{(A)}$ are related to the steady states of the model economy as follows

$$\gamma = 1 + \frac{\gamma^{(Q)}}{100}, \quad \beta = \frac{1}{1 + r^{(A)}/400}, \quad \pi = 1 + \frac{\pi^{(A)}}{400}.$$

The structural parameters are collected in the vector θ. Since in the first-order approximation the parameters ν and ϕ are not separately identifiable, we express the model in terms of κ, defined in (2.2). Let

$$\theta = [\tau, \kappa, \psi_1, \psi_2, \rho_R, \rho_g, \rho_z, r^{(A)}, \pi^{(A)}, \gamma^{(Q)}, \sigma_R, \sigma_g, \sigma_z]'.$$

More generically, the measurement equation (2.10) can be expressed as

$$y_t = \Psi_0(\theta) + \Psi_1(\theta)t + \Psi_2(\theta)s_t + u_t, \tag{2.11}$$

where we allow for a vector of measurement errors u_t.[2]

Equations (2.9) and (2.11) provide a state-space representation for the linearized DSGE model. The challenge in evaluating the likelihood function is that the states s_t are (at least partially) unobserved. Let $X_{t_1:t_2} = \{x_{t_1}, x_{t_1+1}, \ldots, x_{t_2}\}$. The state space representation provides a joint density for the observations and latent states given the parameters: $p(Y_{1:T}, S_{1:T}|\theta)$. However, inference is based on $p(Y_{1:T}|\theta)$ and the hidden states have to be integrated out. The likelihood function can be factorized as follows:

$$p(Y_{1:T}|\theta) = \prod_{t=1}^{T} p(y_t|Y_{1:t-1}, \theta). \tag{2.12}$$

A filter generates a sequence of conditional distributions $s_t|Y_{1:t}$ and as a by-product produces the sequence of densities $p(y_t|Y_{1:t-1}, \theta)$.

[2]The DSGE model solution method implies that certain linear combinations of model variables, namely $w_{2,t}$ in (2.5), are equal to zero. If some elements of $w_{2,t}$ only depend on variables that can be measured in the data, this implication is most likely violated. To cope with this problem, one can either limit the number of observables included in y_t, as we do in the New Keynesian model, or include so-called measurement errors as, for instance, in Sargent (1989), Altug (1989), and Ireland (2004).

Algorithm 1 (Generic Filter) *1. Initialization at time $t-1$: $p(s_{t-1}|Y_{1:t-1},\theta)$*

2. Forecasting t given $t-1$:

(a) Transition equation:

$$p(s_t|Y_{1:t-1},\theta) = \int p(s_t|s_{t-1},Y_{1:t-1},\theta)p(s_{t-1}|Y_{1:t-1},\theta)ds_{t-1}$$

(b) Measurement equation:

$$p(y_t|Y_{1:t-1},\theta) = \int p(y_t|s_t,Y_{1:t-1},\theta)p(s_t|Y_{1:t-1},\theta)ds_t$$

3. Updating with Bayes theorem. Once y_t becomes available:

$$p(s_t|Y_{1:t},\theta) = p(s_t|y_t,Y_{1:t-1},\theta) = \frac{p(y_t|s_t,Y_{1:t-1},\theta)p(s_t|Y_{1:t-1},\theta)}{p(y_t|Y_{1:t-1},\theta)}$$

If the DSGE model is log-linearized and the errors are Gaussian, then the Kalman filter can be used to construct the likelihood function. To complete the model specification we make the following distributional assumptions about the distribution of the structural innovations ϵ_t, the measurement errors u_t, and the initial state s_0:

$$\epsilon_t \sim iidN(0,\Sigma_\epsilon), \quad u_t \sim iidN(0,\Sigma_u), \quad s_0 \sim N\big(\hat{s}_{0|0}, P_{0|0}\big). \tag{2.13}$$

In stationary models it is common to assume that $\bar{s}_{0|0}$ and $P_{0|0}$ corresponds to the invariant distribution associated with the law of motion of s_t in (2.9). The four conditional distributions in the description of Algorithm 1 for a linear Gaussian state space model are summarized in Table 2.1. Derivations can be found in textbook treatments of the Kalman filter, e.g., Hamilton (1994) or Durbin and Koopman (2001).

2.3 Priors

Prior distributions play an important role in the estimation of DSGE models. They allow researchers to incorporate information not contained in the estimation sample Y into the empirical analysis. While priors could in principle be formed by pure introspection, in reality most priors (as well as most model specifications) are based on some empirical observations. To indicate this dependence on non-sample (meaning other than Y) information, we could write $p(\theta|\mathcal{X}^0)$ instead of $p(\theta)$, but for notational convenience we omit the dependence on $\mathcal{X}^0$.

Table 2.1: Conditional Distributions for Kalman Filter

	Distribution	Mean and Variance
$s_{t-1}\|(Y_{1:t-1},\theta)$	$N\big(\bar{s}_{t-1\|t-1}, P_{t-1\|t-1}\big)$	Given from Iteration $t-1$
$s_t\|(Y_{1:t-1},\theta)$	$N\big(\bar{s}_{t\|t-1}, P_{t\|t-1}\big)$	$\bar{s}_{t\|t-1} = \Phi_1 \bar{s}_{t-1\|t-1}$ $P_{t\|t-1} = \Phi_1 P_{t-1\|t-1}\Phi_1' + \Phi_\epsilon \Sigma_\epsilon \Phi_\epsilon'$
$y_t\|(Y_{1:t-1},\theta)$	$N\big(\bar{y}_{t\|t-1}, F_{t\|t-1}\big)$	$\bar{y}_{t\|t-1} = \Psi_0 + \Psi_1 t + \Psi_2 \bar{s}_{t\|t-1}$ $F_{t\|t-1} = \Psi_2 P_{t\|t-1}\Psi_2' + \Sigma_u$
$s_t\|(Y_{1:t},\theta)$	$N\big(\bar{s}_{t\|t}, P_{t\|t}\big)$	$\bar{s}_{t\|t} = \bar{s}_{t\|t-1} + P_{t\|t-1}\Psi_2' F_{t\|t-1}^{-1}(y_t - \bar{y}_{t\|t-1})$ $P_{t\|t} = P_{t\|t-1} - P_{t\|t-1}\Psi_2' F_{t\|t-1}^{-1}\Psi_2 P_{t\|t-1}$

The tacit assumption underlying posterior inference with a prior that is constructed from non-sample information is that $p(Y|\theta,\mathcal{X}^0) = p(Y|\mathcal{X}^0,\theta)$, that is, the two sources of information are independent conditional on θ. This is assumption a reasonable approximation if the observations in $\mathcal{X}^0$ pre-date the observations in Y or if Y consists of macroeconomic time series and $\mathcal{X}^0$ contains micro-level data from an overlapping time period.

Del Negro and Schorfheide (2008) distinguish between three groups of parameters. The first group, denoted by $\theta_{(ss)}$, are parameters that can be identified from the steady-state relationships. For instance, in the small-scale New Keynesian model $\theta_{(ss)} = [r^{(A)}, \pi^{(A)}, \gamma^{(Q)}]'$. These three parameters affect the steady state real interest rate, inflation rate, and overall growth rate of the economy. The second group of parameters consists of parameters that characterize the law of motion of the exogenous shock processes: $\theta_{(exo)} = [\rho_g, \rho_z, \sigma_g, \sigma_z, \sigma_R]'$. Finally, the last group of parameters control the endogenous propagation mechanisms without affecting the steady state of the model: $\theta_{(endo)} = [\tau, \kappa, \psi_1, \psi_2, \rho_R]'$.

Priors for $\theta_{(ss)}$ are often based on pre-sample averages. For instance, if the estimation sample starts in 1983:I, the prior distribution for $r^{(A)}$, $\pi^{(A)}$, and $\gamma^{(Q)}$ may be informed by data from the 1970s. Priors for $\theta_{(endo)}$ may be partly based on microeconometric evidence. For instance, in a version of the New Keynesian model that replaces the quadratic price adjustment costs with a Calvo mechanism (intermediate good producers can re-optimize their prices with an exogenous probability $1-\zeta_p$ and are unable to change their prices with probability ζ_p) the slope of the Phillips curve κ is related to the frequency of price changes, which can be measured from micro-level data. Ríos-Rull, Schorfheide, Fuentes-Albero, Kryshko,

and Santaeulalia-Llopis (2012) provide a very detailed discussion of prior elicitation for the Frisch labor supply elasticity. In the simple New Keynesian model, this elasticity is implicitly fixed at infinity, because of the quasi-linear specification of the household utility function. Priors for $\theta_{(exo)}$ are the most difficult to specify, because the exogenous processes tend to be unobserved. Del Negro and Schorfheide (2008) suggest to elicit priors for $\theta_{(exo)}$ indirectly. Conditional on $\theta_{(ss)}$ and $\theta_{(endo)}$, the exogenous shock parameters determine the volatility and persistence of y_t. Thus, beliefs – possibly informed by pre-sample observations, about the dynamics of the observables, can be mapped into beliefs about the persistence and volatility of the exogenous shocks. This can be done using the formal procedure described in Del Negro and Schorfheide (2008), or it can be done informally using an iterative procedure that starts by specifying an initial prior for $\theta_{(exo)}$, generating parameter draws from this prior, simulating trajectories from the DSGE model, examining the implied sample moments, and potentially re-specifying the prior for $\theta_{(exo)}$ until the desired implicit prior for the moments of y_t is obtained.

Table 2.2 provides a prototypical prior distribution for the coefficients of the small New Keynesian DSGE model. It specifies marginal distributions for all elements of the θ vector. The joint prior distribution is the product of the marginal densities. The domain of the prior is truncated to ensure that the the linearized rational expectations model has a unique stable solution. The prior for the steady state parameters are based on averages from a pre-1983:I sample. Unlike in much of the literature, the prior distributions for many of the other parameters are uniform on a bounded domain. In high-dimensional models, it might be desirable to introduce some dependence among the parameters. Methods to do so are provided in Del Negro and Schorfheide (2008).

Table 2.2: Prior Distribution

Name	Domain		Prior	
		Density	Para (1)	Para (2)
τ	$\mathbb{R}^+$	Gamma	2.00	0.50
κ	$\mathbb{R}^+$	Uniform	0.00	1.00
ψ_1	$\mathbb{R}^+$	Gamma	1.50	0.25
ψ_2	$\mathbb{R}^+$	Gamma	0.50	0.25
ρ_R	$[0, 1)$	Uniform	0.00	1.00
ρ_G	$[0, 1)$	Uniform	0.00	1.00
ρ_Z	$[0, 1)$	Uniform	0.00	1.00
$r^{(A)}$	$\mathbb{R}^+$	Gamma	0.50	0.50
$\pi^{(A)}$	$\mathbb{R}^+$	Gamma	7.00	2.00
$\gamma^{(Q)}$	$\mathbb{R}$	Normal	0.40	0.20
$100\sigma_R$	$\mathbb{R}^+$	InvGamma	0.40	4.00
$100\sigma_G$	$\mathbb{R}^+$	InvGamma	1.00	4.00
$100\sigma_Z$	$\mathbb{R}^+$	InvGamma	0.50	4.00

Notes: Para (1) and Para (2) list the means and the standard deviations for Beta, Gamma, and Normal distributions; the upper and lower bound of the support for the Uniform distribution; s and ν for the Inverse Gamma distribution, where $p_{\mathcal{IG}}(\sigma|\nu, s) \propto \sigma^{-\nu-1}e^{-\nu s^2/2\sigma^2}$. The effective prior is truncated at the boundary of the determinacy region.

Chapter 3

A Primer on Bayesian Inference

The prior distribution $p(\theta)$ discussed in the previous section describes the initial state of knowledge – before observing Y – about the parameter θ. Unlike under the frequentist paradigm, the parameter θ is regarded as a random variable. The calculus of probability is used to characterize the state of knowledge or the degree of beliefs of an individual with respect to events or quantities – such as parameters – that have not (yet) been observed, and maybe cannot be observed, by that individual. The Bayesian approach prescribes consistency among the beliefs held by an individual, and their reasonable relation to any kind of objective data. Learning about θ takes place by updating the prior distribution in light of the data Y. The likelihood function $p(Y|\theta)$ summarizes the information about the parameter contained in the sample Y. According to Bayes Theorem, the conditional distribution of θ given Y is given by

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}. \tag{3.1}$$

This distribution is called *posterior* distribution. The term in the denominator is called marginal likelihood. It is defined as

$$p(Y) = \int p(Y|\theta)p(\theta)d\theta \tag{3.2}$$

and normalizes the posterior density such that it integrates to one.

In a nutshell, Bayesian inference amounts to characterizing properties of the posterior distribution $p(\theta|Y)$. Unfortunately, for many interesting models, including the DSGE models considered in this book, a direct analysis of the posterior is not feasible. All that can be done is to numerically evaluate the prior density $p(\theta)$ and the likelihood function $p(Y|\theta)$. In order to compute posterior quantiles and moments of functions $h(\theta)$ we have to rely on numerical

techniques. In particular, we will use posterior sampler that generate sequences of draws θ^i, $i = 1, \ldots, N$ from $p(\theta|Y)$. To the extent that (Monte Carlo) averages of these draws satisfy a strong law of large numbers (SLLN) and possibly a central limit theorem (CLT), we can use them to approximate posterior moments.

Before delving in the Bayesian inference for DSGE models, we will take a step back and begin in Section 3.1 Bayesian inference in a simple autoregressive (AR) model, which takes the form of a Gaussian linear regression. For this model, the posterior distribution can be characterized analytically and closed-form expressions for its moments are readily available. Draws from the posterior distribution can be easily generated using a direct sampling algorithm. In Section 3.2 we modify the parameterization of the AR(1) model to introduce some identification problems. Lack of or weak identification of key structural parameters is a common occurrence in the context of DSGE models. In our AR(1) example the posterior distribution of the parameter of interest becomes non-Gaussian, and sampling from this posterior is now less straightforward. We proceed by introducing two important posterior samplers. We will subsequently employ variants of these samplers to implement the Bayesian analysis of DSGE models. Section 3.3 focuses on importance sampling, whereas Section 3.4 provides an introduction to the Metropolis-Hastings algorithm. Final, we wrap up this primer on Bayesian inference in Section 3.5, which discusses how to turn posterior distributions – or draws from posterior distributions – into point estimates, interval estimates, forecasts, and how to solve general decision problems.

3.1 The Posterior of A Linear Gaussian Model

Since we do not expect our readers to be experts in Bayesian analysis, we begin with a simple regression model to illustrate some of the principles and mechanics of Bayesian inference. Consider the AR(1) model

$$y_t = \theta y_{t-1} + u_t, \quad u_t|Y_{1:t-1} \sim iid\mathcal{N}(0,1), \quad t = 1, \ldots, T \tag{3.3}$$

Conditional on the initial observation y_0 the likelihood function is of the form

$$\begin{aligned} p(Y_{1:t}|y_0, \theta) &= \prod_{t=1}^{T} p(y_t|Y_{1:t-1}, \theta) \\ &= (2\pi)^{-T/2} \exp\left\{-\frac{1}{2}(Y - X\theta)'(Y - X\theta)\right\}, \end{aligned} \tag{3.4}$$

where $y_{1:t} = \{y_1, \ldots, y_t\}$ and the $T \times 1$ matrices Y and X are composed of the elements y_t and $x_t = y_{t-1}$. Suppose the prior distribution is of the form

$$\theta \sim N\left(0, \tau^2\right) \tag{3.5}$$

with density

$$p(\theta) = (2\pi\tau^2)^{-1/2} \exp\left\{-\frac{1}{2\tau'\tau}\theta^2\right\}. \tag{3.6}$$

The hyperparameter τ controls the variance of the prior distribution. We will subsequently vary τ to illustrate the effect of the prior variance on the posterior distribution.

According to Bayes Theorem the posterior distribution of θ is proportional ($\propto$) to the product of prior density and likelihood function

$$p(\theta|Y) \propto p(\theta)p(Y|\theta). \tag{3.7}$$

To simplify the notation we dropped y_0 from the conditioning set and we replaced $y_{1:t}$ by the matrix Y. Absorbing terms that do not depend on θ into the proportionality constant, the right-hand-side of (3.7) can be written as

$$\begin{aligned} & p(\theta)p(Y|\theta) \\ & \quad \propto \quad \exp\left\{-\frac{1}{2}[Y'Y - \theta'X'Y - Y'X\theta - \theta'X'X\theta - \tau^{-2}\theta'\theta]\right\}. \end{aligned} \tag{3.8}$$

Straightforward algebraic manipulations let us express the exponential term as

$$\begin{aligned} & Y'Y - \theta'X'Y - Y'X\theta - \theta'X'X\theta - \tau^{-2}\theta'\theta \\ & = \quad \left(\theta - (X'X + \tau^{-2})^{-1}X'Y\right)'\left(X'X + \tau^{-2}\right)\left(\theta - (X'X + \tau^{-2})^{-1}X'Y\right) \\ & \quad + Y'Y - Y'X(X'X + \tau^{-2})^{-1}X'Y. \end{aligned} \tag{3.9}$$

Since the exponential term is a quadratic function of θ we can deduce that the posterior distribution is Normal

$$\theta|Y \sim \mathcal{N}(\bar{\theta}, \bar{V}_\theta) \tag{3.10}$$

with posterior mean and covariance

$$\bar{\theta} = (X'X + \tau^{-2})^{-1}X'Y, \quad \bar{V}_\theta = (X'X + \tau^{-2})^{-1}.$$

Define $\hat{\theta}_{mle} = (X'X)^{-1}X'Y$ and write

$$\bar{\theta} = (X'X + \tau^{-2})^{-1}\left(X'X\hat{\theta}_{mle} + \tau^{-2} \cdot 0\right).$$

Thus, the posterior mean of θ is a weighted average of the maximum likelihood estimator and the prior mean, which is zero. The weights depend on the information contents of the likelihood function, $X'X$, and the prior precision τ^{-2}. Holding the data fixed, and decrease in τ shifts the posterior mean toward the prior mean. Moreover, it decreases the prior variance.

In our derivation of the posterior distribution we have deliberately ignored all the normalization constants and only focused on terms that depend on θ. This approach served us well because based on the general shape of the posterior density we were able to determine that it belongs to the family of Gaussian densities, for which the normalization constants are well known. We can use this information to easily derive the marginal density $p(Y)$ that appears in the denominator of Bayes Theorem in (3.1). Write

$$\begin{aligned} p(Y) &= \frac{p(Y|\theta)p(\theta)}{p(\theta|Y)} \\ &= (2\pi)^{-T/2}\exp\left\{-\frac{1}{2}[Y'Y - Y'X(X'X+\tau^{-2})^{-1}X'Y]\right\} \\ &\quad \times|1+\tau^2X'X|^{-1/2}. \end{aligned} \tag{3.11}$$

The second expression on the right-hand-side is obtained by replacing the Gaussian densities $p(Y|\theta)$, $p(\theta)$, and $p(\theta|Y)$ by (3.4), (3.6), and the density associated with (3.10), respectively. The exponential term in (3.11) can be interpreted as goodness-of-fit, whereas $|1+\tau^2X'X|$ is a penalty for model complexity. If τ is close to zero, our model has essentially no free parameters because the tight prior distribution forces the posterior to be close to zero as well. In this case the goodness-of-fit term tends to be small but the penalty for model complexity is also small. If, on the other hand, τ is large, then the goodness-of-fit term is large, as it approximately equals (minus) the sum of squared residuals from an OLS regression. The penalty tends to be large as well. Thus, neither specifications with a very concentrated prior or a very diffuse prior tend to be associated with a high marginal data density.

Marginal data densities are important for Bayesian analysis because they determine the posterior model probabilities. Suppose a researcher assigns prior probabilities $\gamma_{j,0}$ to models M_j, $j=1,\ldots,J$, then the posterior model probabilities are given by

$$\gamma_{j,T} = \frac{\gamma_{j,0}p(Y|M_j)}{\sum_{j=1}^{J}\gamma_{j,0}p(Y|M_j)}. \tag{3.12}$$

We will discuss the role of posterior model probabilities in the evaluation of DSGE models later on.

To economize on notation we will often abbreviate posterior distributions $p(\theta|Y)$ by $\pi(\theta)$ and posterior expectations of $h(\theta)$ by

$$\mathbb{E}_\pi[h] = \mathbb{E}_\pi[h(\theta)] = \int h(\theta)\pi(\theta)d\theta = \int h(\theta)p(\theta|Y)d\theta. \tag{3.13}$$

Much of this monograph focuses on comparing algorithms that generate draws $\{\theta^i\}_{i=1}^N$ from posterior distributions of parameters in DSGE models. These draws can then be transformed into objects of interest, $h(\theta^i)$, and a Monte Carlo average of the form

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^i) \tag{3.14}$$

may be used to approximate the posterior expectation of $\mathbb{E}_\pi[h]$. For the approximation to be useful, it should satisfy a a SLLN and a CLT. In the simple linear regression model with Gaussian posterior given by (3.10) it is possible to sample directly from the posterior distribution and obtain independently and identically distributed (iid) draws from $\pi(\cdot)$.

Algorithm 2 (Direct Sampling) *For $i = 1$ to N, draw θ^i from $N\big(\bar{\theta}, \bar{V}_\theta\big)$.*

Provided that $\mathbb{V}_\pi[h(\theta)] < \infty$ we can deduce from Kolmogorov's SLLN and the Lindeberg-Levy CLT that

$$\begin{aligned} \bar{h}_N &\xrightarrow{a.s.} \mathbb{E}_\pi[h] \\ \sqrt{N}\left(\bar{h}_N - \mathbb{E}_\pi[h]\right) &\Longrightarrow N\big(0, \mathbb{V}_\pi[h(\theta)]\big). \end{aligned} \tag{3.15}$$

Thus, the posterior variance of $h(\theta)$, scaled by $1/N$, determines the accuracy of the Monte Carlo approximation. In the context of DSGE models, direct *iid* sampling from the posterior is generally infeasible and the variance of the Monte Carlo approximation is (much) larger than $\mathbb{V}_\pi[h(\theta)]/N$. The ratio of the actual variance to the infeasible variance $\mathbb{V}_\pi[h(\theta)]/N$ provides a measure of efficiency of the algorithm.

3.2 A Posterior of a Set-Identified Model

There are many applications in which certain parameters of DSGE models are difficult to identify. Identification problems in DSGE models typically come in two varieties: (i) local identification problems in which the likelihood function is fairly flat in certain directions of

the parameter space; (ii) global identification problems in which the likelihood function is multi-modal. The example in this subsection is designed to showcase a local identification problem.

Suppose that y_t follows an AR(1) process with autoregressive coefficient that we now denote by ϕ. However, unlike in the previous section, we now assume that the object of interest is not the autoregressive parameter, but instead a parameter θ that can be bounded based on ϕ as follows:

$$\phi \leq \theta \quad \text{and} \quad \theta \leq \phi + 1.$$

Strictly speaking, the parameter θ is set identified. The interval $\Theta(\phi) = [\phi,\ \phi + 1]$ is called the identified set and in this simple example its length is equal to one. To complete the model specification we specify a prior for θ conditional on ϕ of the form

$$\theta|\phi \sim U[\phi, \phi + 1]. \tag{3.16}$$

The joint posterior distribution of θ and ϕ can be characterized as follows

$$p(\theta, \phi|Y) = p(\phi|Y)p(\theta|\phi, Y) \propto p(Y|\phi)p(\theta|\phi)p(\phi). \tag{3.17}$$

Since θ does not enter the likelihood function, we can immediately deduce that

$$p(\phi|Y) = \frac{p(Y|\phi)p(\phi)}{\int p(Y|\phi)p(\phi)d\phi} \quad \text{and} \quad p(\theta|\phi, Y) = p(\theta|\phi). \tag{3.18}$$

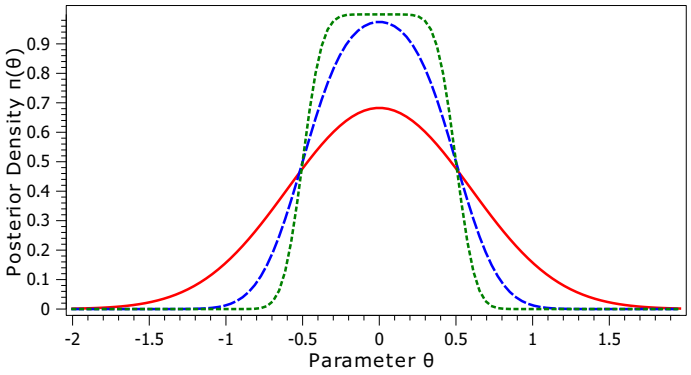
Following (3.10), suppose that the posterior distribution of ϕ takes the form $\phi|Y \sim N(\bar{\phi}, \bar{V}_\phi)$. We deduced that the posterior distribution of θ conditional on ϕ is simply equal to the prior distribution in (3.16). Since the prior of θ on the set $\Theta(\phi)$ is uniform, the marginal posterior distribution of θ is given by

$$\begin{aligned} \pi(\theta) &= \int_{\theta-1}^{\theta} p(\phi|Y)p(\theta|\phi)d\phi \\ &= \Phi_N\left(\frac{\theta - \bar{\phi}}{\sqrt{\bar{V}_\phi}}\right) - \Phi_N\left(\frac{\theta - 1 - \bar{\phi}}{\sqrt{\bar{V}_\phi}}\right), \end{aligned} \tag{3.19}$$

where $\Phi_N(x)$ is the cumulative density function of a $N(0, 1)$.

The posterior of θ has a more complicated shape than the posterior of ϕ. Figure 3.1 depicts the posterior for three choices of $\bar{V}_\phi$. If the posterior variance of the reduced-form parameter ϕ is large, the posterior looks almost Gaussian. However, as $\bar{V}_\phi$ decreases, the posterior starts to resemble the shape of a step function that increases from zero to one at $\theta = -0.5$ and then

Figure 3.1: Posterior Distribution for Set-Identified Model



Notes: The figure depicts the posterior distribution $\pi(\theta)$ in (3.19) for $\bar{\phi} = -0.5$ and $\bar{V}_\phi$ equal to 1/4 (solid red), 1/20 (dashed blue), and 1/100 (dotted green).

drops to zero around $\theta = 0.5$. The flatness of the posterior density on the interval $[\bar{\phi}, \bar{\phi} + 1]$ gets more pronounced as the sample size increases and the uncertainty about the parameter ϕ vanishes. In this stylized example, it is possible to sample from the posterior distribution of θ directly by first sampling $\phi^i \sim N\big(\bar{\phi}, \bar{V}_\phi\big)$ and then sampling $\theta^i|\phi^i \sim U\big[\phi^i, \phi^i + 1\big]$. This scheme generates *iid* draws from the joint posterior $(\phi, \theta)|Y$. The θ^i draws can then be used to construct Monte Carlo approximations for moments associated with the marginal posterior distribution $\theta|Y$. Instead of using a direct sampler to generate draws from the posterior, we will consider an importance sampler, which is an important building block of some of the algorithms considered later on in this book.

3.3 Importance Sampling

Instead of attempting to sample directly from the posterior $\pi(\theta)$ in (3.19), we could approximate $\pi(\cdot)$ by using a different, tractable density $g(\theta)$ that is easy to sample from. Because in many applications the posterior density can only be evaluated up to a constant of proportionality, we write

$$\pi(\theta) = \frac{f(\theta)}{Z}. \tag{3.20}$$

Often, $f(\theta)$ corresponds to the product of likelihood function and prior density $p(Y|\theta)p(\theta)$ which appears in the numerator of (3.1) and Z corresponds to the marginal likelihood $p(Y)$ that appears in the denominator of Bayes Theorem.

3.3.1 The Importance Sampling Algorithm

Importance sampling (IS) is based on the identity

$$E_\pi[h(\theta)] = \int h(\theta)\pi(\theta)d\theta = \frac{1}{Z}\int_\Theta h(\theta)\frac{f(\theta)}{g(\theta)}g(\theta)d\theta. \tag{3.21}$$

The ratio

$$w(\theta) = \frac{f(\theta)}{g(\theta)} \tag{3.22}$$

is called the (unnormalized) importance weight. We can also define a normalized importance weight as

$$v(\theta) = \frac{w(\theta)}{\int w(\theta)g(\theta)d\theta} = \frac{w(\theta)}{\int Z\pi(\theta)d\theta} = \frac{w(\theta)}{Z}. \tag{3.23}$$

It is straightforward to verify based on (3.21) and the definition in (3.22) that $\int v(\theta)h(\theta)d\theta = \mathbb{E}_\pi[h(\theta)]$.

Algorithm 3 (Importance Sampling) *1. For $i = 1$ to N, draw $\theta^i \overset{iid}{\sim} g(\theta)$ and compute the unnormalized importance weights*

$$w^i = w(\theta^i) = \frac{f(\theta^i)}{g(\theta^i)}. \tag{3.24}$$

2. Compute the normalized importance weights

$$W^i = \frac{w^i}{\frac{1}{N}\sum_{i=1}^N w^i}. \tag{3.25}$$

An approximation of $\mathbb{E}_\pi[h(\theta)]$ is given by

$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^N W^i h(\theta^i). \tag{3.26}$$

Note that according to our definitions W^i is different from $v(\theta^i)$. W^i is normalized by the sample average of the unnormalized weights w^i, whereas $v(\theta)$ is normalized by the population normalization constant Z. By construction, the sample average $\frac{1}{N}\sum_{i=1}^N W^i = 1$.

3.3.2 Convergence and Accuracy

Provided that $\mathbb{E}_g[|hf/g|] < \infty$ and $\mathbb{E}_g[|f/g|] < \infty$, see Geweke (1989), the Monte Carlo estimate $\bar{h}_N$ defined in (3.26) converges almost surely (a.s.) to $E_\pi[h(\theta)]$ as $N \longrightarrow \infty$. In Chapter 5 we will refer to the swarm of pairs $\{(\theta^i, W^i)\}_{i=1}^N$ as a particle approximation of $\pi(\theta)$. The accuracy of the approximation is driven by the "closeness" of $g(\cdot)$ to $f(\cdot)$ and is reflected in the distribution of the weights. If the distribution of weights is very uneven, the Monte Carlo approximation $\bar{h}$ is inaccurate. Uniform weights arise if $g(\cdot) \propto f(\cdot)$, which means that we are sampling directly from $\pi(\theta)$.

The limit distribution of the Monte Carlo approximation can be derived as follows. Define the population analogue of the normalized importance weights as $v(\theta) = w(\theta)/Z$ and write

$$\bar{h}_N = \frac{\frac{1}{N}\sum_{i=1}^N (w^i/Z)h(\theta^i)}{\frac{1}{N}\sum_{i=1}^N (w^i/Z)} = \frac{\frac{1}{N}\sum_{i=1}^N v(\theta^i)h(\theta^i)}{\frac{1}{N}\sum_{i=1}^N v(\theta^i)}.$$

Now consider a first-order Taylor series expansion in terms of deviations of the numerator from $\mathbb{E}_\pi[h]$ and deviations of the denominator around 1:

$$\begin{aligned}
\sqrt{N}(\bar{h}_N - \mathbb{E}_\pi[h]) &= \sqrt{N}\left(\frac{1}{N}\sum_{i=1}^N v(\theta^i)h(\theta^i) - \mathbb{E}_\pi[h]\right) \\
&\quad -\mathbb{E}_\pi[h]\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^N v(\theta^i) - 1\right) + o_p(1) \\
&= (I) - \mathbb{E}_\pi[h]\cdot(II) + o_p(1),
\end{aligned} \tag{3.27}$$

say. Provided that $\sup_\theta \ \pi/g < \infty$ and $\mathbb{E}_g[h^2] < \infty$, we can apply a multivariate extension of the Lindeberg-Levy CLT to the terms (I, II). Using straightforward but tedious algebra it can be shown that the variances and covariance of I and II are given by

$$\begin{aligned}
\mathbb{V}_g[hv] &= \mathbb{E}_\pi[(\pi/g)h^2] - \mathbb{E}_\pi^2[h],\ \mathbb{V}_g[v] = \mathbb{E}_\pi[(\pi/g)] - 1, \\
COV_g(hv, v) &= \big(\mathbb{E}_\pi[(\pi/g)h] - \mathbb{E}_\pi[h]\big).
\end{aligned}$$

In turn we can deduce that

$$\sqrt{N}(\bar{h}_N - \mathbb{E}_\pi[h]) \Longrightarrow N\big(0, \Omega(h)\big), \quad \text{where} \quad \Omega(h) = \mathbb{V}_g[(\pi/g)(h - \mathbb{E}_\pi[h])]. \tag{3.28}$$

Using a crude approximation (see, e.g., Liu (2001)), we can factorize $\Omega(h)$ as follows:

$$\Omega(h) \approx \mathbb{V}_\pi[h]\big(\mathbb{V}_g[\pi/g] + 1\big). \tag{3.29}$$

This allows us to define an (approximate) effective measure of sample size that is independent of the function $h(\cdot)$ and only depends on the variance of the importance weights:

$$ESS = N\frac{\mathbb{V}_\pi[h]}{\Omega(h)} \approx \frac{N}{1 + \mathbb{V}_g[\pi/g]}. \tag{3.30}$$

The approximation highlights that the larger the variance of the importance weights, the less accurate the Monte Carlo approximation relative to the accuracy that could be achieved with an *iid* sample from the posterior.

3.3.3 A Numerical Illustration

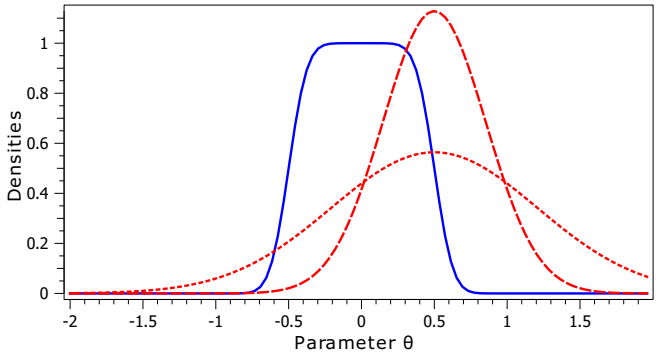
Figure 3.2 provides a numerical illustration of the importance sampling algorithm in the context of the posterior density (3.19) associated with the set-identified model in Section 3.2. Panel (i) depicts the posterior density for $\bar{\phi} = -0.5$ and $\bar{V} = 100$. We consider two importance sampling densities. Both are centered at $\theta = 0.5$. The first density ("concentrated") has a variance of 0.125, whereas the second density ("diffuse") has a larger variance of 0.5. The concentrated importance sampling density assigns a very small probability to the interval $[-0.5, -0.25]$ which has a large probability under the posterior distribution.

The accuracy of the importance sampling approximations are illustrated in Panels (ii) and (iii) as a function of the number of draws N. We depict the inefficiency factor $\Omega(h)/\mathbb{V}_\pi[h]$ as well as a simulation-based inefficiency factor. We run the importance sampling algorithm 500 times and compute the variance of the Monte Carlo approximations of $\mathbb{E}_\pi[\theta]$ and $\mathbb{E}_\pi[\theta^2]$ across the runs. We multiply this variance by N and divide by $\mathbb{V}_\pi[h]$ so that it is on the same scale as the asymptotic inefficiency factor. In general, the asymptotic approximation is very accurate. A comparison between Panels (ii) and (iii) highlights that the approximation with the "concentrated" importance sampling density is a lot less accurate than the approximation obtained with the "diffuse" importance sampling densities, which does a much better job in covering the tails of the posterior distribution. Finally, we also plot N/ESS, where ESS was defined in (3.30). The ESS-based inefficiency measure only provides a very crude approximation of the accuracy of the importance sampling approximation.

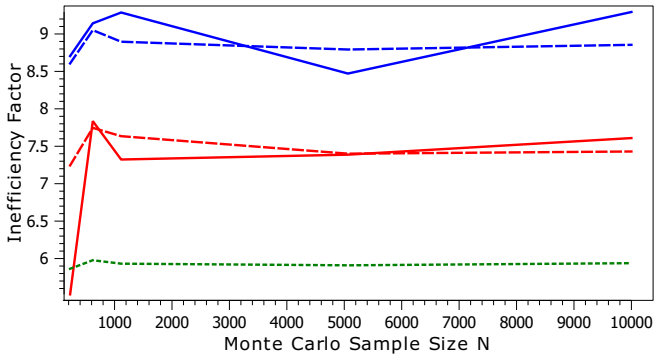
In general, it is important that the importance density g is tailored toward the target distribution π to maintain a small variance of the importance weights. In some applications, a good importance density can be obtained by centering a fat-tailed t distribution at the mode of π and using a scaled version of the inverse Hessian of $\ln \pi$ at the mode to align the contours of the importance density with the contours of the posterior π. The sequential

Figure 3.2: Importance Sampling Approximations of $\mathbb{E}_\pi[\theta]$ and $\mathbb{E}_\pi[\theta^2]$

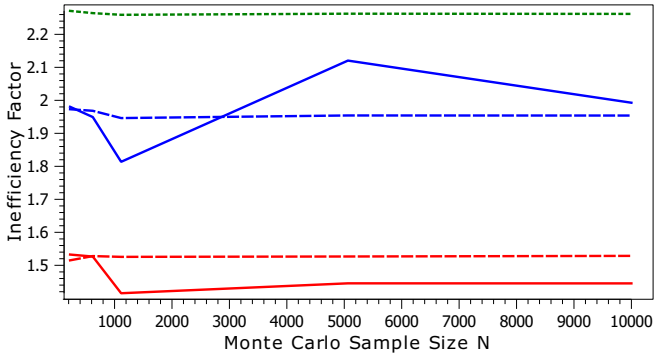
(i) Importance Sampling Densities



(ii) Inefficiency Factors for Concentrated IS Density



(iii) Inefficiency Factors for Diffuse IS Density



Notes: Panel (i) depicts the posterior density $\pi(\theta)$ as well as two importance sampling densities ("concentrated" and "diffuse") $g(\theta)$. Panels (ii) and (iii) depict large sample inefficiency factors $\Omega(h)/\mathbb{V}_\pi[h]$ (dashed) as well as their small sample approximations (solid). We consider $h(\theta) = \theta$ (blue) and $h(\theta) = \theta^2$ (red). The green line depicts the approximation N/ESS.

Monte Carlo algorithms discussed in Chapter 5 construct the importance densities in a sequential manner.

3.4 Metropolis-Hastings Algorithm

The Metropolis-Hastings (MH) algorithm belongs to the class of Markov chain Monte Carlo (MCMC) algorithms. The algorithm generates a Markov chain such that the stationary distribution associated with the Markov chain is unique and equals the posterior distribution of interest.

3.4.1 A Generic MH Algorithm

Key ingredient of the MH algorithm is a proposal distribution $q(\vartheta|\theta^{i-1})$, which potentially depends on the draw θ^{i-1} in iteration $i-1$ of the algorithm. The proposed draw is always accepted if it raises the posterior density (relative to θ^{i-1}) and it is sometimes accepted even if it lowers the posterior density. If the proposed draw is not accepted, then the chain does not move and $\theta^i = \theta^{i-1}$. The acceptance probability is chosen to ensure that the distribution of the draws converges to the target posterior distribution. The algorithm takes the following form:

Algorithm 4 (Generic MH Algorithm) *For $i = 1$ to N:*

1. *Draw ϑ from a density $q(\vartheta|\theta^{i-1})$.*
2. *Set $\theta^i = \vartheta$ with probability*

$$\alpha(\vartheta|\theta^{i-1}) = \min\left\{1, \frac{p(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{i-1})}{p(Y|\theta^{i-1})p(\theta^{i-1}))/q(\theta^{i-1}|\vartheta)}\right\}$$

and $\theta^i = \theta^{i-1}$ otherwise.

Because $p(\theta|Y) \propto p(Y|\theta)p(\theta)$ we can replace the posterior densities in the calculation of the acceptance probabilities $\alpha(\vartheta|\theta^{i-1})$ by the product of likelihood and prior, which does not require the evaluation of the marginal data density $p(Y)$.

Algorithm 4 describes how to generate a parameter draw θ^i conditional on a parameter draw θ^i. Thus, implicitly it characterizes a Markov transition kernel $K(\theta|\tilde{\theta})$, where the

conditioning value $\tilde{\theta}$ corresponds to the parameter draw from iteration i. The probability theory underlying the convergence of Monte Carlo averages constructed from the output of the MH algorithm is considerably more complicated than the theory for the importance sampler. The key questions are the following: (i) suppose that θ^0 is generated from some arbitrary density $g(\cdot)$ and θ^N is obtained by iterating the Markov transition kernel forward N times, then is it true that θ^N is approximately distributed according to $p(\theta|Y)$ and the approximation error vanishes as $N \longrightarrow \infty$? (ii) Suppose that (i) is true, is it also true that sample averages of θ^i, $i = 1, \ldots, N$ satisfy a SLLN and a CLT? For a comprehensive exposition of the convergence theory for Markov chains and MCMC algorithms, we refer the interested reader to textbook treatments such as Robert and Casella (2004) or Geweke (2005). In the remainder of this section we will briefly show why the posterior distribution $p(\theta|Y)$ is an invariant distribution for the Markov chain generated by Algorithm 4 and present a simple discrete example that in which we can analytically solve for the transition kernel of the Markov chain.

3.4.2 An Important Property of the MH Algorithm

For Algorithm 4 to generate a sequence of draws from the posterior distribution $p(\theta|Y)$ a necessary condition is that the posterior distribution is an invariant distribution under the transition kernel $K(\cdot|\cdot)$, that is,

$$p(\theta|Y) = \int K(\theta|\tilde{\theta}p(\tilde{\theta}|Y)d\tilde{\theta}. \tag{3.31}$$

Thus, if θ^{i-1} is a draw from the posterior distribution $p(\theta|Y)$ then θ^i is also a draw from this distribution.

Verifying the invariance property is relatively straightforward. The transition kernel can be expressed as follows:

$$K(\theta|\tilde{\theta}) = u(\theta|\tilde{\theta}) + r(\tilde{\theta})\delta_{\tilde{\theta}}(\theta). \tag{3.32}$$

Here $u(\theta|\tilde{\theta})$ is the density kernel (note that $u(\theta|\cdot)$ does not integrated to one) for accepted draws:

$$u(\theta|\tilde{\theta}) = \alpha(\theta|\tilde{\theta})q(\theta|\tilde{\theta}). \tag{3.33}$$

Recall from Algorithm 4 above that $q(\cdot|\cdot)$ is the density for the proposed draw and $\alpha(\cdot|\cdot)$ is the probability that the draw is accepted. The term $r(\tilde{\theta})$ is the probability that conditional

on $\tilde{\theta}$ the proposed draw will be rejected:

$$r(\tilde{\theta}) = \int \left[1 - \alpha(\theta|\tilde{\theta})\right] q(\theta|\tilde{\theta}) d\theta = 1 - \int u(\theta|\tilde{\theta}) d\theta. \tag{3.34}$$

If the proposed draw is rejected, then the algorithm sets $\theta^i = \theta^{i-1}$, which means that conditional on the rejection, the transition density degenerates to a pointmass at $\theta = \tilde{\theta}$, which is captured by the dirac function $\delta_{\tilde{\theta}}(\theta)$ in (3.32).[1]

The MH step is constructed to be reversible in the following sense. Conditional on the sampler not rejecting the proposed draw, the density associated with a transition from $\tilde{\theta}$ to θ is identical to the density associated with a transition from θ to $\tilde{\theta}$:

$$\begin{aligned} p(\tilde{\theta}|Y) u(\theta|\tilde{\theta}) &= p(\tilde{\theta}|Y) q(\theta|\tilde{\theta}) \min \left\{ 1, \frac{p(\theta|Y)/q(\theta|\tilde{\theta})}{p(\tilde{\theta}|Y)/q(\tilde{\theta}|\theta)} \right\} \\ &= \min \left\{ p(\tilde{\theta}|Y) q(\theta|\tilde{\theta}), p(\theta|Y) q(\tilde{\theta}|\theta) \right\} \\ &= p(\theta|Y) q(\tilde{\theta}|\theta) \min \left\{ \frac{p(\tilde{\theta}|Y)/q(\tilde{\theta}|\theta)}{p(\theta|Y)/q(\theta|\tilde{\theta})}, 1 \right\} \\ &= p(\theta|Y) u(\tilde{\theta}|\theta). \end{aligned} \tag{3.35}$$

Using the reversibility result, we can now verify the invariance property in (3.31):

$$\begin{aligned} \int K(\theta|\tilde{\theta}) p(\tilde{\theta}|Y) d\tilde{\theta} &= \int p(\tilde{\theta}|Y) u(\theta|\tilde{\theta}) d\tilde{\theta} + \int p(\tilde{\theta}|Y) r(\tilde{\theta}) \delta_{\tilde{\theta}}(\theta) d\tilde{\theta} \\ &= \int p(\theta|Y) u(\tilde{\theta}|\theta) d\tilde{\theta} + p(\theta|Y) r(\theta) \\ &= p(\theta|Y) \end{aligned} \tag{3.36}$$

The second equality follows from (3.35) and the properties of the dirac function. The last equality follows from (3.34).

The invariance property in (3.32) is by no means sufficient to guarantee that the Monte Carlo average of draws $h(\theta^i)$ from Algorithm 4) converges to the posterior expectation $\mathbb{E}_\pi[h]$. In particular, one needs to ensure that the transition kernel $K(\cdot|\cdot)$ has a unique invariant distribution, that repeated application of the transition kernel leads to convergence to the unique invariant distribution regardless of the chain's initialization, and that the persistence of the draws θ^i generated by the Markov chain is not so strong such that sample averages do not converge to population means. Rather than providing a general treatment of convergence, we will examine a specific example, in which we can solve for the transition kernel analytically.

[1] The dirac function has the property that $\delta_{\tilde{\theta}}(\theta) = 0$ for $\theta \neq \tilde{\theta}$ and $\int \delta_{\tilde{\theta}}(\theta) d\theta = 1$.

3.4.3 An Analytical Example

Suppose the parameter space is discrete and θ can only take two values: τ_1 and τ_2. The posterior distribution then simplifies to two probabilities which we denote be $\pi_l = \mathbb{P}\{\theta = \tau_l|Y\}$, $l = 1, 2$. The proposal distribution in Algorithm 4 can be represented as a two stage Markov process with transition matrix

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}, \tag{3.37}$$

where q_{lk} is the probability of drawing $\vartheta = \tau_k$ conditional on $\theta^i = \tau_l$. For illustrative purposes, we will assume that

$$q_{11} = q_{22} = q, \quad q_{12} = q_{21} = 1 - q.$$

We can now derive a transition matrix for the Markov chain generated by Algorithm 4. Suppose that $\theta^{i-1} = \tau_1$. Then with probability q, $\vartheta = \tau_1$. The probability that this draw will be accepted is

$$\alpha(\tau_1|\tau_1) = \min\left\{1, \frac{\pi_1/q}{\pi_1/q}\right\} = 1.$$

With probability $1 - q$ the proposed draw is $\vartheta = \tau_2$. The probability that this draw will be rejected is

$$1 - \alpha(\tau_2|\tau_1) = 1 - \min\left\{1, \frac{\pi_2/(1-q)}{\pi_1/(1-q)}\right\} = 0.$$

Thus, the probability of a transition from $\theta^{i-1} = \tau_1$ to $\theta^i = \tau_1$ is equal to q. Using similar calculations and assuming that

$$\pi_2 > \pi_1,$$

it can be verified that the Markov transition matrix for the process $\{\theta^i\}_{i=1}^N$ is given by

$$K = \begin{bmatrix} q & (1-q) \\ (1-q)\frac{\pi_1}{\pi_2} & q + (1-q)\left(1 - \frac{\pi_1}{\pi_2}\right) \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}. \tag{3.38}$$

Straightforward calculations (see, for instance, Hamilton (1994)) reveal that the transition matrix K has two eigenvalues λ_1 and λ_2:

$$\lambda_1(K) = 1, \quad \lambda_2(K) = q - (1-q)\frac{\pi_1}{1-\pi_1}. \tag{3.39}$$

The eigenvector associated with $\lambda_1(P)$ determines the invariant distribution of the Markov chain, which, as we have seen in Section 3.4.2, equals the posterior distribution. Provided

that the second eigenvalue is different from one, the posterior is the unique invariant distribution of the Markov chain.

The persistence of the Markov chain depends on the shape of the proposal distribution in relation to the posterior distribution. In the discrete case, one could easily obtain an *iid* sample from the posterior by setting $q = \pi_1$. While in general it is not feasible to tailor the proposal density to generate serially uncorrelated draws, the goal of MCMC design is to keep the persistence of the chain as low as possible. If q is equal to one, then $\theta^i = \theta^1$ for all i and the equilibrium distribution of the chain is no longer unique. Taking averages of the θ^i's will no longer yield a consistent estimate of its posterior mean. If $q = 0$, the equilibrium distribution remains unique, but a draw of $\theta^i = \tau_1$ is followed by a draw of $\theta^{i+1} = \tau_2$. The subsequent transition to θ^{i+2} is stochastic and the expected number of τ_2 draws is equal to π_2/π_1.

As in Section 3.3, we will now examine the convergence of Monte Carlo averages of $h(\theta^i)$. To do so, we define the transformed parameter

$$\xi^i = \frac{\theta^i - \tau_1}{\tau_2 - \tau_1}. \tag{3.40}$$

This transformed parameter takes the values 0 or 1. We can represent the Markov chain associated with ξ^i as first-order autoregressive process

$$\xi^i = (1 - k_{11}) + \lambda_2(K)\xi^{i-1} + \nu^i. \tag{3.41}$$

Conditional on $\xi^{i-1} = j$, $j = 0, 1$, the innovation ν^i has support on k_{jj} and $(1 - k_{jj})$, its conditional mean is equal to zero, and its conditional variance is equal to $k_{jj}(1-k_{jj})$. Based on this autoregressive representation, it is straightforward to compute the autocovariance function of ξ^i, which then can be converted into the autocovariance function of $h(\theta^i)$:

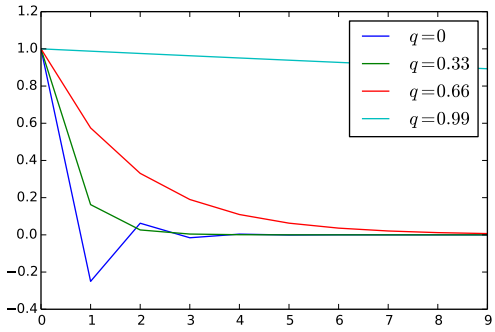
$$\begin{aligned} COV(h(\theta^i), h(\theta^{i-s})) &= \big(h(\tau_2) - h(\tau_1)\big)^2 \pi_1(1 - \pi_1)\left(q - (1 - q)\frac{\pi_1}{1 - \pi_1}\right)^s \\ &= \mathbb{V}_\pi[h]\left(q - (1 - q)\frac{\pi_1}{1 - \pi_1}\right)^s \end{aligned} \tag{3.42}$$

If $q = \pi_1$ then the autocovariances are equal to zero and the draws $h(\theta^i)$ are serially uncorrelated (in fact, in our simple discrete setting they are also independent). Defining the Monte Carlo estimate

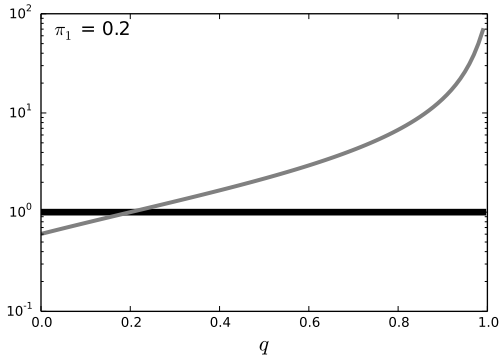
$$\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(\theta^i) \tag{3.43}$$

Figure 3.3: Discrete MH Algorithm

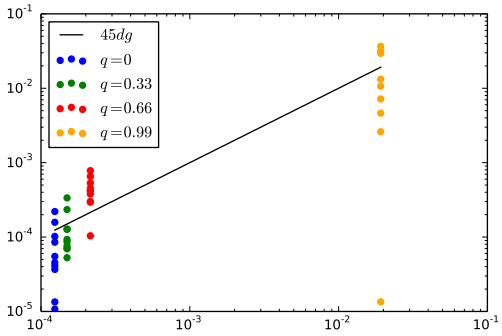
Autocorrelation Function



Relative Variance



Small Sample Variance versus HAC Estimates



Notes: Panel 1 depicts the autocorrelation function of θ^i. Panel 2 depicts the variance ratio $\bar{V}(\bar{\theta})/\mathbb{V}_\pi[\theta]$. of the small sample variance $\hat{V}(\bar{\theta})$ computed across multiple chains (x-axis) versus HAC estimates of $\bar{V}(\bar{\theta})/N$ (y-axis) computed for each chain.

we deduce from a central limit theorem for dependent random variables that

$$\sqrt{N}(\bar{h}_N - \mathbb{E}_\pi[h]) \Longrightarrow N\big(0, \bar{V}(\bar{h})\big), \tag{3.44}$$

where $\bar{V}(\bar{h})$ is the long-run covariance matrix

$$\bar{V}(\bar{h}) = \lim_{N \longrightarrow \infty} \mathbb{V}_\pi[h] \left(1 + 2 \sum_{s=1}^{N} \frac{N-s}{N} \left(q - (1-q) \frac{\pi_1}{1-\pi_1} \right)^s \right).$$

In turn, a measure of effective sample size for the MH algorithm can be defined as

$$ESS = \frac{N}{1 + 2 \sum_{s=1}^{N} \frac{N-s}{N} \left(q - (1-q) \frac{\pi_1}{1-\pi_1} \right)^s}. \tag{3.45}$$

3.4.4 A Numerical Illustration

We proceed with a numerical illustrations of the issues at hand when designing MCMC algorithms. We parameterize the example as a Bernoulli distribution ($\tau_1 = 0, \tau_2 = 1$) with $\pi_1 = 0.2$. To assess the effectiveness of different MH settings, we vary $q \in [0, 1)$. The top panel of Figure 3.3 displays the autocorrelations up to 9 lags for $q = \{0, 0.33, 0.66, 0.99\}$. When $q = 0.99$ (the turquoise line) the chain generated by the MH algorithm is extremely autocorrelated. As discussed, the probability of moving from $\theta^{i-1} = \tau_1$ to $\theta^i = \tau_2$ is $1 - q$, or 0.01. Similarly, the probability of moving from $\theta^{i-1} = \tau_2$ to $\theta^i = \tau_1$ is $(1-q)\pi_1/\pi_2 = 0.0025$. Thus, if the initial draw is $\theta^0 = \tau_1$, one would expect 100 draws before encountering τ_2. However, recall that 80% of the realized draws from the invariant distribution should be τ_2.

Intuitively, the high autocorrelation reflects the fact that it will take a high number of draws to accurately reflect the target distribution, or that the chain is "moving" extremely slowly around the parameter space. This will manifest itself in a high variance of Monte Carlo estimates, as we will see below. When $q = 0.66$ or 0.33 (the green and red lines, respectively), the autocorrelation is substantially weaker than under the $q = 0.99$ sampler. Still, both exhibit positive autocorrelation. Intuitively, when $\theta^{i-1} = \tau_1$, both the samplers will select $\theta^i = \tau_1$ with probability greater than $\pi_1 = 0.2$, inducing a positive autocorrelation in the chain. Finally, when $q = 0$ (the blue line), the MH chain actually has a *n*egative first order autocorrelation. For $\theta^{i-1} = \tau_1$ the probability of τ_1 for θ^i is zero, which is much less than one would expect under *iid* draws. Induced negative autocorrelation can actually serve to reduce Monte Carlo variance relative to theoretical variance, which the next panel highlights.

The middle panel computes the relative variance ratios, $\bar{V}(\bar{h})/\mathbb{V}_\pi[h]$ when q is varied in $[0, 1)$. The grey line shows the relative variance while the black horizontal bar indicates a relative variance of one. The y coordinates are rescaled in log terms. Consistent with the autocorrelations discussed above, for large values of q, the variance of Monte Carlo estimates of h drawn from the MH chain are much larger than the variance of estimates derived from *iid* draws. Indeed, when $q = 0.99$ the variance is about 100 times larger. As q moves closer to π_1 the relative variance shrinks. Indeed, when $q = \pi_1$ the Monte Carlo estimates from the MH sampler and an *iid* sampler have the same variance, as the chain generated by the MH sampler mimics the *iid* sampler. Finally, when $q < \pi_1$, the Monte Carlo variance from the MH sampler is less than that under *iid* draws. While this reduction in MC variance is obviously desirable, a few points should be kept in mind as we move forward. First, the design of a good MH sampler–here, this amounts to picking q–is highly depended on the target distribution, here indexed by π_1. Unfortunately, the reason one often resorts to MCMC techniques is that they don't know important features of the target distribution, i.e, π_1. Second and relatedly, measures such as the relative variance are often impossible to compute (as one doesn't generally know π_1), so instead analysts typically rely on measures like sample autocorrelation to get a sense of performance an MH algorithm.

In an environment where asymptotic variances are not known in closed-form, it is difficult to know when the chain generated by an MH algorithm has converged. There are many diagnostics available for this, some of which we will discuss in more detail in the next section. At the heart of most of the measures, though, is whether the empirical variability of an estimate computed across many runs an MH sampler is consistent with estimates within each chain. With an eye towards between and within chain measurement, we run 10 replications of MH samplers for $q = \{0, 0.33, 0.66, 0.99\}$. The length of each simulation is $N = 1000$. We set $h(\theta) = \theta$; i.e., we are interested in the variance of Monte Carlo estimates of the mean of the distribution. For each replication, we compute an estimate of $\bar{V}(\bar{h})/N$, using a simple Newey-West heteroskedastic- and autocorrelation-consistent (HAC) estimator,

$$HAC[\bar{h}] = \frac{\hat{\gamma}_0 + 2\sum_{l=1}^{L}\left(1 - \frac{l}{L+1}\right)\hat{\gamma}_l}{N},$$

where $\gamma_l = COV(h(\theta^i), h(\theta^{i-l}))$, with L set to 400. We also compute an estimate of the variance of h across the 10 replications. Indexing the replications by j, for each choice of q we can simply compute:

$$\hat{V}(\bar{h}) = VAR(\{\bar{h}_j\}_{j=1}^{10}).$$

The bottom panel of Figure 3.3 examine the relationships between these two estimates. The y-coordinate of the colored dots represent the HAC estimates for each q, while the x-coordinate gives the value of small sample variance of $\hat{V}(\bar{h})$ for each q. The black line gives the 45 degree line. One can see that relative ordering for the qs is preserved in small samples, with $q = 0$ having the lowest small-sample variance and $q = 0.99$ having the highest. More importantly, the small-sample variance for each of the simulators is bracketed by the HAC estimates, indicating by the black line bisecting the dots for each q. That is, the within chain estimates appear consistent with the across chain measures.

3.5 Bayesian Inference and Decision Making

The posterior distribution of the model parameters can be used for inference and decision making. From a Bayesian perspective it is optimal to make decisions that minimize the posterior expected loss of the decision maker. It turns out that many inferential problems, e.g., point or interval estimation, can be restated as decision problems. In general, there is a decision rule $\delta(Y)$ that maps the observations Y into decisions, and a loss function $L(\theta, \delta)$ or $L(y_*, \delta)$ according to which decisions are evaluated. The loss functions may either depend on model parameters, e.g., θ, or on future or counterfactual values of y_t, which we denoted by y_*, or it could depend on both. For the remainder of this section we assume that the loss depends on the parameter θ.

The posterior expected loss associated with a decision $\delta(Y)$ is given by

$$\rho\big(\delta(Y)|Y\big) = \int_{\Theta} L\big(\theta, \delta(Y)\big) p(\theta|Y) d\theta. \tag{3.46}$$

Note that in this calculation the observations Y are fixed and we are integrating over the unknown parameter θ under the posterior distribution. A Bayes decision is a decision that minimizes the posterior expected loss:

$$\delta^*(Y) = \text{argmin}_{\delta}\ \rho\big(\delta|Y\big). \tag{3.47}$$

Because all calculations are conditional on Y, we simply write δ instead of $\delta(Y)$ from now on. For some decision problems, e.g., point estimation under a quadratic loss function (see below), it is possible to solve for δ^* analytically, expressing the optimal decision as a function of moments or quantiles of the posterior distribution of θ. A Monte Carlo approximation can then be used to evaluate δ. For other decision problems it might not be feasible to derive δ^*

analytically. In these case one can replace the posterior risk for each choice of δ by a Monte Carlo approximation of the form

$$\bar{\rho}_N\big(\delta|Y\big) = \frac{1}{N}\sum_{i=1}^{N} L\big(\theta^i, \delta\big), \tag{3.48}$$

where the θ^i's are draws from the posterior $p(\theta|Y)$. If the draws are generated by importance sampling, then the losses have to be reweighted using the importance weights as in (3.26). A numerical approximation to the Bayes decision $\delta^*(\cdot)$ is then given by

$$\delta_N^*(Y) = \text{argmin}_d\ \bar{\rho}_N\big(\delta(\cdot)|Y\big). \tag{3.49}$$

According to the frequentist large sample theory for extremum estimators (see for instance the textbook treatment in van der Vaart (1998)), $\delta_N^*(Y) \xrightarrow{a.s.} \delta^*(Y)$ provided that $\bar{\rho}_N\big(\delta|Y\big)$ converges to $\rho\big(\delta|Y\big)$ uniformly in δ and $N \longrightarrow \infty$.

3.5.1 Point Estimation

Suppose that θ is scalar. The most widely used loss functions are the quadratic loss function $L_2\big(\theta, \delta\big) = (\theta - \delta)^2$ and the absolute error loss function $L_1(\theta, \delta) = |\theta - \delta|$. The Bayes estimator associated with the quadratic loss function is the posterior mean $\mathbb{E}_\pi[\theta]$ which can be approximated by the Monte Carlo average

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N} \theta^i. \tag{3.50}$$

The Bayes estimator associated with the absolute error loss function is the posterior median. More generally, the τ sample quantile can be obtained by solving the problem (see Koenker (2005) for a textbook treatment of quantile regressions)

$$\hat{\theta}_\tau = \text{argmin}_q \left[(1-\tau)\frac{1}{N}\sum_{\theta^i < q}(q - \theta^i) + \tau\frac{1}{N}\sum_{\theta^i \geq q}(\theta^i - q) \right]. \tag{3.51}$$

An easy way to compute the solution to this problem is to sort the draws in ascending order and let $\hat{\theta}_\tau$ the $\lfloor \tau N \rfloor$'th draw.

3.5.2 Interval Estimation

An interval estimator (credible interval) for a scalar parameter θ consists of a lower bound δ_l and an upper bound δ_u. Let $\delta = [\delta_l, \delta_u]'$ and consider the loss function

$$L(\theta, \delta) = \max_{\lambda \in \mathbb{R}^-} (\delta_u - \delta_l) + \lambda\big(\mathcal{I}\{\delta_l \leq \theta \leq \delta_u\} - (1-\alpha)\big), \tag{3.52}$$

where $\mathcal{I}\{x \leq a\}$ is the indicator function that equals one if $x \leq a$ and is zero otherwise. Note that if $\delta_l \leq \theta \leq \delta_u$ then the factor post-multiplying λ is positive for $\alpha > 0$ and the solution to the constrained maximization problem is to set $\lambda = 0$. Thus, the loss corresponds to the length of the interval. If θ lies outside of the interval, then the loss is infinite. The posterior risk is given by

$$\rho(\delta_l(Y), \delta_u(Y)|Y) = (\delta_u - \delta_l) + \max_{\lambda \in \mathbb{R}^-} \lambda\big(\mathbb{P}(\delta_l \leq \theta \leq \delta_u|Y) - (1-\alpha)\big). \tag{3.53}$$

If the posterior density $p(\theta|Y)$ is unimodal, then the credible interval that minimizes this loss function has the property that $p(\delta_l|Y) = p(\delta_u|Y) = \kappa$. It is called the highest posterior density (HPD) set because the density of all values of θ that are included in this set exceeds the threshold κ. If the posterior-density is multimodal the interval that minimizes the posterior expected loss in (3.53) is not necessarily the HPD set. The HPD set may be the union of multiple disjoint intervals (constructed around the modes) and is formally defined as $CS_{HPD} = \big\{\theta \ | p(\theta|Y) \geq \kappa\big\}$. The threshold κ is chosen to guarantee that the set has a $1-\alpha$ coverage probability.

In practice researchers often replace HPD sets by equal-tail-probability sets that satisfy

$$\int_{-\infty}^{\delta_l} p(\theta|Y)d\theta = \int_{\delta_u}^{\infty} p(\theta|Y)d\theta = \alpha/2.$$

While these intervals tend to be longer than HPD intervals, they are easier to compute because δ_l and δ_u are simply the $\alpha/2$ and $1-\alpha_2$ quantiles which can be obtained from (3.51).

3.5.3 Forecasting

In forecasting applications the argument θ of the loss function is replaced by a future observation y_{T+h}: $L(y_{T+h}, \delta)$. For the AR(1) model in (3.3) we can express

$$y_{T+h} = \theta^h y_T + \sum_{s=0}^{h-1} \theta^s u_{T+h-s}, \tag{3.54}$$

which implies that the h-step ahead conditional distribution is

$$y_{T+h}|(Y_{1:T},\theta) \sim N\left(\theta^h y_T, \frac{1-\theta^h}{1-\theta}\right). \tag{3.55}$$

We can express the predictive density of y_{T+h} as

$$p(y_{T+h}|Y_{1:T}) = \int p(y_{T+h}|y_T,\theta)p(\theta|Y_{1:T})d\theta. \tag{3.56}$$

Draws from this predictive density can be easily generated with the following algorithm:

Algorithm 5 (Sampling from Predictive Distribution) *For each draw θ^i from the posterior distribution $p(\theta|Y_{1:T})$ sample a sequence of innovations $u^i_{T+1},\ldots,u^i_{T+h}$ and compute y^i_{T+h} as a function of θ^i, $u^i_{T+1},\ldots,u^i_{T+h}$, and $Y_{1:T}$, e.g., according to (3.54).*

Moments and quantiles of the predictive distribution can be approximated based on the draws y^i_{T+h}. The posterior expected loss is given by

$$\rho\big(\delta|Y_{1:T}\big) = \int_{y_{T+h}} L(y_{T+h},\delta)p(y_{T+h}|Y_{1:T})dy_{T+h} \tag{3.57}$$

and under suitable regularity conditions can be approximated by the Monte Carlo average

$$\bar{\rho}(\delta|Y) = \frac{1}{N}\sum_{i=1}^{N} L(y^i_{T+h},\delta). \tag{3.58}$$

3.5.4 Model Selection and Averaging

The posterior probablities for a collection of M_j, $j = 1,\ldots,J$ are given by $\gamma_{j,T}$ defined in (3.12). The key difficulty in computing posterior model probabilities is the evaluation of the marginal data density $p(Y|M_j)$, which we will discuss in more detail in subsequent chapters. Once the posterior probabilities have been obtained, they can be used for model selection or averaging. Bayesian model selection typically refers to the solution of a decision problem in which the loss associated with selecting the correct model is zero and the loss associated with choosing an incorrect model is one. It can be verified that the solution that minimizes the posterior expected loss is to select the model with the highest posterior probability. Model averaging refers to a procedure in which posterior distributions from a single model are replaced by the mixture of distribution obtained by averaging across all

available models, using the posterior model probabilities as weights. Suppose that θ is a parameter common to all models then we can form

$$p(\theta|Y) = \sum_{j=1}^{J} \gamma_{j,T} p(\theta|Y, M_j). \tag{3.59}$$

Similarly, the predictive distribution for a future observation y_{T+h} takes the form

$$p(y_{T+h}|Y_{1:T}) = \sum_{j=1}^{J} \gamma_{j,T} p(y_{T+h}|Y_{1:T}, M_j). \tag{3.60}$$

Part II

Bayesian Computations for Linearized DSGE Models

Chapter 4

Metropolis-Hastings Algorithms for DSGE Models

To date, the most widely used method to generate draws from posterior distributions of a DSGE model is the random walk MH (RWMH) algorithm. This algorithm is a special case of the generic Algorithm 4 in which the proposal distribution $q(\vartheta|\theta^{i-1})$ can be expressed as the random walk $\vartheta = \theta^{i-1} + \eta$ and η is drawn from a distribution that is centered at zero. We will introduce a benchmark RWMH algorithm in Section 4.1 and apply it to a small-scale New Keynesian DSGE model in Section 4.2. In combination with the a prior distribution that is typically used for this kind of model the posterior distribution has a well-behaved elliptical shape and the output from the simple RWMH algorithm can be used to obtain accurate numerical approximations of posterior moments.

Unfortunately, in many applications, in particular those involving medium- and large-scale DSGE models the posterior distributions could be very non-elliptical. Irregularly shaped posterior distributions are often caused by identification problems. The DSGE model may suffer from a local identification problem that generates posteriors that are very flat in certain directions of the parameter space, at least locally in the neighborhood of the mode, similar to the posterior encountered in the simple set-identified model of Section 3.2. Alternatively, the posterior may exhibit multimodal features. Multimodality could be caused by the data's inability to distinguish between the role of a DSGE model's external and internal propagation mechanisms. For instance, inflation persistence, can be generated by highly autocorrelated cost-push shocks or by firms' inability to frequently re-optimize their prices

in view of fluctuating marginal costs. We use a very stylized state-space model to illustrate these challenges for posterior simulator in Section 4.3.

In view of the difficulties caused by irregularly-shaped posterior surfaces, we review a variety of alternative MH samplers in Section 4.4. These algorithms differ from the RWMH algorithm in two dimensions. First, they use alternative proposal distributions $q(\vartheta|\theta^{i-1})$. In general, we consider distributions of the form

$$q(\cdot|\theta^{i-1}) = p_t(\cdot|\mu(\theta^{i-1}), \Sigma(\theta^{i-1}), \nu),$$

where $p_t(\cdot)$ refers to the density of a student-t distribution Our exploration of different qs will thus concentrate on different ways of forming the location parameter $\mu(\cdot)$ and the scale matrix $\Sigma(\cdot)$. For $\nu = \infty$ this notation nests Gaussian proposal distributions. The second dimension in which we generalize the algorithm is blocking, i.e., we group the parameters into subvectors, and use a Block MH sampler to draw iteratively from conditional posterior distributions.

While the alternative MH samplers are designed for irregular posterior surfaces for which the simple RWMH algorithm generates inaccurate approximations, we illustrate the performance gains obtained through these algorithms using the simple New Keynesian DSGE model in Section 4.5. Similar to the illustrations in Section 3.4, we evaluate the accuracy of the algorithms by computing the variance of Monte Carlo approximations across multiple chains. Our simulations demonstrate that careful tailoring of proposal densities $q(\vartheta|\theta^{i-1})$ as well as blocking of the parameters can drastically improve the accuracy of Monte Carlo approximations. In Section 3.4, we showed directly that the Monte Carlo estimates associated with the discrete MH algorithm satisfied a SLLN and CLT for dependent, identically distributed random variables. All of the MH algorithms here give rise to Markov chains that are (recurrent,) irreducible and aperiodic for the target distribution of interest. These properties are sufficient for a SLLN to hold. However, validating conditions for a CLT to hold is much more difficult and beyond the scope of this book.

Finally, Section 4.6 takes a brief look at the numerical approximation of marginal data densities that are used to compute posterior model probabilities.

4.1 A Benchmark Algorithm

The most widely used MH algorithm in DSGE model applications is the *random walk MH* (RWMH) algorithm. The mean of the proposal distribution is simply the current location

in the chain, with its variance prespecified,

$$\mu(\theta_{t-1}) = \theta_{t-1} \text{ and } \Sigma(\theta_{t-1}) = c^2\hat{\Sigma} \tag{4.1}$$

The name of the algorithm comes from the random walk form of the proposal, which can be written as

$$\vartheta = \theta^{i-1} + \eta$$

where ϵ is mean zero with variance $c^2\hat{\Sigma}$. Given the symmetric nature of the proposal distribution, the acceptance probability becomes

$$\alpha = \min\left\{\frac{p(\vartheta|Y)}{p(\theta^{i-1}|Y)}, 1\right\}.$$

A draw, ϑ, is accepted with probability one if the posterior at ϑ has a higher value than the posterior at θ^{i-1}. The probability of acceptance decreases as the posterior at the candidate value decreases relative to the current posterior.

To implement the RWMH, the user still needs to specify ν, c, and $\hat{\Sigma}$. For all of the variations of the RWMH we implement, we set $\nu = \infty$, that is, we use a multivariate normal proposal distribution in keeping with most of the literature. Typically, the choice of the c is made conditional on $\hat{\Sigma}$, so we first discuss the choice for $\hat{\Sigma}$. The proposal variance controls the relative variances and correlations in the proposal distribution. As we have seen in Section 3.4, the sampler can work very poorly if q is strongly at odds with the target distribution. This intuition extends to the multivariate setting here. Suppose we our vector of parameters θ, contains two parameters, say β and δ, that are highly correlated in the posterior distribution. If the variance of the proposal distribution does not capture this correlation, but instead characterizes β and δ as independent–by, for example, using a diagonal matrix for $\hat{\Sigma}$–then the proposal ϑ is unlikely to reflect the fact that when β is large δ is large, and so on. This means that $p(\vartheta|Y)$ is likely to be small, and so the draw will be rejected. The chain will have many rejections, and consequently c driven be tuned to small values. The chain generated by this algorithm will be very highly autocorrelated and thus the Monte Carlo estimates derived from it will have high variance.

A good choice for $\hat{\Sigma}$ seeks to incorporate information from the posterior, to potentially capture correlations discussed above. Obtaining this information can difficult, as one is necessarily running MCMC because there is not much information about the posterior. A popular approach, used in Schorfheide (2000), is to set $\hat{\Sigma}$ to be the negative of the inverse Hessian at the mode of the log posterior, $\hat{\theta}$, obtained by running an numerical optimization

routine before running MCMC. Using this as an estimate for the covariance of the posterior is attractive, because it can be viewed as a large sample approximation to the posterior covariance matrix as the sample size $T \longrightarrow \infty$. There exists a large literature on the asymptotic normality of posterior distributions. Fundamental conditions can be found, for instance, in Johnson (1970).

Unfortunately, in many applications the maximization of the posterior density is tedious and the numerical approximation of the Hessian may be inaccurate. These problems may arise if the posterior distribution is very non-elliptical and possibly multi-modal, or if the likelihood function is replaced by a non-differentiable particle filter approximation (see Chapter 7 below). In these cases, a (partially) adaptive approach may work well: First, generate a set of posterior draws based on a reasonable initial choice for $\hat{\Sigma}$, e.g. the prior covariance covariance matrix. Second, compute the sample covariance matrix from the first sequence of posterior draws and use it as $\hat{\Sigma}$ in a second run of the RWMH algorithm. In principle, the covariance matrix $\hat{\Sigma}$ can be adjusted more than once. However, $\hat{\Sigma}$ must be fixed for the validity of the algorithm to hold. Samplers which constantly (or automatically) adjust $\hat{\Sigma}$ are known as adaptive samplers and require substantially different theoretical justifications.

Instead of strictly following one of the two approaches that we just described, in many of the numerical illustrations below, we use an estimate of the posterior covariance, V, obtained from an earlier estimation. While this approach is impractical in general, it avoids an mismatch between the Hessian-based estimate and the posterior covariance – in some sense, it is a best-case scenario. To summarize, we examine the following variant of the RWMH algorithm:

$$\text{RWMH-V} \quad : \quad \hat{\Sigma} \text{ is the posterior covariance.}$$

For comparison purposes we also consider setting $\hat{\Sigma}$ to the identity matrix I:

$$\text{RWMH-I} \quad : \quad \hat{\Sigma} = I$$

This is a generic approach, as it does not require any prior knowledge of the posterior to implement, which is attractive because this makes it easy to implement. In particular, this choice does not require a numerical maximization of the posterior, which may be difficult to execute if the posterior is very non-elliptical. The downside of choosing the identity matrix is that it ignores the scaling of the parameters and the orientation of the posterior contours. We will see in Section 4.5 below that this naive choice of proposal distribution covariance matrix

leads to a substantial deterioration of the efficiency of the algorithm. If the prior distribution is proper and the marginal distributions are appropriately scaled, then the identity matrix could be replaced by a diagonal matrix with the prior variances on the diagonal.

The final parameter of the algorithm is the scaling factor c. This parameter is typically adjusted to ensure a "reasonable" acceptance rate. Given the opacity of the posterior, it is difficult to derive a theoretically optimal acceptance rate. If the sampler accepts too frequently, it may be making very small movements, resulting in high Monte Carlo estimates. Similarly, if the chain rejects too frequently, it may be get stuck in one region of the parameter space, again resulting in poor estimates. However, for the special case of a target distribution which is multivariate normal, Roberts, Gelman, and W.R. (1997) has derived a limit (in the size of parameter vector) optimal acceptance rate of 0.234. Most practitioners target an acceptance rate between 0.20 and 0.40. The scaling c factor can be tuned during the burn-in period or via pre-estimation chains. We will discuss the relationship between the accuracy of Monte Carlo approximations and the choice of c in more detail in Section 4.5.1.

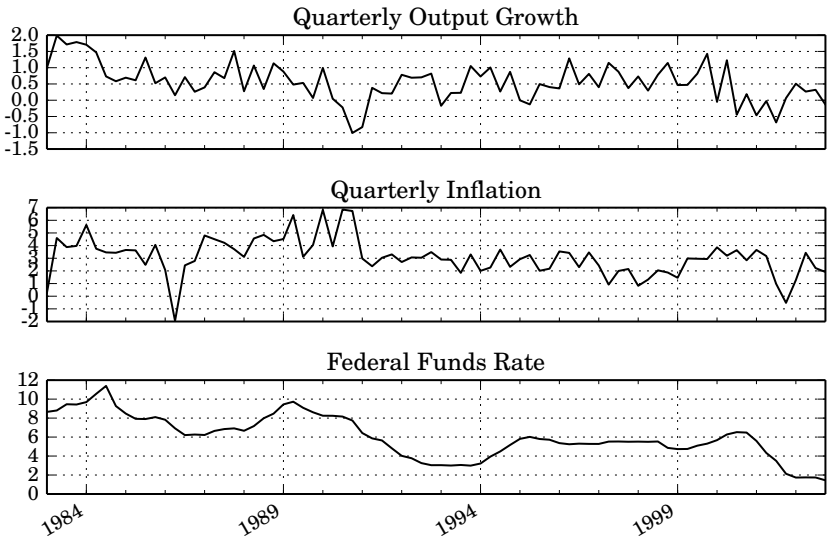
4.2 The RWMH-V Algorithm at Work

We will now apply the RWMH-V algorithm to the estimation of the small-scale New Keynesian model (DSGE Model I) introduced in Section 1.1. The model is solved using a log-linear approximation as described in Section 2.1. We begin with this example on account of its simplicity and because it has been previously studied in An and Schorfheide (2007b). We can thus be confident all of our samplers can converge to the posterior in reasonable time, allowing us to concentrate on the variance of the estimators as a measure of success. In later sections, we will examine more elaborate models where some simulators have trouble replicating key features of the posterior.

The model uses three observables to inform the estimation. We use quarterly per capita GDP growth, quarterly inflation, and the annualized federal funds rate, whose measurement equations were defined in Equation (2.11). The observations span from 1983:I to 2002:IV, giving us a total of 80 observations. The prior distribution was given in Table 2.2. Figure 4.1 plots the observables.

In general the initial draw θ^0 of an MH does not reflect the posterior. Indeed, there may be a large number of draws before the sampler has "converged"–that is when a draw is more or less indistinguishable from a draw from the posterior. For this reason, it is common practice

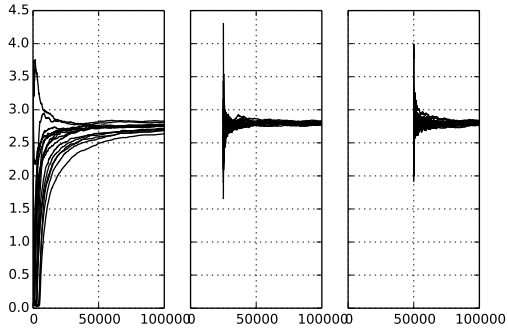
Figure 4.1: DSGE I Observables



to drop a substantial part (say the first N_0 draws) of the initial simulations of the MH chain, known as the "burn-in." Figure 4.2 depicts $\bar{\theta}(N_0, N) = \frac{1}{N-N_0}\sum_{i=1}\theta^i$ as a function of N for multiple runs of the RWMH-V algorithm and three choices of N_0. Initial draws are generated from the prior distribution. The dispersion of initial recursive mean after burn-in corresponds roughly to posterior variance to the extent that the chain converged to its equilibrium distribution after N_0 draws. Each recursive mean appears to approach the same limit point.

While the draws generated by the posterior simulator represent the joint posterior distribution of the parameter vector θ, researchers typically start out the empirical analysis by reporting summary statistics for the marginal posterior distribution of each parameter. Table 4.1 provides posterior mean parameter estimates and 90% credible intervals. Instead of computing HPD intervals, we report the 5th and the 95th percentile of the posterior distribution, which can be easily obtained after sorting the posterior draws for each parameter. The posterior estimates of the DSGE model parameters are broadly in line with estimates reported elsewhere in the literature. The estimated annualized steady state growth rate of the economy is 1.68%, the estimated steady state inflation rate for the sample period is 3.43%, and the implied steady state nominal interest rate is 5.56%. The estimated intertemporal elasticity of substitution $\widehat{1/\tau} = 0.37$. The estimated slope of the New Keynesian Phillips curve is fairly large, $\hat{\kappa} = 0.75$, implying a low degree of price rigidity. The central bank reacts

Figure 4.2: Convergence of Monte Carlo Averages



Notes: The figure depicts recursive means of τ^i for different choices of the burn-in sample size and multiple chains.

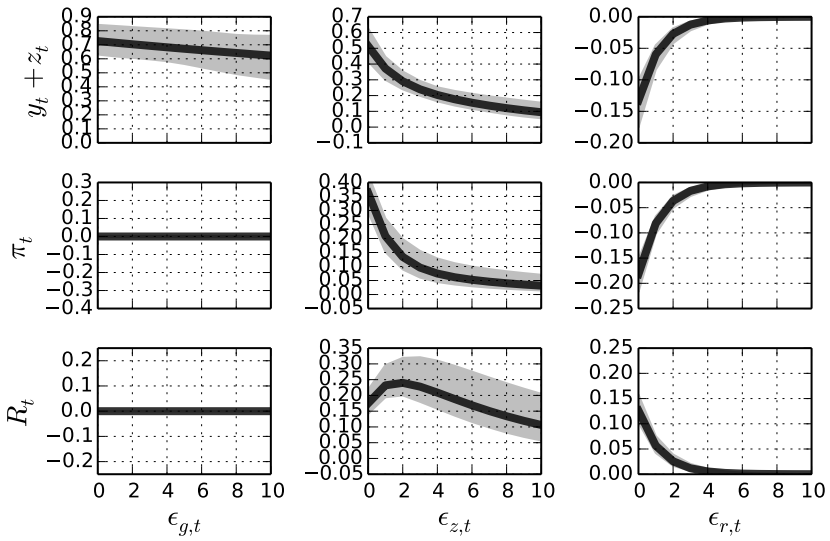
Table 4.1: Posterior Estimates of DSGE Model Parameters

Parameter	Mean	[5, 95]	Parameter	Mean	[5,95]
τ	2.72	[1.48, 3.87]	ρ_r	0.76	[0.69, 0.83]
κ	0.75	[0.37, 0.98]	ρ_g	0.95	[0.61, 1.00]
ψ_1	1.89	[1.41, 2.40]	ρ_z	0.87	[0.78, 0.92]
ψ_2	0.63	[0.20, 1.21]	σ_r	0.24	[0.18, 0.66]
$r^{(A)}$	0.45	[0.04, 1.05]	σ_g	0.79	[0.61, 1.57]
$\pi^{(A)}$	3.43	[2.79, 4.02]	σ_z	0.49	[0.26, 2.48]
$\gamma^{(Q)}$	0.42	[0.04, 0.73]			

strongly to inflation movements as well as deviations of output from flexible price output. In the remainder of this chapter, we will not focus on the posterior estimates *per se* but rather on the accuracy with which various posterior samplers can generate approximations of posterior moments.

The parameter draws can be transformed into other statistics of interest. For instance, the DSGE model can be used to study the propagation of exogenous shocks. Conditional on a parameter vector θ, it is straightforward to compute impulse response functions (IRFs) from the state-space representation of the DSGE model given by (2.9) and (2.11). The mapping from the parameters to the IRFs is an example of a function $h(\theta)$ that is of interest in many DSGE model applications. (Pointwise) Bayesian inference for IRFs can be implemented by

Figure 4.3: Impulse Response to a Monetary Policy Shock



Notes: The figure depicts pointwise posterior means and 90% credible bands.

first converting each draw θ^i into $h(\theta^i)$ and then computing posterior means and credible intervals for each element of the $h(\cdot)$ vector. Results for the small-scale DSGE model are depicted in Figure 4.3. Each column of the figure corresponds to the responses to the government spending shock, the technology growth shock, and the monetary policy shock, respectively; and each row corresponds to the response of output, inflation, and interest rates to the three shocks. The solid lines depict posterior mean responses and the shaded areas are 90% credible bands.

The log-linearized equilibrium conditions for the small-scale DSGE model were summarized in (2.1). A positive government spending (or, more generally, demand) shock raises output, but leaves inflation and interest rates unchanged. In this simple model consumption in deviations from the stochastic trend, $\hat{c}_t$ is the difference between output deviations $\hat{y}_t$ and the government spending shock $\hat{g}_t$. Moreover, $\hat{g}_t$ equals potential output, i.e. the output that would prevail in the absence of price rigidities, and $\hat{c}_t = \hat{y}_t - \hat{g}_t$ can be interpreted as the output gap. If the log-linearized equilibrium conditions are rewritten in terms of $\hat{c}_t$, then the government spending shock drops out of the Euler equation, the New Keynesian Phillips curve, and the monetary policy rule. This implies that the government spending shock only affects output, but not the output gap, i.e. consumption, inflation, and interest rate.

In response to a technology growth shock $\hat{z}_t$, output and consumption react proportionally,

i.e. $\hat{y} = \hat{c}_t$. While the level of output will adjust to the new level of technology in the long-run, expectations of increasing productivity lead agents to increase consumption initially by more than $\hat{z}_t$, meaning that $\hat{y}_t = \hat{c}_t > 0$. According to the Phillips curve, the positive output gap is associated with an increase in inflation, which in turn triggers a rise in interest rate. In the long-run, the levels of output and consumption rise permanently while both inflation and interest rates revert back to their steady states. Finally, an unanticipated increase in nominal interest rates raises the real rate because inflation is slow to adjust. According to the Euler equation, current consumption is minus the sum of future expected real rates, which means that consumption and output fall. According to the price setting equation, a drop in output and consumption leads to a fall in inflation.

Notes: (i) (ii)

4.3 Potential Irregularities in the Posterior

The posterior distribution associated with the small-scale New Keynesian model has an elliptical shape and the RWMH-V algorithm performs well. The advanced computational techniques that we will present subsequently are motivated by the observation that more elaborate DSGE models or DSGE models equipped with a more diffuse prior distribution may generate non-elliptical posterior distributions that are difficult to sample from. To illustrate the difficulties that may arise when generating draws from the posterior density $p(\theta|Y)$, consider the following stylized state-space model discussed in Schorfheide (2010):

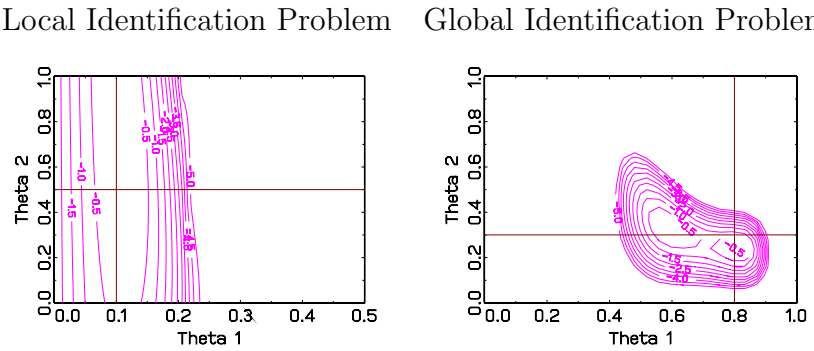
$$y_t = [1\ 1]s_t, \quad s_t = \begin{bmatrix} \phi_1 & 0 \\ \phi_3 & \phi_2 \end{bmatrix} s_{t-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \epsilon_t, \quad \epsilon_t \sim iidN(0,1). \tag{4.2}$$

The mapping between some structural parameters $\theta = [\theta_1, \theta_2]'$ and the reduced-form parameters $\phi = [\phi_1, \phi_2, \phi_3]'$ is assumed to be

$$\phi_1 = \theta_1^2, \quad \phi_2 = (1 - \theta_1^2), \quad \phi_3 - \phi_2 = -\theta_1\theta_2. \tag{4.3}$$

The first state, $s_{1,t}$, looks like a typical exogenous driving force of a DSGE model, e.g., total factor productivity, while the second state $s_{2,t}$ evolves like an endogenous state variable, e.g., the capital stock, driven by the exogenous process and past realizations of itself. The mapping from structural to reduced form parameters is chosen to highlight the identification problems endemic to DSGE models. First, θ_2 is not identifiable when θ_1 is close to 0, since it enters the model only multiplicatively. Second, there is a global identification problem. Root

Figure 4.4: Posteriors For Stylized State Space Model



cancelation in the AR and MA lag polynomials for y_t causes a bimodality in the likelihood function.

Figure 4.4 depicts posterior contours for two hypothetical posteriors. The contours in the left panel highlight the local identification problem that arises if θ_1 is close to zero. The contours in the left panel are based on a posterior for which we we simulate $T = 200$ observations given $\theta = [0.45, 0.45]'$. This parameterization is observationally equivalent to $\theta = [0.89, 0.22]'$. In both cases we use a prior distribution that is uniform on the square $0 \leq \theta_1 \leq 1$ and $0 \leq \theta_2 \leq 1$. For the MH algorithm to be efficient, the posterior on the left requires that the algorithm tries to make relatively large steps in the θ_2 direction and small steps in the θ_1 direction. This is achieved by aligning the contours of the proposal density with the contours of the posterior. Recall that in the benchmark algorithm we used a $\hat{\Sigma}$ that was constructed from (an approximation of) the posterior covariance matrix.

Sampling from the posterior depicted in the right panel is considerably more difficult, because the sampler has to travel from one modal region to the other, crossing a valley. This turns out to be difficult for the benchmark RWMH algorithm. Blocking, i.e., sampling from the posterior of $\theta_2|(\theta_1, Y)$ and $\theta_1|(\theta_2, Y)$ can help and so can a more careful tailoring of the proposal densities for the conditional distributions.

4.4 Alternative MH Samplers

The benchmark RMWH algorithm can be improved in two directions. First, one can tailor the proposal distribution to reduce the persistence in the Markov chain. We consider

two algorithms that have this feature: the Metropolis-Adjusted Langevin algorithm in Section 4.4.1 and the Newton MH algorithm in Section 4.4.2. This list is not exhaustive. For instance, Kohn, Giordani, and Strid (2010) propose an adaptive MH algorithm in which the proposal distribution is a mixture of a random walk proposal, an independence proposal, and a t-copula estimated from previous draws of the chain. While this is a promising approach, but it requires the user to specify a large set of tuning parameters, which may be daunting to the applied macroeconomist. Second, it is often helpful to split the parameters into blocks, and sample from the posterior distribution of each block, conditional on the most recent draws of all the other parameters. Block MH algorithms are discussed in Section 4.4.3.

4.4.1 Metropolis-Adjusted Langevin Algorithm

A natural evolution from the RWMH, which uses only the level of the (unnormalized) posterior, is the *Metropolis-Adjusted Langevin* (MAL) Algorithm, which incorporates the slope of the posterior. The MAL algorithm has a long history, dating back to Phillips and Smith (1994) and Roberts and Tweedie (1992). The MAL algorithm is based on the Langevin diffusion,

$$dX_t = dB_t + \frac{1}{2}\log p(dX_t|Y)dt.$$

X has $p(\cdot|Y)$ as its invariant distribution. The discretization of this process suggests a natural evolution of a Markov chain,

$$\theta^i = \theta^{i-1} + \frac{c_1}{2} \bigtriangledown \log p(\theta^{i-1}|Y) + c_2\epsilon.$$

Here c_1 and c_2 are scaling factors. Clearly, chains based on this evolution will not satisfy the posterior as their ergodic distribution, given the bias induced by the gradient term. It is easy to correct this bias by adding a MH step in the evolution. That is, treat the proposal distribution as $t(\mu(\theta_{t-1}), \Sigma(\theta_{t-1}), \nu)$ with

$$\mu(\theta_{t-1}) = \theta_{t-1} + \frac{c_1}{2} \bigtriangledown \log p(\theta_{t-1}|Y), \quad \Sigma(\theta_{t-1}) = c_2^2 I.$$

This defines a valid MH algorithm. Indeed, Roberts and Rosenthal (1998) show that the optimal rate of acceptance is 57% in the special case when the elements of θ are uncorrelated. The higher acceptance rate suggests improved statistical performance relative to the RW algorithm. Intuitively, the MAL algorithm pushes the chain toward regions of higher probability density, where most of the draws should lie.

Unfortunately, in a multidimensional setting, it becomes difficult to scale step size c_1 as parameters tend to have different magnitudes. Moreover, simply using the gradient ignores any potential relationship between the parameters, the knowledge of which is informative in any MCMC algorithm. It turns out–see, for example, Roberts and Stramer (2002)–that it is extremely helpful to *precondition* the MAL proposal, with

$$\mu(\theta_{t-1}) = \theta_{t-1} + \frac{c_1}{2} M_1 \bigtriangledown \log p(\theta_{t-1}|Y), \quad \Sigma(\theta_{t-1}) = c_2^2 M_2. \tag{4.4}$$

One standard practice is to set $M_1 = M_2 = M$, with

$$M = - \left[\frac{\partial^2 (\log p(\hat{\theta}|Y))}{\partial \theta \partial \theta'} \right]^{-1}. \tag{4.5}$$

To reiterate, $\hat{\theta}$ is the mode of the posterior. The use of the Hessian at the mode in a sense accounts for the "average" relationships between the parameters. If this is not changing much over the parameter space, then the preconditioned MAL (p-MAL) algorithm might be quite efficient. We examine the effectiveness of this algorithm. As with the RWMH-V algorithm, we abstract from the difference between the Hessian and the posterior covariance, V, and simply use in its place.

$$\text{MAL} : M_1 = M_2 = V.$$

4.4.2 Newton MH Algorithm

The connection between posterior simulation and Newtonian optimization is more closely exploited by Qi and Minka (2002), called *Newton MH*. Their algorithm can be seen as a MAL-type algorithm with,

$$\mu(\theta^{i-1}) = \theta^{i-1} - s \left[\frac{\partial^2 (\log p(\theta^{i-1}|Y))}{\partial \theta \partial \theta'} \right]^{-1} \bigtriangledown \log p(\theta^{i-1}|Y) \text{ and} \tag{4.6}$$

$$\hat{\Sigma}(\theta^{i-1}) = - \left[\frac{\partial^2 (\log p(\mu(\theta^{i-1})|Y))}{\partial \theta \partial \theta'} \right]^{-1}. \tag{4.7}$$

Here s is the step size of the Newton step. When the log posterior is quadratic, $s = 1$ is the optimal step size. If the log posterior is quadratic–the distribution is elliptical–then the proposal distribution will directly coincide with the target distribution and one will be perfectly sampling the posterior, θ^{i-1} and ϑ will be uncorrelated. Obviously, there are many departures from normality in the posterior of DSGE models, so this approximation it not

exact. To accommodate this, it is better to let s, sometimes called the learning rate be stochastic (independently of θ^{i-1}).[1]

$$c_1 = 2s, \quad s \sim iid\mathcal{U}[0, \bar{s}],$$

where $\bar{s}$ is a tuning parameter. This means that average step-size is $\bar{s}/2$. For our simulations below, we will set the hyperparameters of the algorithm

$$\text{Newton MH} : \bar{s} = 2, c_2 = 1.$$

The Newton MH and MAL algorithms depart from the RWMH by using local information about the posterior embedded in the slope and curvature of the posterior to potentially build more efficient simulators. The RWMH blindly searches through the parameter space looking areas of high posterior density, whereas both the Newton and MALA algorithms explicitly account for this objective in their proposal distributions. To inclusion of an accept-reject Metropolis step ensures that posterior will still be obtained as the invariant distribution of the chain.

For both MAL and Newton MH algorithm, there is a cost born by including the derivatives of the log posterior. For DSGE models, these cannot be obtained analytically. Herbst (2011) uses matrix calculus to derive efficient algorithms for computing these objects for some DSGE models. Still, using numerical differentiation, while slow, can still produce reasonably sized chains without taking too much time.

4.4.3 Block MH Algorithm

Despite a careful choice of the proposal distribution $q(\cdot|\theta^{i-1})$, it is natural that the efficiency of the MH algorithm decreases as dimension of the parameter vector θ increases. This problem is particularly pronounced for the RWMH, as we will see below. The success of the proposed random walk move decreases as the dimension d of the parameter space increases. One way to alleviate this problem, is to break the parameter vector into blocks. Consider a d dimensional parameter vector, θ. A $p$artition of the parameter space, B, is collection of B sets of indices that this mutually exclusive and collectively exhaustive labels over subsets of the parameters vector. Call the subsets θ_b, $b = 1, \ldots, B$. In the context of a sequence

[1] As long as s and θ^{i-1} are independent, the Markov transition implied will still preserve the posterior as its invariant distribution. This can seen by thinking of an augmented posterior $p(s, \theta|Y)$ and casting the algorithm as the so-called Metropolis-within-Gibbs.

of parameter draws, let θ_b^i refer to the bth block of ith draw of θ and let $\theta_{<b}^i$ refer to the ith draw of all of the blocks before b and similarly for $\theta_{>b}^i$. Algorithm 6 describes a generic Block MH algorithm.

Algorithm 6 (Block MH Algorithm) *1. Draw $\theta^0 \in \Theta$.*

2. Iterate. For $i = 1, \ldots, n$: Partition the parameter vector into N_{blocks} blocks B_i of the form $\theta = [\theta_1, \ldots, \theta_{N_{blocks}}]'$ via some rule (perhaps probabilistic), unrelated to the current state in the chain. For $b = 1, \ldots, N_{blocks}$: Draw $\vartheta_b \sim q(\cdot|\left[\theta_{<b}^i, \theta_b^{i-1}, \theta_{\geq b}^{i-1}\right])$. With probability,

$$\alpha = \max\left\{\frac{p(\left[\theta_{<b}^i, \vartheta_b, \theta_{>b}^{i-1}\right]|Y)q(\theta_b^{i-1}, |\theta_{<b}^i, \vartheta_b, \theta_{>b}^{i-1})}{p(\theta_{<b}^i, \theta_b^{i-1}, \theta_{>b}^{i-1}|Y)q(\vartheta_b|\theta_{<b}^i, \theta_b^{i-1}, \theta_{>b}^{i-1})}, 1\right\},$$

set $\theta_b^i = \vartheta_b$, otherwise set $\theta_b^i = \theta_b^{i-1}$.

In order to make the Block MH algorithm operational the researcher has to decided how to allocate parameters to blocks in each iteration and how to choose the proposal distribution $q(\cdot|\left[\theta_{<b}^i, \theta_b^{i-1}, \theta_{>b}^{i-1}\right])$ for parameters of block b.

In general, the optimal block structure is not known outside of a few special cases – discussed in, for example, Roberts and Sahu (1997). A good rule of thumb, however, is that we want the parameters *within* a block, say, θ^b, to be as correlated as possible while we want the parameters between blocks, say, θ^b and θ^{-b}, should be "as independent as possible," according to Robert and Casella (2004). The intuition for this rule is the following: if A and B are independent, then sampling $p(A|B)$ and $p(B|A)$ iteratively will produce draws from $p(A, B)$, since $p(A|B) = p(A)$ and $p(B|A) = p(B)$. On the other hand, if A and B are perfectly correlated, then sampling $p(A|B)$ amounts to solving a deterministic function for a in b. The subsequent draw from $P(B|A)$ will amount to solving for b in a via the inverse of the original; that is, b will be the same value as before and the chain will not move throughout the parameter space. Unfortunately, picking the "optimal" blocks in this fashion requires a priori knowledge about the posterior and it therefore often infeasible.

The first three papers in the DSGE model literature to consider blocking were Curdia and Reis (2009), Chib and Ramamurthy (2010), and Herbst (2011). Curdia and Reis (2009) group the parameters by type: economic – those related to agents' preferences and production technologies – and statistical – those governing the exogenous processes driving the

model. The rationale for this grouping is that it is relatively straightforward to design proposal distributions for the statistical parameters. However, the grouping is unlikely to be optimal, because, for instance, economic parameters related to the persistence generated by the internal propagation mechanism of a DSGE model may be highly correlated with the parameters of the exogenous processes. Chib and Ramamurthy (2010) propose grouping parameters randomly. Essentially, the user specifies how many blocks to partition the parameter vector into and every iteration a new set of blocks is constructed. While there will be correlated blocks sometimes, the randomization ensures that this does not occur as persistent feature of the chain. Key to the algorithm is that the block configuration is independent of the Markov-Chain. This is crucial for ensuring the validity of the SLLN. Otherwise, the chain is said to be adaptive and the asymptotic theory is substantially more complicated. Herbst (2011) constructs a Block MH algorithm in which the blocking is explicitly based on the posterior correlation structure which is approximated based on draws from a burn-in period. He provides evidence that the distributional blocking procedure outperforms the random blocking.

In the remainder of this book we will use Random-Block MH algorithms of the following form:

Algorithm 7 (Random-Block MH Algorithm) *0. Generate a sequence of random partitions* $\{B_i\}_{i=1}^{n}$ *of the parameter vector* θ *into* N_{blocks} *equally sized blocks, denoted by* θ_b, $b = 1, \ldots, N_{blocks}$*: assign an iid*$U[0,1]$ *draw to each element of* θ*, sort the parameters according to the assigned random number, and then let the b'th block consists of parameters* $(b-1)N_{blocks}, \ldots, bN_{blocks}$.[2]

1. Execute Algorithm 6.

In order to tailor the block-specific proposal distributions, Chib and Ramamurthy (2010) advocates using an optimization routine – specifically, simulated annealing – to find the mode of the conditional posterior distribution. As in the RWHM-V algorithm, the variance of the proposal distribution is based on the inverse Hessian of the conditional log posterior density evaluated at the mode. This algorithm is called Tailorized Random Block MH (TaRBMH) algorithm. While the TaRBMH algorithm is very successful in reducing the persistence of the Markov chain relative to the benchmark RWMH-V algorithm, the downside is that the

[2]If the number of parameters is not divisible by N_{blocks}, then the size of a subset of the blocks has to be adjusted.

algorithm is very slow due to the likelihood evaluations required to execute the simulated annealing step and the computation of the Hessian.

4.5 A Look at the Accuracy of MH Samplers

We proceed by assessing the accuracy of the various MH sampler that we introduced in the preceding sections. While our earlier discussion was couched in terms of a general h, here we simply use Monte Carlo estimates of the posterior means. We adjust our notation slightly so that $\bar{\theta}$ is Monte Carlo approximation of $\mathbb{E}_\pi[\theta]$, $\mathbb{V}_\pi[\theta]$ is posterior variance of θ, and $\bar{V}[\bar{\theta}]$ is asymptotic variance of $\bar{\theta}$. Using the notation of Section 3.4 we consider two measures of accuracy: the Newey-West HAC estimator $HAC[\bar{\theta}]$ that can be computed for each run of the MH algorithm and the small sample variance $\hat{V}(\bar{\theta})$ of the Monte Carlo approximation $\bar{\theta}$ that is computed based on the output of 20 chains. In Section 4.5.1 we illustrate the effect of the scaling constant c in the RWMH-V algorithm on the accuracy of the Monte Carlo approximation. In Section 4.5.2 we examine the accuracy of the MAL algorithm and the Newton MH algorithm and we consider the effect of blocking on the accuracy of the RWMH-V and the RWMH-I algorithms.

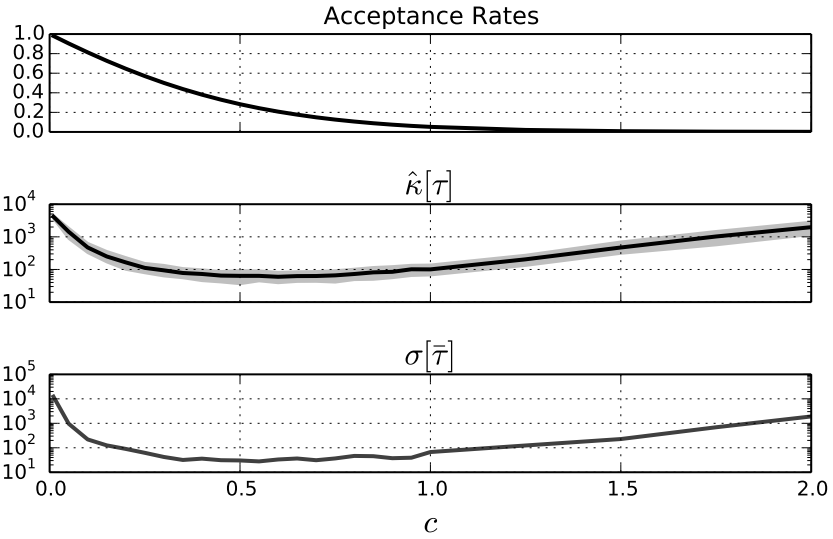
4.5.1 The Effect of Scaling the Proposal

Given the widespread use of the RWMH-V algorithm, it is instructive to investigate the effect of the scaling constant c. To do so, we run the single-block RWMH-V algorithm for different choices c. Here we are taking for granted that each of the simulators have converged to the posterior; detailed examination of the posterior (not shown) confirms this. For each choice of c, we run 20 Monte Carlo chains, with draws initialized around the posterior mode, in line with standard practice. Moreover, we use a burn-in period equal to half of the chain length.

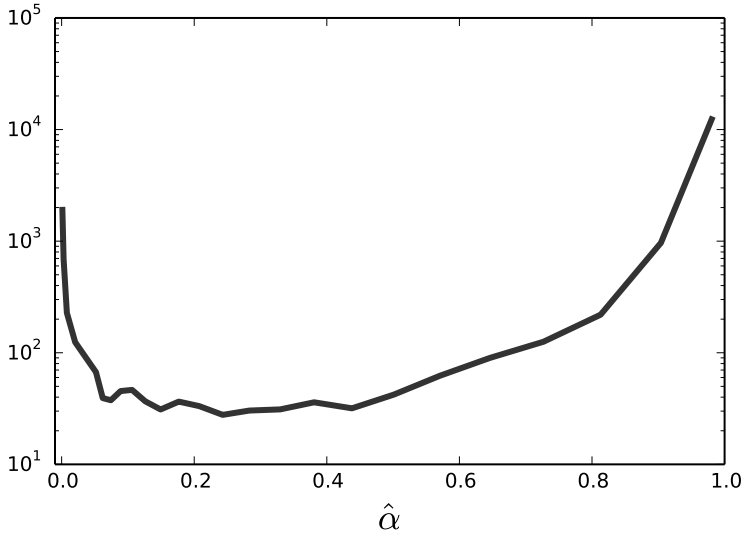
The results are depicted in Figure 4.5. The acceptance probability of the RWMH-V sampler is decreasing in c. If c is small, the proposed random-walk steps of the sampler are tiny and the probability that the proposed draws are accepted is very high. As c increases, the average proposed step sizes get larger and the probability of acceptance decreases because it becomes more likely to propose a parameter value that is associated with a low posterior density. We measure the variance of the Monte Carlo approximation using $HAC(h)$ and

Figure 4.5: Scaling of Proposal Density versus Accuracy of Posterior Mean of τ

(i) Effect of Scaling



(ii) Acceptance Rate versus Accuracy



Notes: Panel (i): we plot as a function of the scaling constant c the acceptance rate, the estimates $\hat{\kappa} = 1 + 2\sum_{l=1}^{L}\big(1 - l/(L+1)\big)\hat{\rho}_l$ for multiple chains, and $N\hat{V}(\bar{\tau})/\mathbb{V}_\pi[\tau]$ where $\hat{V}(\bar{\tau})$ is the small sample variance of $\bar{\tau}$ across multiple runs. Panel (ii): we plot $N\hat{V}(\bar{\tau})/\mathbb{V}_\pi[\tau]$ versus the acceptance rate $\hat{\alpha}$.

$\hat{V}(h)$. Both measures are very similar and indicate that the accuracy of the posterior mean approximation has an inverted-U shape as a function of c. While 4.5 focuses on τ, the results

for the other parameters are qualitatively similar. The minimum, i.e., the highest precision, is attained for $c = 0.5$. Intuitively, for small values of c the serial correlation and hence $\hat{\kappa}$ is large because the step-sizes are very small. For very large values of c the serial correlation is high because the probability that a proposed parameter draw is rejected and therefore $\theta^i = \theta^{i-1}$ is very high. Panel (ii) of Figure 4.5 depicts the relationship between the acceptance rate and the accuracy of the Monte Carlo approximation. For the posterior distribution of τ the Monte Carlo approximation error is smallest for acceptance rates between 20% and 40%.

4.5.2 A Comparison of Algorithms

We now proceed to the comparison of several different MH algorithms. The comparison includes the naive 1-Block RWMH-I and 3-Block RWMH-I algorithms, the benchmark 1-Block RWHM-V algorithm as well as a 3-Block RWMH-V algorithm, a 3-Block MAL algorithm, and a 4-Block Newton MH algorithm. For the Block RWMH-V algorithm we use a random allocation of parameters to blocks. To configure the proposal distributions, we use the posterior covariance matrix for the parameters included in each block and adjust the scaling constant to target an acceptance rate of 30-40%. As before, for each algorithm, we run 20 Monte Carlo chains, with draws initialized around the posterior mode. Given the different computational time for each algorithm, we run chains of different lengths for each algorithm. For the variance calculations, the we use a burn-in period equal to half of the chain length. Finally, for each algorithm, we pick the tuning coefficient c to achieve an acceptance rate between 30% and 50%.

The average running time [minutes:seconds] and the acceptance rate for each algorithm are reported in Table 4.2. The single block RWMH algorithms are the fastest because they only require one likelihood evaluation per draw. The 3-Block RWMH-I algorithm requires three likelihood evaluation and some additional time to assign parameters to blocks. The 3-Block RWMH-V algorithm is noticeably slower because of the likelihood evaluations required to tailor the proposal distributions for the conditional posteriors. MAL and Newton MH are computationally the most demanding algorithms.[3]

Before examining the accuracy of the Monte Carlo approximations, we will take a look at the persistence of the Markov chains generated by the six algorithms. The top panel of

[3]Rather than using the same number of draws for each algorithm, we reduce the number of draws for the algorithms with a longer running time. This allows us to control the overall computational time allocated to experiments in which we run the algorithm many times.

Table 4.2: Tuning of MH Algorithm

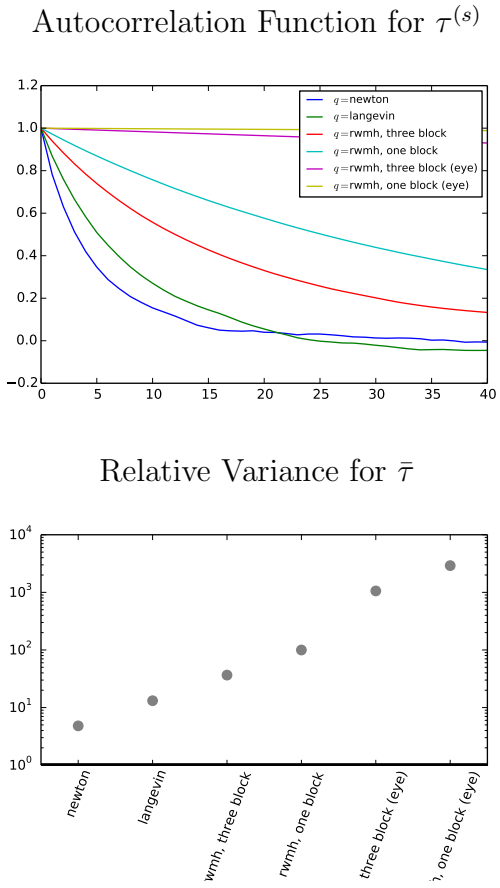
Algorithm	N	Avg. Running Time [minutes:seconds]	Avg. Acceptance Rate
1-Block RWMH-I	100,000	00:37	0.30
3-Block RWMH-I	100,000	01:53	0.41
1-Block RWMH-V	100,000	00:38	0.38
3-Block RWMH-V	100,000	01:49	0.35
3-Block MAL	10,000	02:53	0.42
4-Block Newton MH	10,000	17:37	0.53

Figure 4.6 shows estimated autocorrelation functions up to 40 lags for a a single chain. for the sequence $\{\tau^i\}_{i=N_0+1}^{N}$, where τ is the coefficient of relative risk aversion. The choice of proposal distribution for the MH algorithm has a profound effect on the persistence of the chain. The comparison between the 1-Block RWMH-I and the 1-Block RWMH-V algorithms highlights that aligning the contours of the proposal distribution with the contours of the posterior distribution (at the mode) leads to a drastic reduction in the persistence. While the chain generated by the 1-Block RWMH-I algorithm is nearly perfectly correlated even at a displacement of 40, the autocorrelation of the the RWMH-V chain drops below 0.5 after about 25 iterations of the algorithm.

Once the number of blocks is increased from one to three the persistence of the Markov chains generated by the RWMH-I and RWMH-V algorithms drops. Thus, blocking has indeed the desired effect. The 3-Block RWMH-I algorithm, however performs worse than the 1-Block RWMH-V algorithm, highlighting the importance of well-tailored proposal densities. The autocorrelation of the 3-Block RWMH-V algorithm falls below 0.5 after about 12 iterations. Finally, the MAL and Newton MH algorithms yield very low serial correlation but are also quite costly computationally, as the running times in Table 4.2 indicate.

The second panel of Figure 4.6 shows an estimate of variance ratio $N\hat{V}(\bar{\tau})/\mathbb{V}_\pi[\tau]$ of each sampler, again for the relative risk aversion parameter. As discussed previously, this ratio can be interpreted as a measure of inefficiency. The MH variance is computed across the 20 runs, while the estimate of the posterior variance is obtained from a very large number of posterior draws and can be regarded as exact. The inefficiency measures are in line with the autocorrelation plot in the top panel of the figure. The inefficiency factor for the 1-

Figure 4.6: Performance of Different MH Algorithms

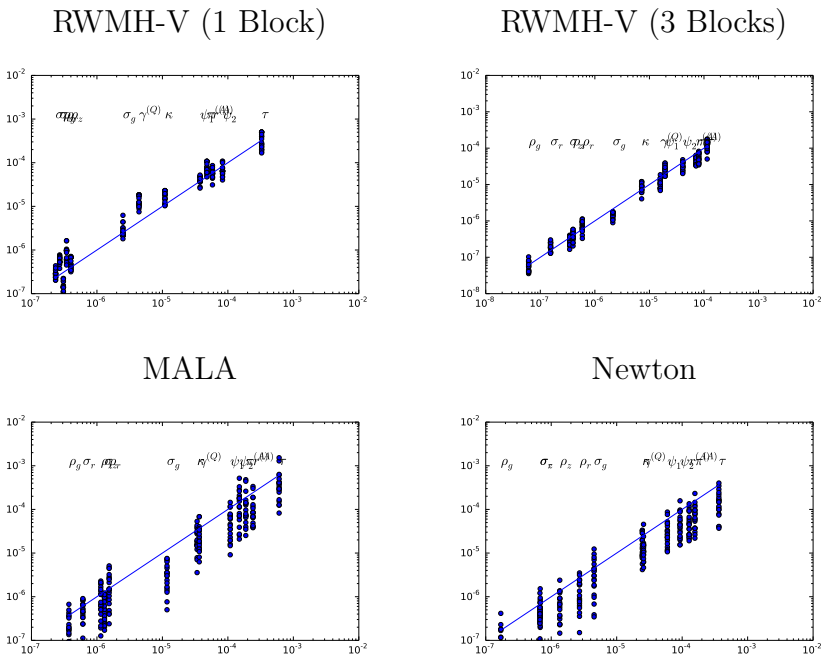


Notes: Panel 1 depicts the autocorrelation function of τ^i. Panel 2 depicts the variance ratio $N\hat{V}(\bar{\tau})/\mathbb{V}_\pi[\tau]$, where $\hat{V}(\bar{\tau})$ is the small sample variance of $\bar{\tau}$ computed across multiple runs. The variance ratio can be interpreted as inefficiency factor.

Block RWMH-I algorithm is about 5,000, meaning that the 100,000 draws that we generated deliver a Monte Carlo approximation that is about as accurate as an approximation obtained from 20 *iid* draws. The 1-Block RWMH-V algorithm has an inefficiency factor of about 100 and blocking reduces it to 50. Thus, the 10,000 draws obtained from the 3-Block RWMH-V algorithm are equivalent to 200 *iid*-equivalent draws.

Adjusting for the running time, the 3-Block RWMH-V algorithm generates about 18.3 *iid*-equivalent draws per second, whereas the 1-Block RWMH-V algorithm produces 26.3 *iid*-equivalent draws per second. Thus, while blocking reduces the persistence in the chain, there is also a computational cost associated with the additional likelihood evaluations. On balance, in this particular application the single-block algorithm comes out ahead in terms

Figure 4.7: Performance of Different MH Algorithms



Notes: Each panel contains scatter plots of the small sample variance $\hat{V}(\bar{\theta})$ computed across multiple chains (x-axis) versus the $HAC(\bar{h})$ estimates of $\bar{V}(\bar{\theta})/N$ (y-axis) computed for each chain.

of generating *iid*-equivalent draws per second. The MAL and Newton MH algorithms have inefficiency ratios of 10 and 5, respectively, which translates into 5.8 and 1.9 *iid*-equivalent draws per second. Thus, in terms of *iid*-equivalent draws per second the benchmark 1-Block RWMH-V algorithm is in fact the most efficient.

Finally, in Figure 4.7 we compare the small sample variance $\hat{V}(\bar{\theta})$ computed as the sample variance of $\bar{\theta}$ across multiple chains to the HAC estimates $HAC(\bar{h})$ computed for each chain. If the chains have converged and the central limit theorem is operational, then the HAC estimates should be very close to the small sample variance of $\bar{h}$. It turns out that this is the case for the small-scale New Keynesian DSGE model: by and large the estimates line up along the 45 degree line.

4.6 Evaluation of the Marginal Data Density

As discussed in Section 3, marginal data densities play an important role in the evaluation of models. The marginal data density of a model M_j is defined as

$$p(Y|M_j) = \int p(Y|\theta, M_j)p(\theta|M_j)d\theta \tag{4.8}$$

and it is used to turn prior model probabilities into posterior model probabilities, see (3.12). In general, the evaluation of the marginal data density involves a high-dimensional integral. Numerical approximations can be obtained by post-processing the output of the MH sampler. In this section we will consider the approximations proposed by Geweke (1999) and Chib and Jeliazkov (2001). The numerical accuracy of alternative algorithms is discussed in Ardia, Bastürk, Hoogerheide, and van Dijk (2012).

4.6.1 Geweke's Harmonic Mean Estimator

Starting point for the harmonic mean (or reciprocal importance sampling) estimator of $p(Y)$ (we omit the model index M_j from the conditioning set) is the slightly rewritten version of Bayes Theorem

$$\frac{1}{p(Y)} = \frac{1}{p(Y|\theta)p(\theta)}p(\theta|Y). \tag{4.9}$$

Note that we can multiply both sides of this equation by a function $f(\theta)$ with the property that $\int f(\theta)d\theta$. Thus,

$$\frac{1}{p(Y)} = \int \frac{f(\theta)}{p(Y|\theta)p(\theta)}p(\theta|Y)d\theta. \tag{4.10}$$

Recall that the MH samplers deliver a sequence of draws $\{\theta^i\}_{i=1}^N$ from the posterior distribution $p(\theta|Y)$. This suggests that a Monte Carlo approximation of the inverse marginal data density can be obtained as

$$\frac{1}{p(Y)} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(\theta^i)}{p(Y|\theta^i)p(\theta^i)}. \tag{4.11}$$

The convergence of the Monte Carlo average depends on the existence of the moments of the ratio of $f(\theta^i)/[p(Y|\theta^i)p(\theta^i)]$. Note that draws of θ^i associated with a low likelihood value can generate large outliers and invalidate the convergence of the Monte Carlo average. For this reason, Geweke (1999) suggests to choose a function $f(\theta)$ that approximates the shape of the posterior distribution but is equal to zero for parameter draws in the tails of the posterior distribution. A function that satisfies this property, at least if the posterior

distribution is approximately elliptical, is the density of a truncated multivariate normal distribution. Let d be the dimension of the parameter space θ. Moreover, let $\bar{\theta}$ and V_θ be numerical approximations of the posterior mean and covariance matrix of θ computed from the output of the posterior sampler. Now define $f(\theta)$ as

$$\begin{aligned} f(\theta) &= \tau^{-1}(2\pi)^{-d/2}|V_\theta|^{-1/2}\exp\left[-0.5(\theta-\bar{\theta})'V_\theta^{-1}(\theta-\bar{\theta})\right] \\ &\quad \times\mathcal{I}\left\{(\theta-\bar{\theta})'V_\theta^{-1}(\theta-\bar{\theta}) \leq F^{-1}_{\chi^2_d}(\tau)\right\}, \end{aligned} \tag{4.12}$$

where $\mathcal{I}\{x \leq a\}$ is the indicator function that equals one if $x \leq a$ and is zero otherwise. Overall, this leads to the approximation

$$\hat{p}_G(Y) = \left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(\theta^i)}{p(Y|\theta^i)p(\theta^i)}\right]^{-1}. \tag{4.13}$$

The selection of τ, the probability associated with truncation, is left to the user. A low value for τ eliminates θ^i which lie in tails of the posterior. This reduces the influence that outliers can have, reducing the variability of the estimator. On the other hand, the more draws that are excluded (i.e., $f(\theta^i) = 0$) in Equation 4.14, the higher the variability the estimator owing to the smaller sample used. In situations where the posterior is approximately normal, it is better to use a higher τ. Usually, when the posterior has already been sampled, the associated posterior kernel $\{p(Y|\theta^i)p(\theta^i)\}_{i=1}^N$ has already been stored, making the evaluation of Equation 4.14. It is recommended, then, to try different values of τ, to assess the stability of the estimator.

4.6.2 Sims, Waggoner, Zha's Estimator

When the posterior is not approximately elliptical, the ratio $f(\theta)/[p(Y|\theta)p(\theta)]$ can vary substantially across the parameter space, leading to poor estimates from Geweke's modified harmonic mean estimator. Sims, Waggoner, and Zha (2008), hereafter SWZ, proposal an alternative modified harmonic mean estimator through a different approximating function f, an elliptical distribution, whose construction we describe below.

f is centered at the posterior mode, $\hat{\theta}$. When the a distribution is multimodal, the posterior mean may be an area of very low density, leading to mismatch between f and the posterior. Using draws from the posterior, a scaling measure about this mode is constructed as

$$V_{\hat{\theta}} = \frac{1}{N}\sum_{i=1}^{N}(\theta^i-\hat{\theta})(\theta^i-\hat{\theta})'.$$

Defining (and slightly abusing notation), for any draw θ^i, the distance

$$r^i = \sqrt{(\theta^i - \hat{\theta})' V_{\hat{\theta}}^{-1} (\theta^i - \hat{\theta})},$$

To construct the elliptical distribution, f, SWZ first construct a univariate density, g, to match the behavior of the posterior. The density of g is given by,

$$g(r) = \frac{\nu r^{\nu-1}}{b^\nu - a^\nu}, \quad r \in [a, b]$$

and 0 otherwise. Letting c_1, c_{10}, and c_{90} be the first, tenth and ninetieth percentiles of empirical distribution of $\{r^i\}_{i=1}^N$, respectively, the hyperparameters of g are given by,

$$\nu = \frac{\log(1/9)}{\log(c_{10}/c_{90})}, \quad a = c_1, \quad \text{and } b = \frac{c_{90}}{0.9^{1/\nu}}.$$

ν and b are chosen so that the tenth and ninetieth percentiles of g would correspond exactly with those of the $\{r^i\}_{i=1}^N$ if $a = 0$. Given that a is ultimately set to c_1, these percentiles will not match. Using g, SWZ construct the elliptical distribution as

$$\tilde{f}(r) = \frac{\Gamma(d/2)}{2\pi^{d/2} |V_{\hat{\theta}}|^{1/2}} \frac{g(r)}{r^{d-1}}$$

The construction of the final, truncated version of $\tilde{f}$ is also more complicated than Geweke's approach. Potential multimodality means that density may vary substantially even where the posterior is concentrated. Given this, SWZ directly exclude the lowest $1 - q$ proportion of the posterior, which has an associated log posterior kernel cutoff L_{1-q}. Moreover, by construction of the elliptical distribution, posterior draws θ^i for which $r^i \notin [a, b]$ are also truncated. The indicator associated with this truncation region is given by,

$$\mathcal{I}(\theta) = \{\ln p(Y|\theta)p(\theta) > L_{1-q}\} \times \{r \in [a, b]\}.$$

Unlike, Geweke's estimator, the probability that θ lies in this region can only be achieved through simulation, since in relies on information from the posterior and not only on the properties $\tilde{f}$ *per se*. An estimate of the probability, $\hat{\tau}$ is:

$$\hat{\tau} = \frac{1}{J} \sum_{j=1}^{J} I(\theta^j), \quad \theta^j \sim iid \tilde{f}.$$

Putting these together, we have the approximating function

$$f_{SWZ}(\theta) = \hat{\tau}^{-1} \tilde{f} \left(\sqrt{(\theta - \hat{\theta})' V_{\hat{\theta}}^{-1} (\theta - \hat{\theta})} \right) \mathcal{I}(\theta).$$

The modified harmonic mean estimator is then given by:

$$\hat{p}_{SWZ}(Y) = \left[\frac{1}{N}\sum_{i=1}^{N}\frac{f_{SWZ}(\theta^i)}{p(Y|\theta^i)p(\theta^i)}\right]^{-1}. \tag{4.14}$$

There are two draws back associated with this estimator. First, it can be quite noisy, requiring a large J to achieve a stable estimate. Second, computationally it can be quite costly to evaluate the log posterior kernel of a DSGE model repeatedly. Still, if the posterior exhibits multimodality and/or fat tails, the SWZ estimator can yield tremendous advantages.

4.6.3 Chib and Jeliazkov's Estimator

While Geweke's (1999) and SWZ's (2008) harmonic mean estimators could be computed for the output of any posterior simulator, the following method proposed by Chib and Jeliazkov (2001) is closely tied to the MH algorithm (Algorithm 4). We start by rewriting Bayes Theorem as follows:

$$p(Y) = \frac{p(Y|\tilde{\theta})p(\tilde{\theta})}{p(\tilde{\theta}|Y)}. \tag{4.15}$$

Note that this relationship holds for any parameter value $\tilde{\theta}$. We will take $\tilde{\theta}$ to be a parameter value that is associated with a high posterior density, e.g., the posterior mode. In order to make the formula operational, we need to numerically approximate the value of the posterior density $p(\tilde{\theta}|Y)$.

Using the notation introduced in Section 3.4, the proposal density for a transition from θ to $\tilde{\theta}$ is given by $q(\tilde{\theta}|\theta)$. Moreover, the probability of accepting the proposed draw is

$$\alpha(\tilde{\theta}|\theta) = \min\left\{1, \frac{p(\tilde{\theta}|Y)/q(\tilde{\theta}|\theta)}{p(\theta|Y)/q(\theta|\tilde{\theta})}\right\}.$$

Using the definition of $\alpha(\tilde{\theta}|\theta)$ it follows that

$$\begin{aligned}
&\int \alpha(\tilde{\theta}|\theta)q(\tilde{\theta}|\theta)p(\theta|Y)d\theta \qquad (4.16)\\
&= \int \min\left\{1, \frac{p(\tilde{\theta}|Y)q(\tilde{\theta}|\theta)}{p(\theta|Y)/q(\theta|\tilde{\theta})}\right\} q(\tilde{\theta}|\theta)p(\theta|Y)d\theta\\
&= p(\tilde{\theta}|Y)\int \min\left\{\frac{p(\theta|Y)q(\theta|\tilde{\theta})}{p(\tilde{\theta}|Y)/q(\tilde{\theta}|\theta)}, 1\right\} d\theta\\
&= p(\tilde{\theta}|Y)\int \alpha(\theta|\tilde{\theta})q(\theta|\tilde{\theta})d\theta.
\end{aligned}$$

In turn, the posterior density at $\tilde{\theta}$ can be approximated as

$$\hat{p}(\tilde{\theta}|Y) = \frac{\frac{1}{N}\sum_{i=1}^{N}\alpha(\tilde{\theta}|\theta^i)q(\tilde{\theta}|\theta^i)}{\frac{1}{J}\sum_{j=1}^{J}\alpha(\theta^j|\tilde{\theta})}. \tag{4.17}$$

Here $\{\theta^i\}_{i=1}^N$ is the sequence of draws generated with the MH algorithm and $\{\theta^j\}_{j=1}^J$ are draws from $q(\theta|\tilde{\theta})$ that can be generated by direct sampling. The final approximation of the marginal data density is given by

$$\hat{p}_{CS}(Y) = \frac{p(Y|\tilde{\theta})p(\tilde{\theta})}{\hat{p}(\tilde{\theta}|Y)}. \tag{4.18}$$

Like the SWZ estimator, this estimator requires the user evaluate the log posterior kernel J additional, which can be expensive for large models.

4.6.4 Illustration

We estimate log marginal data density of the simple DSGE model using 5000 draws from the posterior for 20 separate runs of the RWMH-V algorithm. Table 4.3 displays the mean and standard deviation across the 20 runs for each estiamtor. For the Geweke and SWZ estimators we use two different truncation probabilities.

All the algorithms give roughly the same answer, although the Chib and Jeliazkov estimator is much more variable. Moving to the modified harmonic mean estimators, the SWZ estimate is more-or-less unchanged. The mean of the Geweke estimator changes, consistent with the observation that the has DSGE posterior has slightly fatter tails than a multivariate normal, which the SWZ is less affected by.

Table 4.3: Marginal Data Density

Model	Mean($\ln \hat{p}(Y)$)	Std. Dev.($\ln \hat{p}(Y)$)
Geweke ($\tau = 0.9$)	-346.06	0.06
Geweke ($\tau = 0.5$)	-346.15	0.05
SWZ ($q = 0.9$)	-346.27	0.02
SWZ ($q = 0.5$)	-346.29	0.02
Chib and Jeliazkov	-345.38	0.68

Notes: Table shows mean and standard deviation of different estimators of the log marginal data density, computed over twenty runs of the RWMH sampler using 5000 draws. The SWZ estimator uses $J = 10000$ draws to compute $\hat{\tau}$, while the CJ estimators uses $J = 10000$ Table 6.1.3.

Chapter 5

Sequential Monte Carlo Methods

In Section 3.3 we introduced the idea of importance sampling. The key difficulty, in particular in high dimensional parameter spaces, is to find a good proposal densities. In this chapter, we will explore methods in which proposal densities are constructed sequentially. Suppose ϕ_n, $n = 1, \ldots, N_\phi$, is a sequence that slowly increases from zero to one. We can define a sequence of tempered posteriors as

$$\pi_n(\theta) = \frac{[p(Y|\theta)]^{\phi_n} p(\theta)}{\int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta} \quad n = 1, \ldots, N_\phi, \quad \phi_n \uparrow 1. \tag{5.1}$$

Provided that ϕ_1 is close to zero, the prior density $p(\theta)$ may serve as an efficient proposal density for $\pi_1(\theta)$. Likewise, the density $\pi_n(\theta)$ may be a good proposal density for $\pi_{n+1}(\theta)$. Sequential Monte Carlo (SMC) algorithms try to exploit this insight efficiently.

SMC algorithms were initially developed to solve filtering problems that arise in nonlinear state-space models. We will consider such filtering applications in detail in Chapter 7. Chopin (2002) showed how to adopted the particle filtering techniques to conduct posterior inference for a static parameter vector. Textbook treatments can be found in Cappé, Moulines, and Ryden (2005), Liu (2001) and a recent survey focusing on econometric applications is provided by Creal (2012). The first paper that applied SMC techniques to posterior inference in DSGE models is Creal (2007). He presents a basic SMC algorithm and uses it for posterior inference in a small-scale DSGE model that is similar to the model in Section 1.1. Herbst and Schorfheide (2014) developed the algorithm further, provided some convergence results for an adaptive version of the algorithm building on the theoretical analysis of Chopin (2004), and showed that a properly tailored SMC algorithm delivers more reliable posterior inference for large-scale DSGE models with multi-modal posterior than the

widely-used RMWH-V algorithm. Much of the subsequent exposition borrows from Herbst and Schorfheide (2014). An additional advantage of the SMC algorithms over MCMC algorithms, on the computational front, highlighted by Durham and Geweke (2012), is that SMC is much more amenable to parallelization. Durham and Geweke (2012) show how to implement an SMC algorithm on graphical processing unit (GPU), facilitating massive speed gains in estimations. While the evaluation of DSGE likelihoods is not (yet) amenable to GPU calculation, we will show how to exploit the parallel structure of the algorithm.

Because we will generate draws of θ sequentially, from a sequence of posterior distributions $\{\pi_n(\theta)\}_{n=1}^{N_\phi}$, it is useful to equip the parameter vector with a subscript n. Thus, θ_n is associated with the density $\pi_n(\cdot)$.

5.1 An SMC Algorithm for DSGE Models

The sequence of posteriors in (5.1) was obtained by tempering the likelihood function, that is, we replaced $p(Y|\theta)$ by $[p(Y|\theta)]^{\phi_n}$. Alternatively, one could construct the sequence of posteriors by sequentially adding observations to the likelihood function, that is, $\pi_n(\theta)$ is based on $p(Y_{1:\lfloor \phi_n T \rfloor}|\theta)$:

$$\pi_n^{(D)}(\theta) = \frac{p(Y_{1:\lfloor \phi_n T \rfloor})p(\theta)}{\int p(Y_{1:\lfloor \phi_n T \rfloor})p(\theta)d\theta} \tag{5.2}$$

This data tempering is particularly attractive in sequential applications. Due to the fact that individual observations are not divisible, the data tempering approach is slightly less flexible. This may matter for the early stages of the SMC sampler in which it may be advantageous to add information in very small increments. The subsequent algorithm is presented in terms of likelihood tempering. However, we will also discuss the necessary adjustments for data tempering.

We begin with the description of the basic algorithm in Section 5.1.1. This algorithm consists of three steps, using Chopin (2004)'s terminology: *correction*, that is, reweighting the particles to reflect the density in iteration n; *selection*, that is, eliminating any particle degeneracy by resampling the particles; and *mutation*, that is, propagating the particles forward using a Markov transition kernel to adapt to the current bridge density. Section 5.1.2 provides details on the choice of the transition kernel in the mutation step, and the adaptive choice of various tuning parameters is discussed in Section 5.1.3.

5.1.1 The Basic Algorithm

Just as the basic importance sampling algorithm, SMC algorithms generate weighted draws from the sequence of posteriors $\{\pi_n\}_{n=1}^{N_\phi}$. The weighted draws are called particles. We denote the overall number of particles by N. At any stage the posterior distribution $\pi_n(\theta)$ is represented by a swarm of particles $\{\theta_n^i, W_n^i\}_{i=1}^N$ in the sense that the Monte Carlo average

$$\bar{h}_{n,N} = \frac{1}{N}\sum_{i=1}^{N} W_n^i h(\theta^i) \xrightarrow{a.s.} \mathbb{E}_\pi[h(\theta_n)]. \tag{5.3}$$

Starting from stage $n-1$ particles $\{\theta_{n-1}^i, W_{n-1}^i\}_{i=1}^N$ the algorithm proceeds in three steps: *correction*, that is, reweighting the particles to reflect the density in iteration n; *selection*, that is, eliminating any particle degeneracy by resampling the particles; and *mutation*, that is, propagating the particles forward using a Markov transition kernel to adapt to the current bridge density.

The algorithm provided below relies sequences of tuning parameters that will ultimately be chosen adaptively in Section 5.1.3. However, to make the exposition more transparent, we begin by assuming that these sequences are provided *ex ante*. Let $\{\rho_n\}_{n=2}^{N_\phi}$ be a sequence of zeros and ones that determine whether the particles are resampled in the selection step and let $\{\zeta_n\}_{n=2}^{\phi_N}$ of tuning parameters for the Markov transition density in the mutation step (see below).

Algorithm 8 (Simulated Tempering SMC)

1. **Initialization.** *($\phi_1 = 0$). Draw the initial particles from the prior: $\theta_1^i \overset{iid}{\sim} p(\theta)$ and $W_1^i = 1$, $i = 1, \ldots, N$.*

2. **Recursion.** *For $n = 2, \ldots, N_\phi$,*

   (a) **Correction.** *Reweight the particles from stage $n-1$ by defining the incremental weights*

   $$\tilde{w}_n^i = [p(Y|\theta_{n-1}^i)]^{\phi_n - \phi_{n-1}} \tag{5.4}$$

   and the normalized weights

   $$\tilde{W}_n^i = \frac{\tilde{w}_n^i W_{n-1}^i}{\frac{1}{N}\sum_{i=1}^N \tilde{w}_n^i W_{n-1}^i}, \quad i = 1, \ldots, N. \tag{5.5}$$

An approximation of $\mathbb{E}_{\pi_n}[h(\theta)]$ *is given by*

$$\tilde{h}_{n,N} = \frac{1}{N}\sum_{i=1}^{N} \tilde{W}_n^i h(\theta_{n-1}^i). \tag{5.6}$$

(b) **Selection.**

Case (i)*: If* $\rho_n = 1$*, resample the particles via multinomial resampling. Let* $\{\hat{\theta}\}_{i=1}^N$ *denote* N *iid draws from a multinomial distribution characterized by support points and weights* $\{\theta_{n-1}^i, \tilde{W}_n^i\}_{i=1}^N$ *and set* $W_n^i = 1$*.*
Case (ii)*: If* $\rho_n = 0$*, let* $\hat{\theta}_n^i = \theta_{n-1}^i$ *and* $W_n^i = \tilde{W}_n^i$*,* $i = 1, \ldots, N$*. An approximation of* $\mathbb{E}_{\pi_n}[h(\theta)]$ *is given by*

$$\hat{h}_{n,N} = \frac{1}{N}\sum_{i=1}^{N} W_n^i h(\hat{\theta}_n^i). \tag{5.7}$$

(c) **Mutation.** *Propagate the particles* $\{\hat{\theta}_i, W_n^i\}$ *via* M *steps of a MH algorithm with transition density* $\theta_n^i \sim K_n(\theta_n|\hat{\theta}_n^i; \zeta_n)$ *and stationary distribution* $\pi_n(\theta)$ *(see Algorithm 9 for details below). An approximation of* $\mathbb{E}_{\pi_n}[h(\theta)]$ *is given by*

$$\bar{h}_{n,N} = \frac{1}{N}\sum_{i=1}^{N} h(\theta_n^i) W_n^i. \tag{5.8}$$

3. *For* $n = N_\phi$ $(\phi_{N_\phi} = 1)$ *the final importance sampling approximation of* $\mathbb{E}_\pi[h(\theta)]$ *is given by:*

$$\bar{h}_{N_\phi,N} = \sum_{i=1}^{N} h(\theta_{N_\phi}^i) W_{N_\phi}^i. \tag{5.9}$$

Algorithm 8 is initialized for $n = 1$ by generating *iid* draws from the prior distribution. This initialization will work well as long as the prior is sufficiently diffuse to assign non-trivial probability mass to the area of the parameter space in which the likelihood function peaks. There do exist papers in the DSGE model estimation literature in which the *posterior mean* of some parameters is several *prior* standard deviations away from the *prior mean*. For such applications it might be necessary to choose $\phi_1 > 0$ and to use an initial distribution that is also informed by the tempered likelihood function $[p(Y|\theta)]^{\phi_1}$. If the particles are initialized based on a more general distribution with density $g(\theta)$, then for $n = 2$ the incremental weights have to be corrected by the ratio $p(\theta)/g(\theta)$.

The correction step reweighs the stage $n-1$ particles to generate an importance sampling approximation of π_n. Because the parameter value θ^i does not change in this step, no

further evaluation of the likelihood function is required. The likelihood value $p(Y|\theta_{n-1}^i)$ was computed as a by-product of the mutation step in iteration $n-1$. As discussed in Section 3.3, the accuracy of the importance sampling approximation depends on the distribution of the particle weights $\tilde{W}_n^i$. The more uniformly the weights are distributed, the more accuracy the approximation. If likelihood tempering is replaced by data tempering, then the incremental weights $\tilde{w}_n^i$ in (5.4) have to be defined as

$$\tilde{w}_n^{i(D)} = p(Y_{(\lfloor \phi_n T \rfloor + 1):\lfloor \phi_n T \rfloor} | \theta). \tag{5.10}$$

The correction steps deliver a numerical approximation of the marginal data density as a by-product. It can be verified that the unnormalized particle weights converge under suitable regularity conditions as follows:

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N} \tilde{w}_n^i W_{n-1}^i &\xrightarrow{a.s.} \int [p(Y|\theta)]^{\phi_n - \phi_{n-1}} \frac{[p(Y|\theta)]^{\phi_{n-1}} p(\theta)}{\int [p(Y|\theta)]^{\phi_{n-1}} p(\theta) d\theta} d\theta \\ &= \frac{\int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta}{\int [p(Y|\theta)]^{\phi_{n-1}} p(\theta) d\theta}. \end{aligned} \tag{5.11}$$

Thus, the data density approximation is given by

$$\hat{p}_{SMC}(Y) = \prod_{n=1}^{N_\phi} \left(\frac{1}{N} \sum_{i=1}^{N} \tilde{w}_n^i W_{n-1}^i \right). \tag{5.12}$$

Computing this approximation does not require any additional likelihood evaluations.

The selection step is designed to equalize the particle weights, if its distribution becomes very uneven. In Algorithm 8 the particles are being resampled whenever the indicator ρ_n is equal to one. In Section 5.1.3 we consider a version of the algorithm in which ρ_n is equal to one, if the effective sample size ESS_n (which is a function of the variance of the particle weights, see Equation (3.30)) falls below a threshold, and otherwise is equal to zero. On the one hand, resampling introduces noise in the Monte Carlo approximation, which makes it undesirable. On the other hand, resampling equalizies the particle weights and therefore increases the accuracy of the correction step in the subsequent iteration. In Algorithm 8 we use multinomial resampling. Alternative resampling algorithms are discussed, for instance, in Liu (2001). While some of these alternative procedures lead to a smaller Monte Carlo variance than multinomial resampling, it is more difficult to establish CLTs.

The mutation step changes the values of the particles from θ_{n-1}^i to θ_n^i. To understand the importance of the mutation step, consider what would happen without this step. For

simplicity, suppose also that $\rho_n = 0$ for all n. In this case the particle values would never change, that is, $\theta_n^i = \theta_1^i$ for all n. Thus, we would be using the prior as importance sampling distribution and reweigh the draws from the prior by the tempered likelihood function $[p(Y|\theta_1^i)]^{\phi_n}$. Given the information contents in a typical DSGE model likelihood function, this procedure would lead to a degenerate distribution of particles, in which in the last stage N_ϕ the weight is concentrated on a very small number of particles and the importance sampling approximation is very inaccurate. Thus, the goal of the mutation step is to adapt the values of the stage n particles to $\pi_n(\theta)$. This is achieved by using steps of an MH algorithm with a transition density that satisfies the invariance property

$$\int K_n(\theta_n|\hat{\theta}_n^i)\pi_n(\hat{\theta}_n^i)d\hat{\theta}_n^i = \pi_n(\theta_n).$$

The execution of the MH steps during the particle mutation phase requires at least one, but possibly multiple, evaluations of the likelihood function for each particle i. To the extent that the likelihood function is recursively evaluated with a filter, data tempering has a computational advantages over likelihood tempering, because the former only requires $\lfloor \phi_n T \rfloor \leq T$ iterations of the filter, whereas the latter requires T iterations. The particle mutation is ideally suited for parallelization, because the MH steps are independent across particles and do not require any communication across processors. For DSGE models, the evaluation of the likelihood function is computationally very costly because it requires to run a model solution procedure as well as a filtering algorithm. Thus, gains from parallelization are potentially quite large.

5.1.2 The Transition Kernel for the Mutation Step

The transition kernel $K_n(\theta_n|\hat{\theta}_n;\zeta_n)$ in the particle mutation phase is generated through a sequence of MH steps. The kernel is indexed by a vector of tuning parameters ζ_n, which may be different at every stage n. In our subsequent applications we will use M steps of a Block RWMH-V algorithm to transform the particle values $\hat{\theta}_n^i$ into θ_n^i. Under a Gaussian proposal density, this algorithm requires a covariance matrix Σ_n^*, which can be partitioned into submatrices for the various parameter blocks, as well as a scaling constant c_n. In principle, this scaling constant could be different for each block, but in our experience with DSGE models the gain from using block-specific scaling is small. Let

$$\zeta_n = \left[c_n,\, vech(\Sigma_n^*)'\right]'. \tag{5.13}$$

In Section 5.1.3 we will replace Σ_n^* by a particle approximation of $\mathbb{V}_{\pi_n}[\theta]$ and update c_n recursively based on past rejection rates of the MH steps. The transition kernel is constructed such that for each ζ_n the posterior $\pi_n(\theta)$ is an invariant distribution. The MH steps are summarized in the following algorithm.

Algorithm 9 (Particle Mutation) *Prior to executing Algorithm 8:*

0. *Generate a sequence of random partitions $\{B_n\}_{n=2}^{N_\phi}$ of the parameter vector θ_n into N_{blocks} equally sized blocks, denoted by $\theta_{n,b}$, $b = 1, \ldots, N_{blocks}$. (See Algorithm 7.) Let $\Sigma_{n,b}^*$ be the partitions of Σ_n^* that correspond to the subvector $\theta_{n,b}$.*

In Step 2(c) in iteration n of Algorithm 8:

1. *For each particle i, run M steps of the Block MH Algorithm 6 using a RWMH-V proposal density of the form*

$$\vartheta_{n,b}^{i,m} \sim N\left(\theta_{n,b}^{i,m-1}, c_n^2 \Sigma_{n,b}^*\right). \tag{5.14}$$

For expository purposes, the sequence of blocks $\{B_n\}$ in Algorithm 8 is generated prior to running the SMC algorithm. This is of no practical consequences – one can also generate B_n as part of Step 2(c) in iteration n of Algorithm 8. The Block RWMH-V could be replaced by some of the alternative MH samplers discussed in Chapter 4. However, in our experience the most important consideration for the performance of the SMC algorithm is parameter blocking and the careful tailoring of the scaling constant c_n and the covariance matrix Σ_n^*. As stated, the matrix $\Sigma_{n,b}^*$ refers to the covariance matrix associated with the marginal distribution of $\theta_{n,b}$. Alternatively, one could also use the covariance matrix associated with the conditional distribution of $\theta_{n,b}|(\theta_{n,<b}, \theta_{n,>b})$.

5.1.3 Tuning and Adaption of the Algorithm

The SMC algorithm involves several tuning parameters. Some of these tuning parameters are chosen *ex ante*, whereas others are determined adaptively, based on the output of the algorithm in earlier stages. This section provides a broad overview of the tuning parameters. Their effect on the performance of the algorithm will be studied in Section 5.3 below.

Number of Particles, Number of Stages, Tempering Schedule. In our implementation of Algorithm 8 the tuning parameters N, N_ϕ, and λ are fixed *ex ante*. The number

of particles N scales the overall accuracy of the Monte Carlo approximation. Because most of the computational burden arises in the mutation step, the computing time increases approximately linearly in N. Under suitable regularity conditions $\bar{h}_{N_\phi,N}$ is $\sqrt{N}$ consistent and satisfies a CLT. N_ϕ determines the number of stages $\pi_n(\cdot)$ used to approximate the posterior distribution $\pi(\cdot)$. Increasing the number of stages, N_ϕ, will decrease the distance between bridge distributions and thus make it easier to maintain particle weights that are close to being uniform. The cost of increasing N_ϕ is that each stage requires additional likelihood evaluations.

The user also has to determine the tempering schedule $\{\phi_n\}_{n=1}^{N_\phi}$. To control its shape we introduce a parameter λ and let

$$\phi_n = \left(\frac{n-1}{N_\phi - 1}\right)^\lambda. \tag{5.15}$$

A large value of λ implies that the bridge distributions will be very similar (and close to the prior) for small values of n and very different as n approaches N_ϕ. In the DSGE model applications we found a value of $\lambda = 2$ to be very useful because for smaller values the information from the likelihood function will dominate the priors too quickly and only a few particles will survive the correction and selection steps. Conversely, if λ is much larger than 2, it makes some of the bridge distributions essentially redundant and leads to unnecessary computations. The choice of λ does not affect the overall number of likelihood evaluations.

Resampling. Resampling becomes necessary when the distribution of particles degenerates. As discussed in Section 3.3, a rule-of-thumb measure of this degeneracy is given by the reciprocal of the uncentered particle variance of the particles, which is computed in the selection step of Algorithm 8 and we called effective sample size:

$$\widehat{ESS}_n = N \Big/ \left(\frac{1}{N}\sum_{i=1}^{N}(\tilde{W}_i^n)^2\right). \tag{5.16}$$

If all particles receive equal weights, then $\widehat{ESS}_n = N$. Using this degeneracy measure, we can now replace the sequence of resampling indicators $\{\rho_n\}_{n=2}^{N_\phi}$ by the adaptive indicators

$$\hat{\rho}_n = \mathcal{I}\{\widehat{ESS}_n < N/2\}, \tag{5.17}$$

where $\mathcal{I}\{x < a\}$ is the indicator function that is equal to one if $x < a$ and equal to zero otherwise.

Mutation Step. The number of MH steps in the mutation phase of the SMC algorithm affects the likelihood with which a particle mutation occurs. The larger M, the higher the

probability that during the M steps at least one of the proposed draws is accepted and the particle value changes. However, each additional MH step also requires additional likelihood evaluations. As we have seen in Chapter 4, increasing the number of blocks N_{blocks} generally reduces the persistence of the MH chain, which increases the probability of a significant change in the particle value.

We choose the sequence of tuning parameters ζ_n defined in (5.13) for the proposal distribution of the Block RWMH-V algorithm adaptively. First, we replace Σ_n^* by the importance sampling approximation of $\mathbb{V}_{\pi_n}[\theta]$. Second, we adjust the scaling factor c_n to ensure that the acceptance rate in the MH step is approximately 25%, which according to Panel (ii) of Figure 4.5 delivers a high degree of accuracy. At each iteration n we then replace ζ_n in (5.13) with[1]

$$\hat{\zeta}_n = \big[\hat{c}_n, vech(\tilde{\Sigma}_n)'\big]'. \tag{5.18}$$

The following algorithm describes how $\hat{\zeta}_n$ is constructed at each iteration n.

Algorithm 10 (Adaptive Particle Mutation) *For $n \geq 2$, prior to Step 1 of Algorithm 9:*

1. *Compute an importance sampling approximation $\tilde{\Sigma}_n$ of $\mathbb{V}_{\pi_n}[\theta]$ based on the particles $\{\theta_{n-1}^i, \tilde{W}_n^i\}_{i=1}^N$.*

2. *Compute the average empirical rejection rate $\hat{R}_{n-1}(\hat{\zeta}_{n-1})$, based on the Mutation step in iteration $n-1$. The average are computed across the N_{blocks} blocks.*

3. *Adjust the scaling factor according to $\hat{c}_2 = c^*$ and $\hat{c}_n = \hat{c}_{n-1} f\big(1 - \hat{R}_{n-1}(\hat{\zeta}_{n-1})\big)$ for $n \geq 3$, where $f(x) = 0.95 + 0.10\frac{e^{16(x-0.25)}}{1+e^{16(x-0.25)}}$.*

4. *Execute Algorithm 9 by replacing ζ_n with $\hat{\zeta}_n = \big[\hat{c}_n, vech(\tilde{\Sigma}_n)'\big]'$.*

Note that $f(0.25) = 1$, which means that the scaling factor stays constant whenever the target acceptance rate is achieved. If the acceptance rate is below (above) 25% the scaling factor is decreased (increased). The range of the adjustment is determined by the factor 0.1 and the sensitivity to the deviation of the actual from the targeted acceptance rate is determined by the factor 16 in the expression for $f(x)$. We found that the particular constants in the definition of $f(x)$ worked well in practice. To satisfy the regularity conditions for the theoretical analysis we chose a function $f(x)$ that is differentiable.

[1] We use "tilde" instead of "hat" for θ and Σ because the approximations are based on the correction step in Algorithm 8.

5.1.4 Convergence

A detailed theoretical analysis of the convergence properties of SMC algorithms is provided by Chopin (2004). Herbst and Schorfheide (2014) adapt the proofs of the SLLN and CLT in Chopin (2004) to cover the specifics of Algorithm 8. The convergence results can be elegantly proved recursively, that is, by showing that the convergence of $\bar{h}_{n-1,N}$ implies the convergence of $\bar{h}_{n,N}$. We briefly outline the derivation of the limit distribution of the Monte Carlo approximations for the non-adaptive version of the SMC algorithm (Algorithm 8), meaning we assume that the sequences $\{\rho_n\}$, $\{B_n\}$, and $\{\zeta_n\}$ are predetermined. A rigorous proof, using the same notation as below, can be found in the supplemental appendix to Herbst and Schorfheide (2014).

The convergence results rely on three types of assumptions. First, we assume that the prior is proper, the likelihood function is uniformly bounded that the tempered marginal likelihood of the data in stage $n = 2$ is non-zero:

$$\int p(\theta)d\theta < \infty, \ \sup_{\theta\in\Theta} p(Y|\theta) < M < \infty, \ \int [p(Y|\theta)]^{\phi_2} p(\theta)d\theta > 0. \tag{5.19}$$

Second, we require the existence of moments by considering functions $h(\theta)$ that belong to the classes $\mathcal{H}_1$ and $\mathcal{H}_2$ as

$$\begin{aligned} \mathcal{H}_1 &= \left\{ h(\theta) \mid \exists \delta > 0 \text{ s.t. } \int |h(\theta)|^{1+\delta} p(\theta)d\theta < \infty \right\} \\ \mathcal{H}_2 &= \left\{ h(\theta) \mid \exists \delta > 0 \text{ s.t. } \int |h(\theta)|^{2+\delta} p(\theta)d\theta < \infty \right\}. \end{aligned} \tag{5.20}$$

Because the likelihood is assumed to be bounded, we can immediately deduce the existence of $j + \delta$ moments of the tempered posterior distributions $\pi_n(\theta)$, $n = 2, \ldots, N_\phi$, for functions $\mathcal{H}_j$. Note that $\mathcal{H}_2 \subseteq \mathcal{H}_1$. We obtain a SLLN for functions contained in $\mathcal{H}_1$ and a CLT for functions in $\mathcal{H}_2$. Third, we will assume that

$$\bar{h}_{n-1,N} \xrightarrow{a.s.} \mathbb{E}_{\pi_{n-1}}[h], \ \sqrt{N}\big(\bar{h}_{n-1,N} - \mathbb{E}_{\pi_{n-1}}[h]\big) \Longrightarrow N(0, \Omega_{n-1}(h)) \tag{5.21}$$

as the number of particles $N \longrightarrow \infty$. Recall that in stage $n = 1$ we directly sample from the prior distribution. Thus, the moment bounds in (5.20) suffices to ensure the convergence of $\bar{h}_{1,N}$.

Correction Step. For the subsequent calculations, it is convenient to normalize the incremental weight as follows:

$$v_n(\theta) = \frac{Z_{n-1}}{Z_n} [p(Y|\theta)]^{\phi_n - \phi_{n-1}}. \tag{5.22}$$

In terms of the normalized incremental weights, we can write the Monte Carlo approximation in the correction step as

$$\tilde{h}_{n,N} = \frac{1}{N}\sum_{i=1}^{N} \tilde{W}_n^i h(\theta_{n-1}^i) = \frac{\frac{1}{N}\sum_{i=1}^{N} h(\theta_{n-1}^i) v_n(\theta_{n-1}^i) W_{n-1}^i}{\frac{1}{N}\sum_{i=1}^{N} v_n(\theta_{n-1}^i) W_{n-1}^i}. \tag{5.23}$$

The normalized incremental weights have the following useful property:

$$\begin{aligned} \int h(\theta) v(\theta) \pi_{n-1}(\theta) d\theta &= \int \frac{Z_{n-1}}{Z_n} [p(Y|\theta)]^{\phi_n - \phi_{n-1}} \frac{[p(Y|\theta)]^{\phi_{n-1}} p(\theta)}{Z_{n-1}} d\theta \\ &= \int h(\theta) \pi_n(\theta) d\theta. \end{aligned} \tag{5.24}$$

As the number of particles $N \longrightarrow \infty$, the Monte Carlo approximation $\tilde{h}_{n,N}$ converges as follows:

$$\begin{aligned} \tilde{h}_{n,N} &\xrightarrow{a.s.} \int h(\theta_n) \pi_n(\theta_n) d\theta_n = \mathbb{E}_{\pi_n}[h] \\ \sqrt{N}\big(\tilde{h}_{n,N} - \mathbb{E}_{\pi_n}[h]\big) &\implies N\big(0, \tilde{\Omega}_n(h)\big), \quad \tilde{\Omega}_n(h) = \Omega_{n-1}\big(v_n(\theta)(h(\theta) - \mathbb{E}_{\pi_n}[h])\big). \end{aligned} \tag{5.25}$$

The almost-sure convergence follows from the SLLN in (5.21) and the property of the normalized incremental weights in (5.24). The asymptotic covariance matrix associated with $\tilde{h}_{n,N}$ has the same form as the asymptotic covariance matrix $\Omega(h)$ in (3.28) associated with the importance sampler.

Selection Step. The accuracy of the Monte Carlo approximation in (5.25) depends on the distribution of the incremental weights $v_n(\theta_{n-1}^i)$. Therefore, the adaptive version of the algorithm described in Section 5.1.3 monitors the variance of the particle weights, transformed into $\widehat{ESS}_n$. If the distribution of particle weights is very uneven, then the particles are resampled. In the non-adaptive version of the algorithm resampling occurs whenever $\rho_n = 1$.

To examine the effect of resampling on the accuracy of the Monte Carlo approximation, recall that we denoted the resampled particles by $\hat{\theta}_n^i$. Let $\mathcal{F}_{n-1,N}$ be the σ-algebra generated by $\{\theta_{n-1}^i, \tilde{W}_n^i\}$, where $\tilde{W}_n^i$ are the normalized particle weights computed in the correction step (see (5.23)). Under multinomial resampling, the expected value of functions of resampled particles is given by

$$\mathbb{E}[h(\hat{\theta})|\mathcal{F}_{n-1,N}] = \frac{1}{N}\sum_{i=1}^{N} h(\theta_{n-1}^i)\tilde{W}_n^i = \tilde{h}_{n,N}. \tag{5.26}$$

Using this equality, we can decompose

$$\begin{aligned}\hat{h}_{n,N} - \mathbb{E}_{\pi_n}[h] &= \big(\tilde{h}_{n,N} - \mathbb{E}_{\pi_n}[h]\big) + \frac{1}{N}\sum_{i=1}^{N}\big(h(\hat{\theta}_n^i) - \mathbb{E}[h(\hat{\theta})|\mathcal{F}_{n-1,N}]\big) \\ &= I + II,\end{aligned} \tag{5.27}$$

say. The large sample behavior of I follows directly from (5.25). Conditional on $\mathcal{F}_{n-1,N}$ the $h(\hat{\theta}_n^i)$ form a triangular array of (discrete) random variables that are *iid* within each row with mean $\mathbb{E}[h(\hat{\theta})|\mathcal{F}_{n-1,N}]$. Thus, the behavior of term II can be analyzed with a SLLN and a CLT for triangular arrays of *iid* random variables. It can be shown that

$$\begin{aligned}\hat{h}_{n,N} &\xrightarrow{a.s.} \int h(\theta_n)\pi_n(\theta_n)d\theta_n \\ \sqrt{N}\big(\hat{h}_{n,N} - \mathbb{E}_{\pi_n}[h]\big) &\Longrightarrow N\big(0, \hat{\Omega}_n(h)\big), \quad \hat{\Omega}_n(h) = \tilde{\Omega}_n(h) + \mathbb{V}_{\pi_n}[h].\end{aligned} \tag{5.28}$$

The second term in asymptotic variance $\hat{\Omega}_n(h)$ indicates that resampling increases the variance of the Monte Carlo approximation. However, it also equalizes the particle weights which tends to lower the approximation errors in the subsequent iteration of the algorithm.

Mutation Step. Denote the conditional mean and variance associated with the transition kernel $K_n(\theta|\hat{\theta};\zeta_n)$ by $\mathbb{E}_{K_n(\cdot|\hat{\theta};\zeta_n)}[\cdot]$ and $\mathbb{V}_{K_n(\cdot|\hat{\theta};\zeta_n)}[\cdot]$. Because π_n is the invariant distribution associated with the transition kernel K_n, note that if $\hat{\theta} \sim \pi_n$, then

$$\begin{aligned}\int_{\hat{\theta}} \mathbb{E}_{K_n(\cdot|\hat{\theta};\zeta_n)}[h]\pi_n(\hat{\theta})d\hat{\theta} &= \int_{\hat{\theta}}\int_{\theta} h(\theta)K_n(\theta|\hat{\theta};\zeta_n)d\theta\pi_n(\hat{\theta})d\hat{\theta} \\ &= \int_{\theta} h(\theta)\int_{\hat{\theta}} K_n(\theta|\hat{\theta};\zeta_n)\pi_n(\hat{\theta})d\hat{\theta}d\theta \\ &= \int_{\theta} h(\theta)\pi_n(\theta)d\theta = \mathbb{E}_{\pi_n}[h].\end{aligned} \tag{5.29}$$

Using the fact that $\frac{1}{N}\sum_{i=1}^{N} W_n^i = 1$ we can write

$$\begin{aligned}\bar{h}_{n,N} - \mathbb{E}_{\pi_n}[h] &= \frac{1}{N}\sum_{i=1}^{N}\big(h(\theta_n^i) - \mathbb{E}_{K_n(\cdot|\hat{\theta}_n^i;\zeta_n)}[h]\big)W_n^i \\ &\quad + \frac{1}{N}\sum_{i=1}^{N}\big(\mathbb{E}_{K_n(\cdot|\hat{\theta}_n^i;\zeta_n)}[h] - \mathbb{E}_{\pi_n}[h]\big)W_n^i \\ &= I + II,\end{aligned} \tag{5.30}$$

say. Let $\hat{\mathcal{F}}_{n,N}$ be the σ-algebra generated by $\{\hat{\theta}_n^i, W_n^i\}_{i=1}^N$. Notice that conditional on $\hat{\mathcal{F}}_{n,N}$ the weights W_n^i are known and the summands in term I form a triangular array of mean-zero random variables that within each row are independently but not identically distributed

because the (conditional) variance and higher-order moments of $h(\theta_n^i)$ may depend on $\hat{\theta}_n^i$. While term I captures deviations of $h(\theta_n^i)$ from its conditional mean $\mathbb{E}_{K_n(\cdot|\hat{\theta}_n^i;\zeta_n)}[h]$, the second term captures deviations from the conditional mean of $\mathbb{E}_{K_n(\cdot|\hat{\theta}_n^i;\zeta_n)}[h]$ from the tempered posterior $\mathbb{E}_{\pi_n}[h]$.

The large sample behavior of the Monte Carlo approximation in the mutation step, $\bar{h}_{n,N}$, depends on the particle weights W_n^i, which in turn depends on how many iterations ago the resampling step was executed. To simplify the exposition, we assume that $\rho_n = 1$, which implies that $W_n^i = 1$. The convergence of I follows from a SLLN and a CLT for a triangular array of independently and non-identically distributed random variables. The convergence of II is a consequence of (5.28). It can be shown that

$$\begin{aligned} \bar{h}_{n,N} &\xrightarrow{a.s.} \int h(\theta_n)\pi_n(\theta_n)d\theta_n \qquad (5.31) \\ \sqrt{N}\big(\hat{h}_{n,N} - \mathbb{E}_{\pi_n}[h]\big) &\Longrightarrow N\big(0, \Omega_n(h)\big), \quad \Omega_n(h) = \mathbb{E}_{\pi_n}\big[\mathbb{V}_{K_n(\cdot|\hat{\theta};\zeta_n)}[h]\big] + \hat{\Omega}_n\big(\mathbb{E}_{K_n(\cdot|\hat{\theta};\zeta_n)}[h]\big). \end{aligned}$$

To establish the convergence results (5.25), (5.28), and (5.31) in a rigorous manner mainly requires the verification of moment bounds for the various random variables that are being averaged in the Monte Carlo approximations. The recursive form of the asymptotic covariances makes it difficult to use the results in practice. Estimates of the Ω matrices can be obtained by running the SMC algorithm multiple times. Herbst and Schorfheide (2014) provide some high-level assumptions that ensure that choosing the tuning sequences adaptively according to $\{\hat{\rho}_n, \hat{\zeta}_n\}$ does not affect the asymptotic covariance matrices.

5.1.5 Beyond Multinomial Resampling

The resampling step in Algorithm 8 is based on multinomial resampling. While the use of multinomial resampling facilitates the theoretical analysis of the algorithm, in particular the derivation of a CLT, it is not the most efficient resampling algorithm. We will provide a brief overview of alternative resampling algorithms and refer the reader for more detailed treatments to the books by Liu (2001) or Cappé, Moulines, and Ryden (2005) (and references cited therein) as well as Murray, Lee, and Jacob (2014) for a discussion of parallelization of these algorithms.

Resampling algorithms take as input the collection of particle weights $\{\tilde{W}_n^i\}_{i=1}^N$ and produce as output either an ancestry vector or a vector the contains the number of offsprings for each particle. An ancestry vector A_n has elements A_n^i such that $A_n^i = j$ if and only if particle j

is the ancestor of resampled particle j, that is, $\hat{\theta}_n^i = \theta_{n-1}^i$. Alternatively, the offspring vector O_n with elements O_n^i would contain the number of offsprings for each particle θ_{n-1}^i. Both A_n and O_n contain the same information and each one can be transformed into the other. Most algorithms have the unbiasedness property $\mathbb{E}[O_n^i] = \tilde{W}_n^i$. The multinomial resampling embedded in Algorithm 8 can be implemented by computing standardized cumulative weights $\tilde{W}_n^{c,i} = \sum_{j=1}^i \tilde{W}_n^j$ and generating N iid $U[0,N]$ random numbers u^i. The element A^i can then be defined as $LB\big(\{\tilde{W}_n^{c,N}\}, u_i\big)$, where the function $LB(W,u)$ returns the smallest integer i such that the scalar u can be inserted into position i of a vector W, sorted in ascending order, while maintaining the sorting.[2]

The variance of multinomial resampling can be reduced by stratification. Suppose we divide the unit interval into N strata of the form $\mathcal{U}^i = ((i-1)/N, i/N]$ and for each stratum generate a uniform random number u_i. We can still define the ancestor vector A^i as $LB\big(\{\tilde{W}_n^{c,N}\}, u_i\big)$, but now the distribution has changed. For illustrative purposes consider the can $N = 2$ and $\tilde{W}_n^1 \leq 1/2$. For particle $i = 1$, $u_1 \sim U(0, 1/2]$, which means that with probability $2\tilde{W}_n^1$ the value θ_{n-1}^1 is selected and with probability $1 - 2\tilde{W}_n^1$ the value θ_{n-1}^2 is chosen. For particle $i = 2$ we one always choose θ_{n-1}^2 because $u_2 \geq 1/2 \geq \tilde{W}_n^1$. The distribution of offsprings takes the form

$$O_n^i = \begin{cases} 0 & \text{w. prob. } (1 - 2\tilde{W}_n^1) \\ 1 & \text{w. prob. } 2\tilde{W}_n^1 \\ 2 & \text{w. prob. } 0 \end{cases} .$$

For the regular multinomial resampling described above, the distribution of offsprings is

$$O_n^i = \begin{cases} 0 & \text{w. prob. } (1 - \tilde{W}_n^1)^2 \\ 1 & \text{w. prob. } 2(1 - \tilde{W}_n^1)\tilde{W}_n^1 \\ 2 & \text{w. prob. } (\tilde{W}_n^1)^2 \end{cases} .$$

Both resampling algorithms are unbiased but the stratified resampler has a lower variance. The variance reduction extends to $N = 2$ (see, e.g., Cappé, Moulines, and Ryden (2005)). A stratified resampling algorithm to efficiently compute the cumulative offspring function is provided in Murray, Lee, and Jacob (2014).

Stratified resampling aims to reduce the discrepancy between the empirical distribution of the generated draws and the uniform distribution. This is achieved by defining $u^i = (i-1)/N + \xi^i$ where $\xi^i \sim iidU[0, 1/N]$. Alternatively, one could consider the sequence

[2]Suppose that $W = [3, 5, 6, 10]$ and $u = 4$. One could replace either element 1 or element 2 of the vector W without affecting the sorting. The function LB will return the value 1.

$u^i = (i-1)/N + \xi$ where $\xi \sim U[0, 1/N]$. This method is known as systematic resampling. The theoretical properties of systematic resampling algorithms are more difficult to establish because the draws u^i, $i = 1, \ldots, N$, are perfectly correlated. In sequential Monte Carlo applications, this generates cross-sectional dependence of particles.

Let $\lfloor x \rfloor$ denote the floor operator, i.e., the largest integer that is less or equal than $x \geq 0$. The residual resampling algorithm initially assigns $\lfloor \tilde{W}_n^i \rfloor$ offsprings to each particle and then determines the remaining offsprings randomly:

$$O_n^i = \lfloor \tilde{W}_n^i \rfloor + \hat{O}_n^i. \tag{5.32}$$

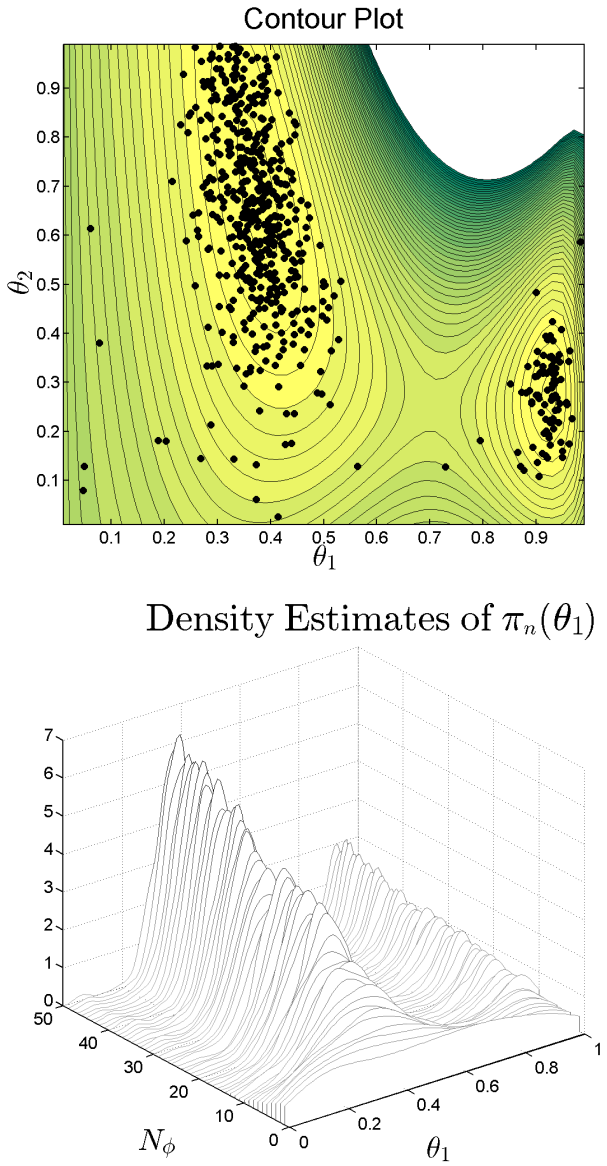
Now only $N - \sum_{i=1}^N \lfloor \tilde{W}_n^i \rfloor$ draws are required and the probability associated with particle i is proportional to $\tilde{W}_n^i - \lfloor \tilde{W}_n^i \rfloor$. The residuals $\hat{O}_n^i$ can be generated with one of the algorithms described above. None of the algorithms discussed thus far is well suited for parallelization because it is necessary to compute the sum of the particle weights (the summation step appears as the last operation the correction step in Algorithm 8). The Metropolis resampling algorithm and the rejection resampling algorithm discussed in Murray, Lee, and Jacob (2014) are designed to avoid collective operations over the weights.

5.2 An Application to A Simple State-Space Model

Before applying the SMC algorithm to a full-fledged DSGE model, we use the stylized state-space model of Section 4.3, which can generate a bimodal likelihood function (see the left panel of Figure 4.4). As before, we simulate $T = 200$ observations given $\theta = [0.45, 0.45]'$, which is observationally equivalent to $\theta = [0.89, 0.22]'$, and use a prior distribution that is uniform on the square $0 \leq \theta_1 \leq 1$ and $0 \leq \theta_2 \leq 1$. Since the state-space model has only two parameters and the model used for posterior inference is correctly specified, the SMC algorithm works extremely well. It is configured as follows. We use $N = 1024$ particles, $N_\phi = 50$ stages, and a linear tempering schedule $\lambda = 1$. In the mutation phase we use $N_{blocks} = 1$ block and $M = 1$ step for the RWMH-V algorithm.

Some of the output of the SMC algorithm is depicted in Figure 5.1. The top panel shows a contour plot of the posterior density as well as draws from this posterior generated by Algorithm 8. The algorithm successfully generates draws from the two high-posterior-density regions. The bottom panel displays the sequence of tempered (marginal) posterior distributions $\pi_n(\theta_1)$. It clearly shows that the tempering dampens the posterior density. While the

Figure 5.1: SMC Posterior Approximations



Notes: The top panel depicts the contours of the posterior as well as draws from $\pi(\theta)$. The bottom panel depicts kernel density estimates of the sequence $\pi_n(\theta)$ for $n = 1, \ldots, 50$.

posterior is still unimodal during the first few stages, a clear bimodal shape as emerged for $n = 10$. As ϕ_n approaches one, the bimodality of the posterior becomes more pronounced. The bottom panel also suggests, that the stage $n = N_\phi - 1$ tempered posterior provides a much better importance sampling distribution for the overall posterior $\pi(\cdot)$ than the stage $n = 1$ (prior) distribution.

Figure 5.2: Adaption of the Algorithm

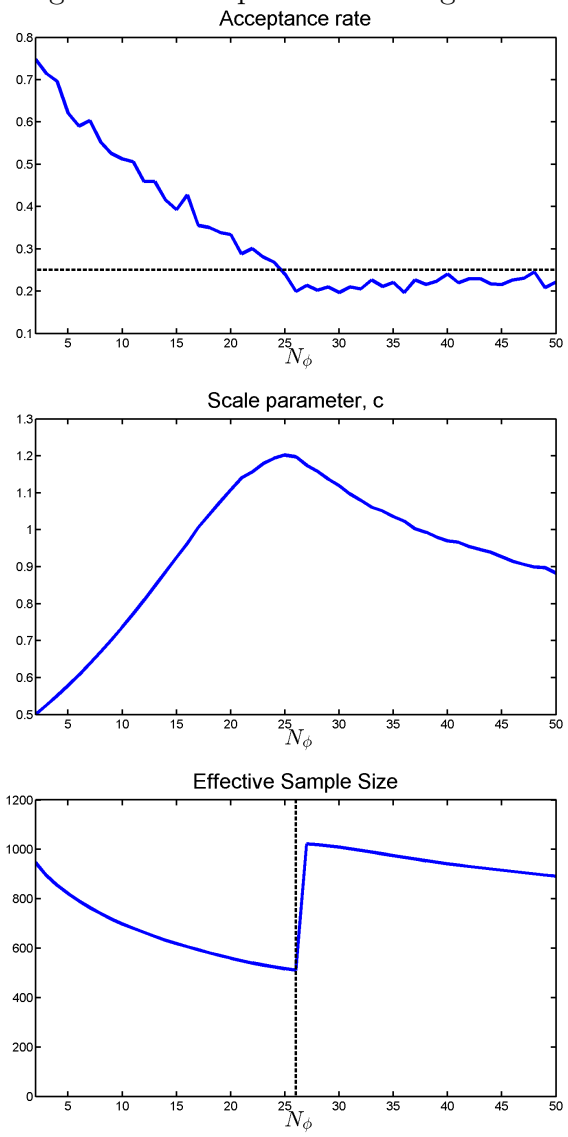
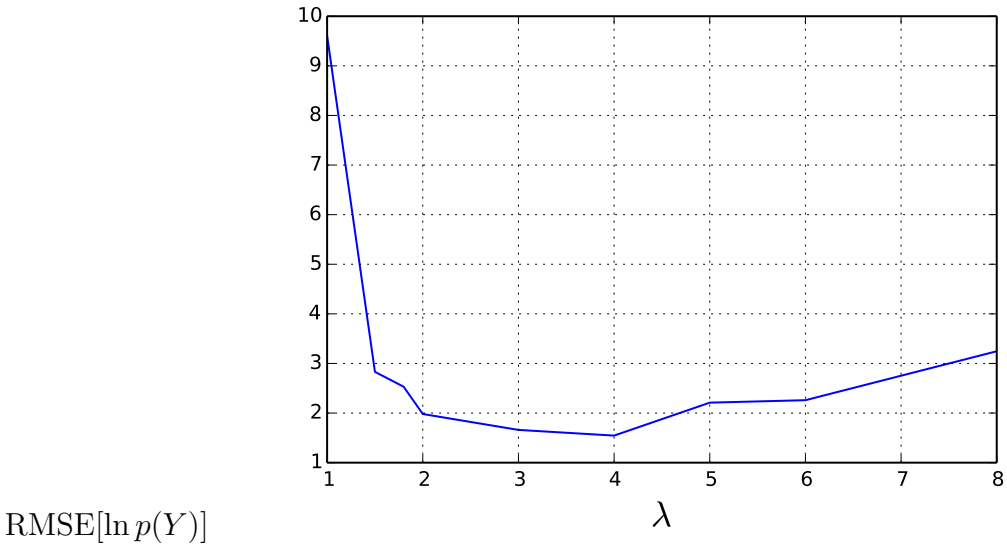


Figure 5.2 illustrates the adaptive choice of the scaling constant c in the RWMH-V mutation step. The algorithm is configured to target an acceptance rate of 25%. The initial value of the scaling constant is 0.5, which leads to an acceptance rate of more than 70% in the first few steps of the algorithm. Gradually, the scaling constant is lowered according to Algorithm 10. In stage $n = 25$ we are reaching the desired acceptance rate. The acceptance rate subsequently drops slighly below 25% which triggers a drop in c. Starting from a value exceeding 900, the effective sample size $\widehat{ESS}_n$ slowly decreases and at $n = 26$ falls below the threshold of $N/2$. This triggers the resampling of particles and in turn $\widehat{ESS}_n$ jumps up toward about 1,000. Thus, qualitatively, the adaption of the algorithm works as desired: the

Figure 5.3: SMC: Importance of λ



Notes: I think we should also plot hairs of $\sqrt{\frac{N\hat{V}[\bar{\theta}]}{V_\pi[\theta]}}$ *as function of* λ *as in Figure 5.4.*

scaling constant c in the RWMH-V algorithm is adjusted to achieve the desired acceptance rate and the particles are resampled if the distribution of weights becomes uneven.

5.3 An Application to the Small-Scale New Keynesian Model

We will now apply the SMC algorithm to conduct posterior inference for the three-equation New Keynesian model (DSGE Model I). We use the same prior specification and the same data set as in Section 4.2. We will illustrate how the accuracy of the SMC approximation changes as we vary the choice of tuning parameters for the algorithm. We run each configuration of the algorithm 50 times verify and compute the cross-sectional variance of the Monte Carlo approximations $\bar{\theta}_{N_\phi,N}$ which we denote by $\hat{V}[\bar{\theta}]$. Because under *iid* sampling the asymptotic variance of Monte Carlo approximation is given by $\mathbb{V}_\pi[\theta]$ we will use $\hat{V}[\bar{\theta}]/(\mathbb{V}_\pi[\theta]/N)$, or transformations thereof, as measures of accuracy. Throughout this section we fix the number of MH steps in the mutation step to be equal to $M = 1$

Figure 5.3 explores the connection between accuracy and the tempering schedule ϕ_n. The results are based on $N =$, $N_\phi =$, $N_{blocks} =$. For $\lambda = 1$ this schedule is linear in n and for

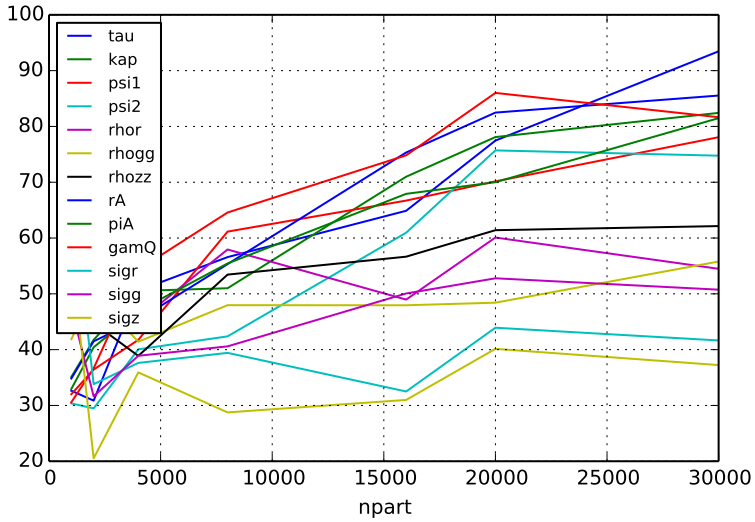
$\lambda > 1$ it is convex. A convex tempering schedule implies that we add very little information in the initial stages of the SMC algorithm to ensure that the particles adapt well to the bridge distribution. As n increases, the incremental amount of likelihood information added in each stage also increases. According to the linear schedule performs very poorly. A choice of λ in the range of 2 to 4 yields the most accurate approximation and for values larger than 5 the performance slowly deteriorates. Note that the choice of λ has essentially no effect the number of likelihood evaluations and on the computational time (except that poor choices of λ may require additional resampling steps). In the subsequent experiments we let $\lambda = 2$.

According to the results discussed in Section 5.1.4, the Monte Carlo approximations should satisfy a CLT, which means that the cross-sectional variance should decay at the rate $1/N$. This implies that the scaled ratio of standard deviations $\sqrt{\hat{V}[\theta]/(V_\pi[\theta]/N)}$ should be approximately constant for large values of N. We show bundles of the scaled variance ratio in Figure 5.4, where each hairline corresponds to one of the DSGE model parameters. Here $N_{blocks} =$. The top panel shows results for $N_\phi = 25$ stages and the bottom panel results for $N_\phi = 100$ stages. A comparison between $N_\phi = 25$ and $N_\phi = 100$ indicates that increasing the number of stages raises the precision (and lowers the inefficiency factor) of the Monte Carlo approximation. For $N_\phi = 25$ the inefficiency profiles are fairly flat for more than 20,000 particles. For $N_\phi = 100$, the inefficiency profiles are increasing.

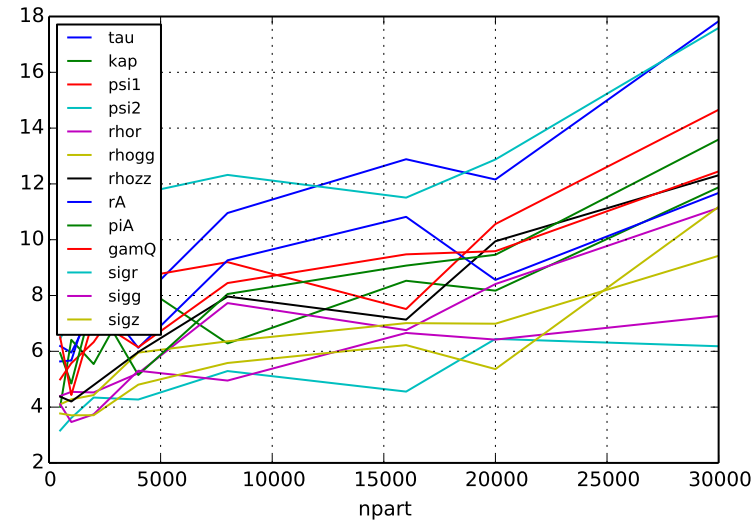
Finally, Figure 5.5 explores the trade-offs between number of particles N, number of stages N_ϕ, and number of blocks N_{blocks} in the mutation step. *** we dropped the N factor because N is different for each line *** The experiments are designed such that we keep the number of likelihood evaluations constant. The top panel indicates that a large number of particles, e.g., $N = 4,000$, combined with a moderate number of stages, e.g., $N_\phi = 25$, delivers a more accurate approximation than a small number of particles, e.g., $N = 250$, and a large number of stages, e.g., $N_\phi = 400$. Of course, if we would to reduce the number of stages more drastically, the accuracy would at some point deteriorate. A very large number of stages is not helpful because a lot of computational resources are allocated to approximating very similar bridge distributions. The bottom panel depicts the effect of blocking. In this small-scale model, the effect of blocking, i.e., using 4 blocks and in combination with 250 particles instead of 1 block and 1,000 particles improves the accuracy only slightly for 3 out of the 7 parameters depicted in the figure. For the policy rule coefficients ψ_1 and ψ_2 and the inverse elasticity of substitution, a large number of particles appears to be more useful than a large number of blocks.

Figure 5.4: Sequential Monte Carlo Central Limit Theorem

$N_\phi = 25$



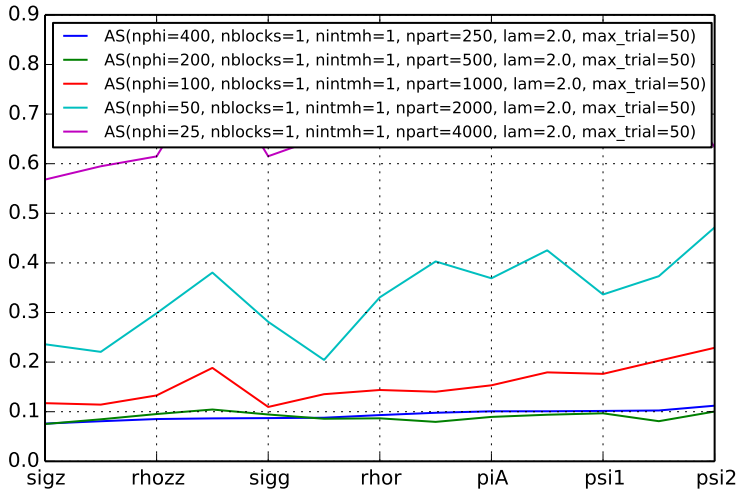
$N_\phi = 100$



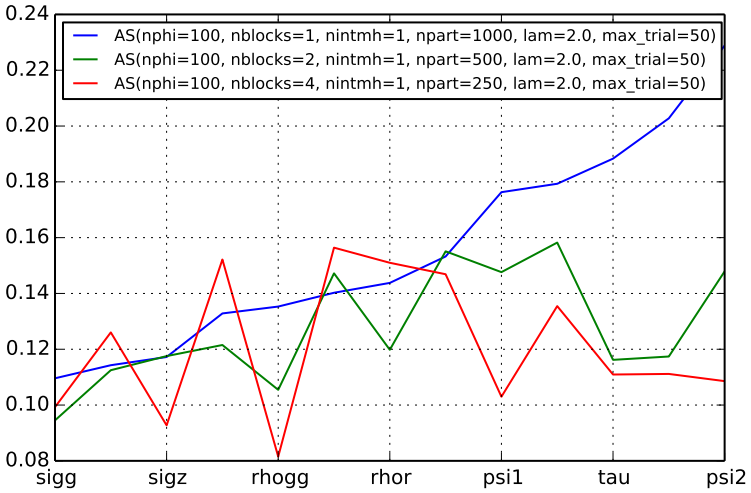
Notes: Each panel shows $\sqrt{\frac{\hat{V}[\bar{\theta}]}{V_\pi[\theta]/N}}$ for each parameter as a function of N.

Figure 5.5: SMC: Relative Importance of Tuning Parameters

N_ϕ vs N



N_{blocks} vs N



Notes: Each panel shows $\sqrt{\frac{\hat{V}[\bar{\theta}]}{V_\pi[\theta]}}$.

Chapter 6

Case Studies

This chapter considers three different empirical applications: the estimation of a small-scale New Keynesian model with correlated shocks in Section 6.1, the estimation of the Smets-Wouters model under a more diffuse prior distribution in Section 6.2, and finally the estimation of DSGE model designed to analyze fiscal policy questions in Section 6.3. In each of these applications we highlight distinct non-elliptical features of the posterior distribution and document to what extent the MH and SMC algorithms of Chapters 4 and 5 accurately approximate the posteriors.

6.1 New Keynesian Model with Correlated Shocks

The fit of DSGE models can be improved either by enriching the endogenous propagation mechanism of the model or by generalizing the law of motion for the exogenous shocks. Most of the DSGE model literature has focused on augmenting the basic neoclassical stochastic growth model with more sophisticated economic mechanism, e.g., frictions in the adjustment of labor and capital inputs, or costs of changing nominal prices and wages, or information imperfections. These mechanisms interact with the effects of monetary and fiscal policy and incorporating empirically-important mechanisms into a DSGE model is crucial for obtaining reliable policy predictions. At the same time, the exogenous shocks are typically assumed to follow independent AR(1) processes. However, *a priori* there is nothing that rules out a correlation between, say, neutral and investment-specific technology shocks or between nominal price and wage markup shocks. Nor is it *a priori* unreasonable to assume that the exogenous shocks follow an AR(1) process rather than a richer ARMA(p,q) process. The

only reason that many authors to prefer with rather simple exogenous processes is that one of the goals of the DSGE research program is to develop economic mechanisms that can generate the observed comovements and persistence of macroeconomic time series from a set of uncorrelated exogenous shocks.

Nonetheless, in environments in which model fit is important, e.g., central bank forecasting with DSGE models, the generalization of the law of motion of exogenous shocks is a plausible modeling strategy. Most prominently, this has been done to a limited extent by using ARMA(1,1) shocks instead of AR(1) shocks and by introducing correlation between certain shock innovations in the context of the SW model which we examine in more detail in Section 6.2. Curdia and Reis (2010) take this approach a step further and consider a fairly general vector-autoregressive law of motion for the exogenous processes of a small-scale DSGE model. While this modeling strategy is helpful in overcoming DSGE model misspecification, it also introduces potential identification problems. The more flexible and densely parameterized the law of motion of the exogenous shocks, the more difficult it becomes to identify the shock parameters and the parameters associated with the endogenous propagation mechanism jointly. From a computational perspective, this may introduce multi-modal posterior distributions, which are the focus of the remainder of this section.

6.1.1 Model Specification

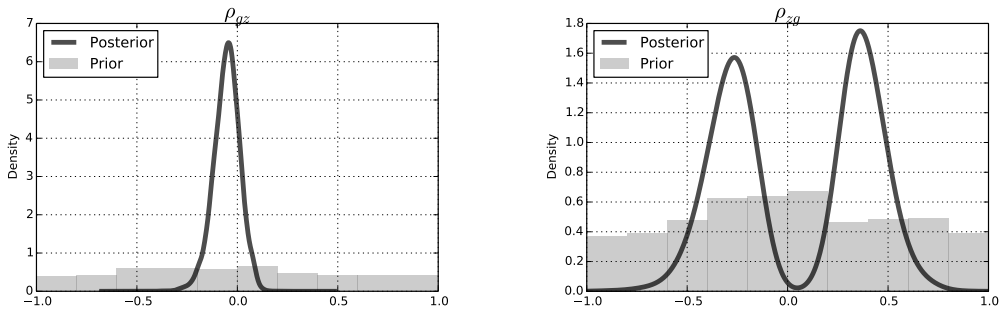
In the small-scale DSGE model of Chapter 1.1 the technology growth shock $\hat{z}_t$ and the government spending shock $\hat{g}_t$ evolve according to independent AR(1) processes. We now replace the two AR(1) processes in (1.24) and (1.25) by the following VAR process:

$$\begin{bmatrix} \hat{z}_t \\ \hat{g}_t \end{bmatrix} = \begin{bmatrix} \rho_z & \rho_{zg} \\ \rho_{gz} & \rho_g \end{bmatrix} \begin{bmatrix} \hat{z}_{t-1} \\ \hat{g}_{t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{z,t} \\ \epsilon_{g,t} \end{bmatrix}, \quad \begin{bmatrix} \epsilon_{z,t} \\ \epsilon_{g,t} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_z^2 & 0 \\ 0 & \sigma_g^2 \end{bmatrix} \right). \quad (6.1)$$

Thus, the small-scale DSGE model with correlated exogenous shocks is comprised of (2.1) and (6.1). Unlike in the version of the model estimated in Chapters 4.2 and 5.3, the lagged government spending (or demand) shock potentially effects current technology growth or vice versa. A non-zero coefficient ρ_{gz} could be interpreted as a reduced-form fiscal policy rule in which government spending is increased if the supply conditions are poor. Likewise, a positive coefficient ρ_{zg} could potentially capture productivity enhancing public infrastructure investments. While these interpretations suggests that $\rho_{gz} < 0$ whereas $\rho_{zg} > 0$, we use more agnostic priors of the form

$$\rho_g, \rho_z \sim U[0,1], \quad \rho_{gz}, \rho_{zg} \sim U[-1,1]. \quad (6.2)$$

Figure 6.1: Posterior of ρ_{gz} and ρ_{zg}



Notes: The two panels depict histograms of prior distributions and kernel density estimates of the posterior densities.

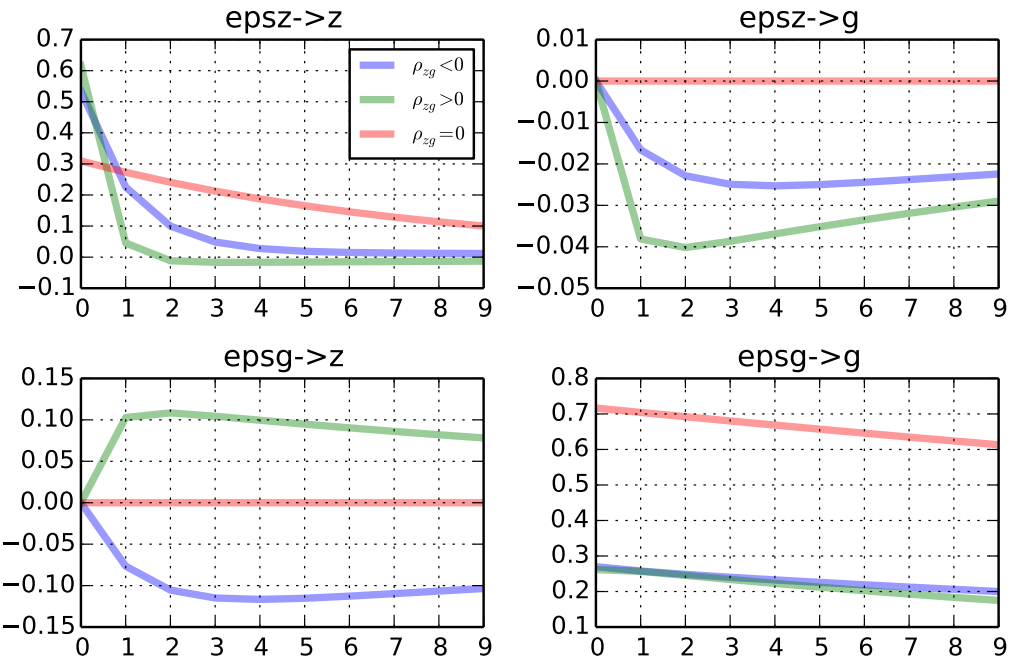
The marginal prior distributions for the remaining DSGE model parameters are identical to those in Table 2.2. The joint prior distribution is truncated to ensure stationarity of $\hat{z}_t$ and $\hat{g}_t$ and determinacy of the overall system.

6.1.2 Estimation Results from a Highly Accurate SMC Run

We estimate the modified small-scale DSGE model using the same data as in Chapters 4.2 and 5.3. We begin the numerical analysis by examining the posterior distribution based on a highly accurate run of the SMC algorithm. The most striking feature of this posterior is depicted in Figure 6.1 which shows the marginal prior and posterior distributions of ρ_{gz} and ρ_{zg}. The marginal prior distributions are represented by the gray shaded histograms. After the truncation induced by the stationarity restriction the marginal prior distributions of the two parameters are no longer uniform, but they are unimodal and spread out across the unit interval. The posterior distributions are represented by kernel density estimates. While the posterior density of ρ_{gz} is unimodal and sharply peaks around zero, the posterior of ρ_{zg}, on the other hand, is bimodal with peaks at approximately -0.3 and 0.3 respectively.

IRFs associated with the parameter estimates are depicted in Figures 6.2 and 6.3. The panels in each figure show three types of IRFs: responses associated with $\rho_{zg} < 0$, $\rho_{zg} > 0$ and $\rho_{zg} = 0$. The $\rho_{zg} = 0$ responses are identical to the posterior mean responses of the benchmark small-scale New Keynesian DSGE model reported in Chapter 4.2 (see Figure 4.3). The other two sets of responses are constructed from the conditional posterior distributions $\theta|(Y, \rho_{gz} > 0)$ and $\theta|(Y, \rho_{gz} < 0)$ as $\mathbb{E}[IRF|Y, \rho_{gz} > 0]$ and $\mathbb{E}[IRF|Y, \rho_{gz} < 0]$, respectively.

Figure 6.2: Correlated Shocks Model: Impulse Responses (Part 1)

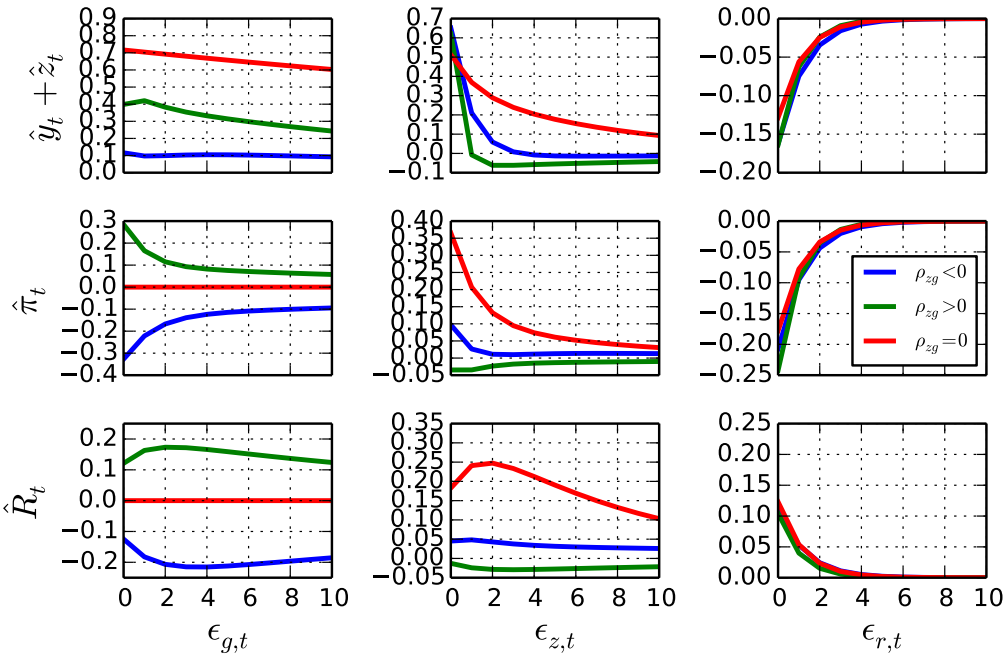


Notes: The graphs depict posterior means of impulse response functions based on the conditional posteriors $\theta|(Y, \rho_{gz} > 0)$, and $\theta|(Y, \rho_{gz} < 0)$. For comparison, the graph also shows responses from the small-scale DSGE model with uncorrelated shocks (see Chapter 4.2).

Figure 6.2 shows the responses of the exogenous processes to government spending and technology growth shock innovations. Under the benchmark specification of the DSGE model, the exogenous shocks are independent of each other, which means that the demand shifter processes $\hat{g}_t$ does not respond to the technology shock innovation $\epsilon_{z,t}$ and vice versa. In the DSGE model with correlated exogenous shocks, on the other hand, there are spillovers. The government spending process drops slightly (2 to 4 basis points) in response to a 50 to 60 basis points increase in $\epsilon_{z,t}$. The response is qualitatively very similar for the two modes of the posterior, which is consistent with the unimodal shape of the marginal posterior of ρ_{gz}. More interesting is the response of technology growth to a government spending (or general demand) shock innovation. The impulse responses in the bottom left panel of Figure 6.2 reflect the bimodal shape of the ρ_{zg} posterior. If $\rho_{zg} > 0$ ($\rho_{zg} > 0$) then technology growth increases (decreases) by about 10 basis points in response to a 25 basis point $\epsilon_{g,t}$ shock.

Figure 6.3 depicts the impulse responses of output, inflation, and interest rates. The effect of a monetary policy shock is approximately the same for the two modes of the model

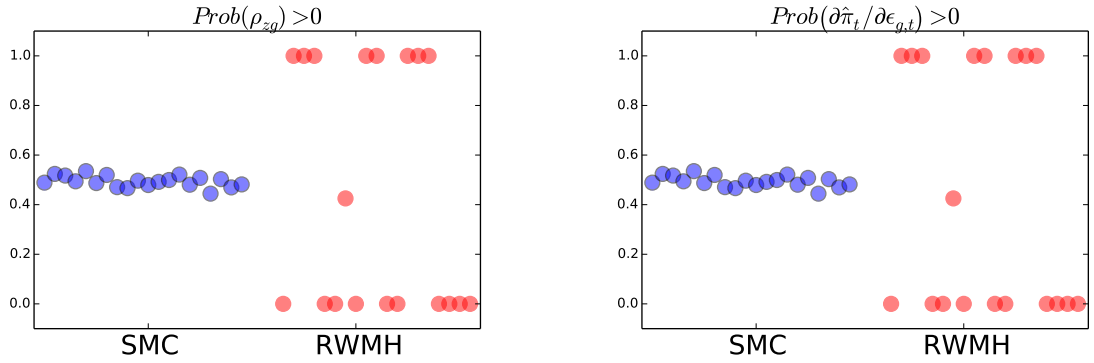
Figure 6.3: Correlated Shocks Model: Impulse Responses (Part 2)



Notes: The graphs depict posterior means of impulse response functions based on the conditional posteriors $\theta|(Y, \rho_{gz} > 0)$, and $\theta|(Y, \rho_{gz} < 0)$. For comparison, the graph also shows responses from the small-scale DSGE model with uncorrelated shocks (see Chapter 4.2).

with correlated shocks. The monetary policy responses closely resemble the IRFs obtained from the baseline version of the small-scale DSGE model with uncorrelated shocks. The IRFs for the government spending and the technology growth shock, on the other hand, are markedly different for the two modes of the correlated shocks model and the baseline model. In the baseline model, neither inflation nor interest rates respond to a change in government spending. In the correlated shocks model, a rise in government spending also triggers a change in technology. We saw that depending on the mode, technology growth either rises or falls in response to a positive $\epsilon_{g,t}$ innovation. As a consequence, inflation and interest rates may either rise or fall in response to a positive demand shock, depending on which mode is selected. Moreover, because a drop in technology growth is associated with lower output, the magnitude of the output response also differs significantly. The IRFs of inflation and interest rates to a technology growth shock are generally more muted under the correlated shocks specification than under the baseline specification. Conditional on $\rho_{zg} > 0$ these responses are slightly positive, whereas for $\rho_{zg} < 0$ they are slightly negative.

Figure 6.4: Posterior Probability of $\rho_{zg} > 0$ and $\partial\hat{\pi}_t/\partial\hat{\epsilon}_{g,t} > 0$



Notes: Each dot (20 in total) correspond to one run of the SMC algorithm (blue) or the RWMH algorithm (red).

Table 6.1: Configuration of Algorithms for Correlated Shocks Model

RWMH-V	SMC
$N = 100,000$	$N = 4,800$
$N_{burn} = 50000$	$N_{\phi} = 500$
$N_{blocks} = 1$	$N_{blocks} = 6$, $M = 1$
$c = 0.125$	$\lambda = 4$
Run time: 1 min (1 core)	Run time: 1 min (12 cores)

Notes: We run each algorithm 20 times.

6.1.3 Comparison of RWMH-V and SMC Performance

Given the stylized nature of the small-scale DSGE model we do not offer a detailed economic interpretation of the estimation results. The small-scale DSGE model places strong restrictions on the autocovariance function of output growth, inflation, and interest rates and the generalization of the law of motion of the exogenous shocks relaxes these restrictions somewhat, pointing toward an omitted endogenous propagation mechanism. In the remainder of this section we direct our attention to the computational challenges created by the bimodal posterior. We will compare the accuracy of the standard RWMH-V algorithm to the SMC algorithm.

Accuracy Assessment and Tuning of Algorithms. To compare the performance of

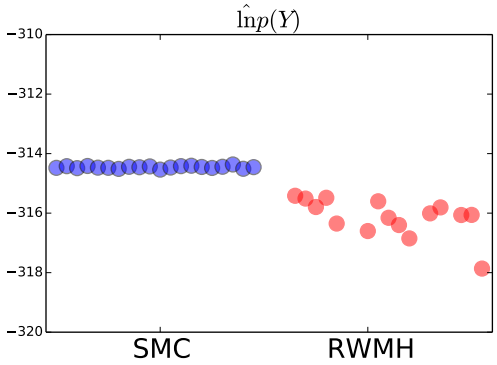
the 1-Block RWMH-V algorithm and a SMC algorithm, we run each of these algorithms 20 times and evaluate the posterior probability that $\rho_{zg} >$ and the probability that inflation increases in response to a government spending shock. The 1-Block RWMH-V algorithm is initialized with a random draw from the prior distribution of the DSGE model parameters and it runs for about 1 minute on a single processor. The SMC algorithm uses 4,800 particles, 6 blocks, 500 stages, and $\lambda = 4$. The run time of the SMC algorithm is about 1 minute on 12 processors. Allocating more computational resources to the RWMH algorithm did not change the result. A summary is provided in Table 6.1.3.

Results. The two panels of Figure 6.4 show estimates of posterior probability that $\rho_{zg} > 0$ and inflation responds positively to an expansionary government spending shock. The bimodal posterior density depicted in Figure 6.1 in conjunction with the IRF plots in Figure 6.3 imply that these posterior probabilities should be around 50%. In order correctly estimate these probabilities the posterior sampler has to generate draws from the two high-posterior-density areas in the correct proportion. The Monte Carlo approximations of the posterior probabilities obtained from the SMC algorithm are very stable and close to 50% in all 20 runs. The RWMH algorithm, on the other hand, generates estimates that essentially switch between zero and one across runs, depending on whether the sampler gets stuck near the $\rho_{zg} > 0$ mode or the $\rho_{zg} < 0$ mode. In other words, the RWMH sampler does not travel frequently enough between the two modes in order to generate draws from the two high-posterior-probabilities areas of the parameter space in the correct proportion. Increasing the number of draws from 100,000 to 1,000,000 did not correct the problem.

Estimates of the marginal data density associated with the generalized shock model are depicted in Figure 6.5. The Monte Carlo approximation generated by the SMC algorithm is very stable, whereas the approximation obtained with the modified harmonic mean estimator described in Chapter 4.6 appears to be downward biased (as it misses a high-likelihood region of the parameter space) and highly variable. In fact, four estimates constructed from the RWMH output were outside of the limits of the chart.

Overall allowing for correlated technology and demand shocks is import for model fit. Table 6.2 displays estimates of the log marginal data density for the small-scale DSGE model under the standard (see Chapter x) and diffuse priors. We use the estimates computing from the SMC sampler in both cases because of its high accuracy. The baseline prior model has a log marginal data density of -346.2, over 30 points below that of the diffuse prior model. Under the calculus of probabilities associated with Bayesian model comparison, the marginal

Figure 6.5: Marginal Data Density Approximation: SMC versus Modified Harmonic Mean



Notes: Each dot (20 in total) correspond to one run of the SMC algorithm (blue) or the RWMH algorithm (red). The SMC algorithm automatically generates and estimate of the MDD, for the RWMH algorithm we use Geweke's modified harmonic mean estimator. 4 estimates from the RWMH output were off the chart. The estimated values are -333.02, -362.8, -372.7, - 499.36.

Table 6.2: Marginal Data Density: Effect of Prior

Model	Mean($\ln \hat{p}(Y)$)	Std. Dev.($\ln \hat{p}(Y)$)
Baseline Prior	-346.18	0.05
Diffuse Prior	-314.46	0.04

Notes: Table shows mean and standard deviation of SMC-based estimate of the log marginal data density, computed over twenty runs of the SMC sampler under each prior. The hyper-parameters used for the SMC algorithm are given in Table 6.1.3.

data densities place overwhelmingly odds on the diffuse prior model, indicating that the $AR(1)$ restrictions are severe.

6.2 Smets-Wouters Model

We now turn to the estimation of the SW model. The SW model is considered to be a medium-scale DSGE model. It is considerably larger, both in terms of state variables as well as in terms of parameters to be estimated, than the small-scale New Keynesian model considered thus far. The SW model forms the core of many of the DSGE models that are

used in central banks to conduct monetary policy analysis and generate DSGE model-based forecasts. The subsequent empirical illustration is based on Herbst and Schorfheide (2014). However, unlike in Herbst and Schorfheide (2014) instead of considering the accuracy of approximations of posterior means, we will focus on the accuracy of quantiles of the posterior distribution.

6.2.1 Model Specification

Our version of the SW model is identical to the version presented in Smets and Wouters (2007). The log-linearized equilibrium conditions, steady states, and measurement equations are reproduced in Appendix A.1. The model is estimated using the growth rates of GDP, aggregate consumption, and investment; the log level of hours worked; and price inflation, wage inflation, and the federal funds rate. Our estimation differs from Smets and Wouters (2007) in that we are using a more diffuse prior distribution.

Some researchers have argued that the prior distribution originally used by SW is implausibly tight, in the sense that it seems hard to rationalize based on information independent of the information in the estimation sample. For instance, the tight prior on the steady-state inflation rate is unlikely to reflect *a priori* beliefs of someone who has seen macroeconomic data only from the 1950s and 1960s. At the same time, this prior has a strong influence on the empirical performance of the model, as discussed in Del Negro and Schorfheide (2013). Closely related, Müller (2011) derives an analytical approximation for the sensitivity of posterior means to shifts in prior means and finds evidence that the stickiness of prices and wages is driven substantially by the priors.

One side benefit of tight prior distributions is that they tend to smooth out the posterior surface by down-weighting areas of the parameter space that exhibit local peaks in the likelihood function but are deemed unlikely under the prior distribution. Moreover, if the likelihood function contains hardly any information about certain parameters and is essentially flat with respect to these parameters, tight priors induce curvature in the posterior. In both cases the prior information stabilizes the posterior computations. For simulators such as the RWMH this is crucial, as they work best when the posterior is well-behaved.

The results presented below are based on a more diffuse prior, which is identical to the one that we used in Herbst and Schorfheide (2014). For parameters on the unit interval we replace Beta distributions by uniform distributions. Morever, we scale the prior variances of the other parameters by a factor of three – with the exception that we leave the priors for

the shock standard deviations unchanged. A table with the full specification of the diffuse prior is available in Appendix A.1.

6.2.2 Estimation Results from a Highly-Accurate SMC Run

Table 6.3 summarizes the estimates of the quantiles of the marginal posterior distribution for each DSGE model parameter. The estimates are obtained from the output of a highly-accurate run of the SMC algorithm. The quantile estimates are computed as order statistics by sorting the posterior draws $\{\theta_j^i\}_{i=1}^N$ for each element j of the parameter vector θ. Alternatively, the sample quantiles can be computed by solving the following minimization problem:

$$\hat{q}_\tau(\theta_j) = \text{argmin}_q \left[(1-\tau)\frac{1}{N}\sum_{i:\,\theta_j^i<q}(\theta_j^i - q) + \tau\frac{1}{N}\sum_{i:\,\theta_j^i\geq q}(\theta_j^i - q) \right]. \tag{6.3}$$

This is a special case of a quantile regression (see Koenker and Bassett (1978)) in which the regressor is simply a constant term. The quantiles can be used to construct equal-tail-probability credible intervals. While these credible intervals are typically not the shortest intervals that have a pre-specified posterior coverage probability, they are easier to compute than highest-posterior-density intervals and frequently reported in practice. Under direct (iid) sampling from the posterior distribution the accuracy of the quantile estimates is given by the following CLT:

$$\sqrt{N}(\hat{q}_\tau - q_\tau) \Longrightarrow N\left(0, \frac{\tau(1-\tau)}{\pi^2(q_\tau)}\right), \tag{6.4}$$

where $\pi(\theta)$ is the posterior density.[1] The further the quantile in the tails of the posterior distribution, the less precise the its estimate. We will use an estimate of the asymptotic variance in (6.4) to standardize the Monte Carlo variance of the posterior samplers.

6.2.3 Comparison of RWMH-V and SMC Performance

In the remainder of this section we compare the accuracy of the quantile estimates obtained from the RWMH-V and the SMC algorithm. The computations are executed exactly as de-

[1] If the posterior distribution of θ_j is $N(\bar{\theta}_j, \bar{V}_{\theta_j})$ then $q_\tau(\theta_j) = \bar{\theta}_j + \Phi_N^{-1}(\tau)\sqrt{\bar{V}_{\theta_j}}$, where $\Phi_N(\cdot)$ is the cdf of a $N(0,1)$. In turn, $\pi(q_\tau) = \phi_N(\Phi_N^{-1}(\tau))/\sqrt{V_{\theta_h}}$.

Table 6.3: SW Model with Diffuse Prior: Quantiles of Posterior

	Quantile τ [%]						Quantile τ [%]				
	2.5	5.0	50	95	97.5		2.5	5.0	50	95	97.5
φ	3.58	4.23	7.96	12.62	13.43	α	0.13	0.14	0.17	0.20	0.21
σ_c	1.28	1.33	1.63	2.03	2.12	ρ_a	0.95	0.96	0.97	0.98	0.98
h	0.55	0.59	0.70	0.78	0.80	ρ_b	0.01	0.02	0.16	0.44	0.58
ξ_w	0.77	0.80	0.96	0.99	1.00	ρ_g	0.97	0.97	0.99	1.00	1.00
σ_l	1.17	1.38	2.91	5.27	5.83	ρ_i	0.59	0.61	0.72	0.83	0.85
ξ_p	0.58	0.60	0.73	0.82	0.84	ρ_r	0.00	0.00	0.04	0.14	0.17
ι_w	0.29	0.37	0.76	0.97	0.99	ρ_p	0.78	0.81	0.92	1.00	1.00
ι_p	0.00	0.01	0.09	0.29	0.33	ρ_w	0.15	0.20	0.72	1.01	1.02
ψ	0.45	0.50	0.76	0.96	0.98	μ_p	0.38	0.47	0.79	0.98	1.00
Φ	1.46	1.50	1.71	1.94	1.99	μ_w	0.05	0.09	0.68	0.99	1.00
r_π	2.02	2.12	2.75	3.52	3.67	ρ_{ga}	0.22	0.25	0.43	0.61	0.64
ρ	0.83	0.84	0.88	0.92	0.92	σ_a	0.40	0.41	0.46	0.51	0.52
r_y	0.07	0.08	0.15	0.24	0.26	σ_b	0.16	0.18	0.24	0.29	0.29
$r_{\Delta y}$	0.21	0.22	0.28	0.35	0.36	σ_g	0.49	0.50	0.54	0.60	0.61
π	0.34	0.41	0.87	1.23	1.31	σ_i	0.37	0.39	0.46	0.55	0.57
$100(\beta^{-1}-1)$	-0.00	-0.00	0.04	0.19	0.22	σ_r	0.21	0.22	0.24	0.27	0.27
l	-3.58	-3.00	-0.08	2.93	3.54	σ_p	0.09	0.09	0.13	0.23	0.25
γ	0.36	0.37	0.40	0.44	0.45	σ_w	0.21	0.22	0.25	0.30	0.30

Table 6.4: Configuration of Algorithms for SW Model (Diffuse Prior)

RWMH-V	SMC
$N = 10,000,000$	$N = 12,000$
$N_{burn} = 5,000,000$	$N_\phi = 500$
$N_{blocks} = 1$	$N_{blocks} = 6$, $M = 1$
$c = 0.08$	$\lambda = 2.1$
Run time: 20:00 hours (1 core)	Run time: 2:30 hours (24 cores)

Notes: We run each algorithm 50 times.

scribed in Herbst and Schorfheide (2014). For convenience, we reproduce the most important details.

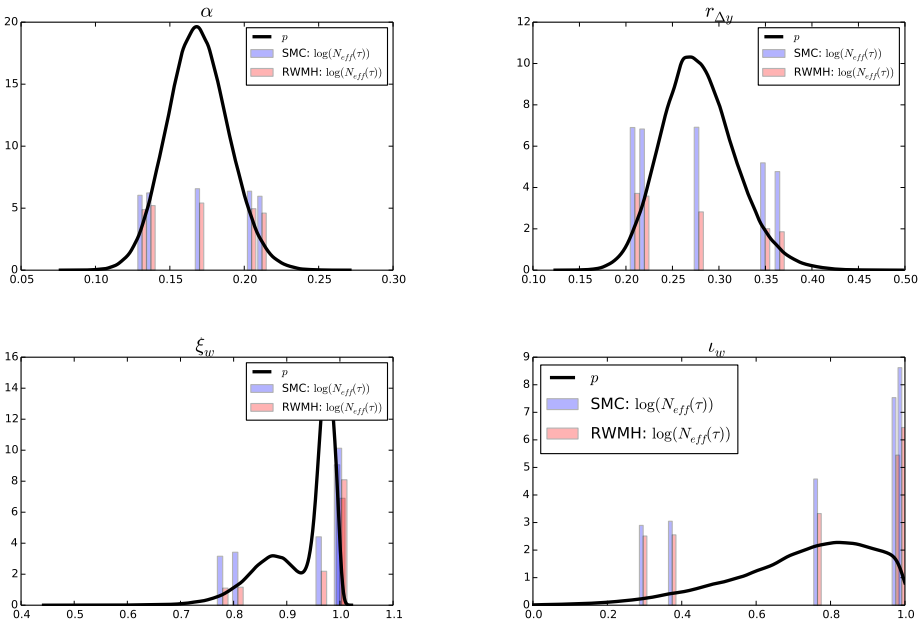
Accuracy Assessment and Tuning of Algorithms. To assess the precision of the Monte Carlo approximations, we run both algorithms 50 times and compute standard deviations of quantile estimates across runs. We constrained the processing time to be roughly the same across algorithms. The SMC algorithm runs about 2 hours and 30 minutes using 24 processors in parallel. In principle, we could instead run 24 copies of the RWMH on separate processor cores and merge the results afterwards. This may reduce sampling variance if each of the RWMH chains has reliably converged to the posterior distribution. However, if there is a bias in the chains – because of, say, the failure to mix on a mode in a multimodal posterior or simply a slowly converging chain – then merging chains will not eliminate that bias. Moreover, choosing the length of the "burn-in" phase may become an issue as discussed in Rosenthal (2000). Instead, we use a poor-man's parallelization of the RWMH algorithm. It is possible to parallelize MH algorithms via pre-fetching as discussed in Strid (2009). Pre-fetching tries to anticipate the points in the parameter space that the MH algorithm is likely to visit in the next k iterations and executes the likelihood evaluation for these parameter values in parallel. Once the likelihood values have been computed one can quickly determine the next k draws. While coding the parallel MCMC algorithm efficiently is quite difficult, the simulation results reported in Strid (2009) suggest that a parallelization using 24 processors would lead to a speedup factor of eight at best. Thus, in our poor-man's parallelization, we simply increase the running time of the RWMH algorithm on a single CPU by a factor of eight. This results in approximately 10 million draws.

The hyperparameters of the SMC algorithm are $N = 12,000$, $N_\phi = 500$, $\lambda = 2.1$, and $N_{blocks} = 6$, $M = 1$. In this application we follow Herbst and Schorfheide (2014) and use a mixture proposal distribution in the mutation step of the SMC algorithm:

$$\begin{aligned} &\vartheta_b|(\theta^i_{n,b,m-1}, \theta^i_{n,-b,m}, \theta^*_{n,b}, \Sigma^*_{n,b}) \\ &\sim \alpha N\left(\theta^i_{n,b,m-1}, c_n^2\Sigma^*_{n,b}\right) + \frac{1-\alpha}{2}N\left(\theta^i_{n,b,m-1}, c_n^2 diag(\Sigma^*_{n,b})\right) \\ &\quad + \frac{1-\alpha}{2}N\left(\theta^*_{n,b}, c_n^2\Sigma^*_{n,b}\right). \end{aligned} \tag{6.5}$$

The choice of this mixture proposal is based on ideas in in Kohn, Giordani, and Strid (2010) on how to improve MH algorithms for DSGE models. The first part corresponds to the standard random-walk proposal, the second part sets the off-diagonal elements to to zero, and the third part is an independence MH proposal. In the implementation of the algorithm the vector of means θ^*_n and the the covariance matrix Σ^*_n are replaced by SMC approximations constructed after the correction step. We set the weight on the mixture

Figure 6.6: Marginal Posterior Densities and Precision of Quantile Approximations



Notes:

components to $\alpha = 0.1$. The choice of $N_\phi = 500$ ensured that the bridge distributions were never too "different." The parameter λ was calibrated by examining the correction step at $n = 1$. Essentially, we increased λ until the effective sample size after adding the first piece of information from the likelihood was at least $10,000$; roughly speaking, 80% of the initial particles retained substantial weight. We settled on the number of blocks by examining the behavior of the adaptive scaling parameter c in a preliminary run. Setting $N_{blocks} = 6$ ensured that the proposal variances were never scaled down too much for sufficient mutation. For the RWMH algorithm, we scale the proposal covariance to achieve an acceptance rate of approximately 30% over 5 million draws after a burn-in period of 5 million. Each RWMH chain was initialized with a draw from the prior distribution. A summary of the configuration of the algorithms is provided in Table 6.2.3.

Results. Figure 6.6 depicts estimates of marginal posterior densities for four parameters: the capital share parameter α, the policy rule coefficient on output growth $r_{\delta y}$, the wage stickiness parameter ζ_w, and the degree of wage indexation to lagged inflation and productivity growth ι_w. The posterior of α is fairly symmetric around its mean/mode, the posterior of $r_{\Delta y}$ is skewed toward the right, the posterior of ζ_w is bimodal, and the posterior of ι_w has a long left tail.

The multimodal features of the posterior distribution are discussed in detail in Herbst and Schorfheide (2014). At one of the modes the values of the wage stickiness parameter, ξ_w, and wage indexation parameter, ι_w, are relatively low, while the parameters governing the exogenous wage markup process imply a lot of persistence. At the other mode, the relative importance of endogenous and exogenous propagation is reversed. The persistence of measured wages is captured by ζ_w and ι_w that are close to one. The multimodality of the joint posterior translates into a bimodal marginal posterior density of for the wage stickiness parameter ζ_w, which peaks around 0.87 and 0.97, respectively.

In addition to the posterior densities, the Figure 6.6 also shows a measure of efficiency defined as

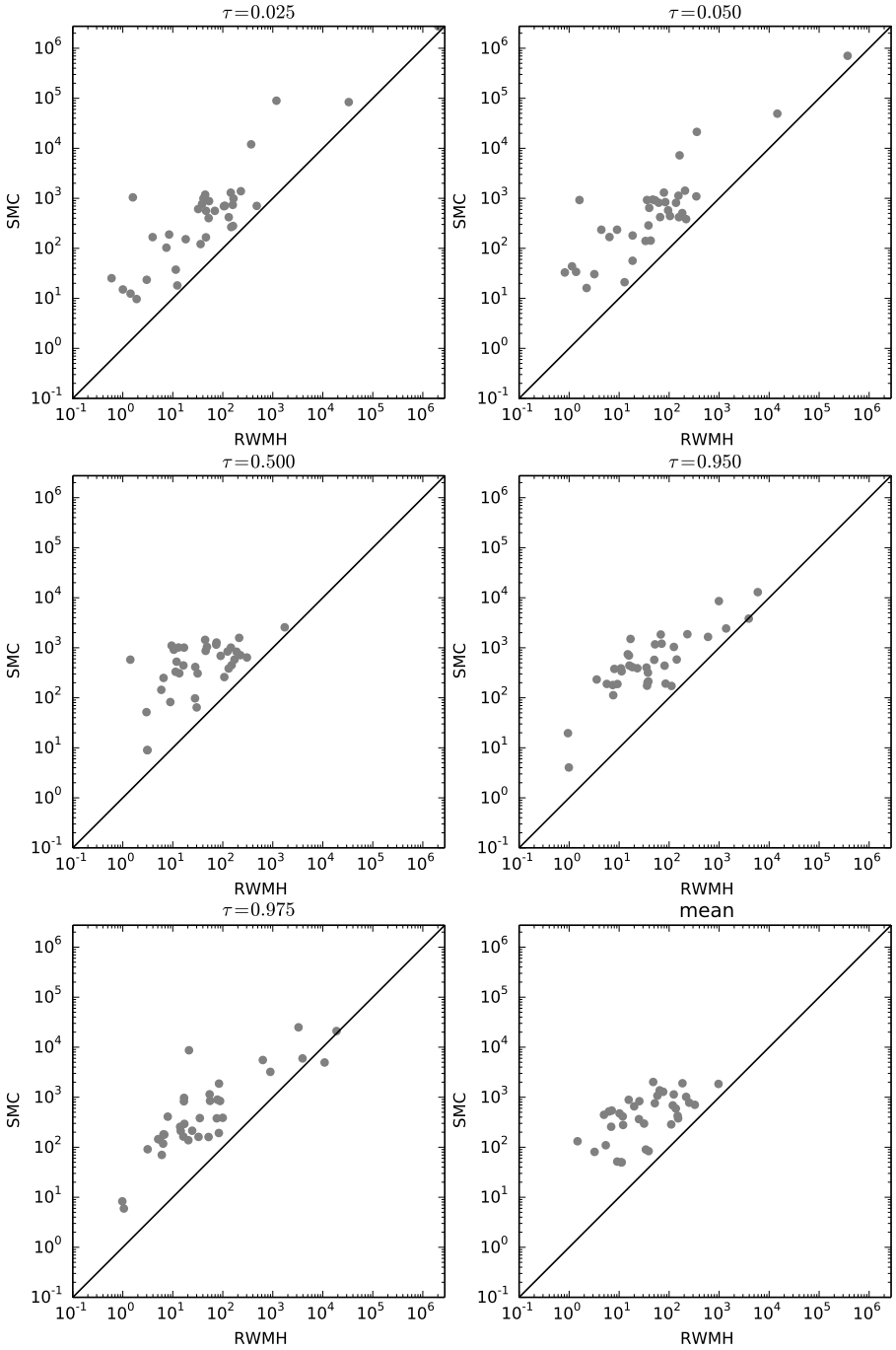
$$N_{eff} = \frac{\tau(1-\tau)/\hat{\pi}^2(\hat{q}_\tau)}{\hat{V}[\hat{q}_\tau]}. \tag{6.6}$$

Thus, N_{eff} measures the number of *iid* draws that one has to generate from the posterior distribution to achieve a quantile approximation that is as accurate as the approximation obtained from the posterior simulators. The SMC approximation of the quantiles are generally more accurate than the MCMC approximations from the RWMH-V algorithm. This difference is most pronounced for $r_{\Delta y}$. For α the measure N_{eff} does not vary much as a function of the quantile τ. For the parameters ξ_w and ι_w the efficiency measure is larger for the 0.95 and 0.975 quantiles, which may be due to the fact that we are using a kernel density estimator that does not account for the upper bound of one for these two parameters.

The precision of the quantile approximations for all of the estimated DSGE model parameters is summarized in Figure 6.7. Each panel corresponds to a particular quantile and the bottom right panel contains results for the posterior mean. Each dot in the scatter plots depicts N_{eff} for the RWMH-V and the SMC approximation of the posterior quantile of a particular parameter. Essentially all dots lie above the 45-degree line, indicating that the SMC algorithm provides more accurate approximations than the RWMH algorithm. For most parameters the gain in accuracy from using the SMC algorithm exceeds a factor of five.

As with the the quantile estimates for the parameters, the SMC estimate of the marginal data density is more accurate. Herbst and Schorfheide (2014) report that the standard deviation of the estimate of the log marginal data density is five times larger under 20 simulations from the RWMH-V than it is under the SMC sampler, owing the poor performance of both the posterior simulator and modified harmonic mean estimators on multi-modal models. As with the simple DSGE model, using more diffuse priors substantially improves model fit. See Herbst and Schorfheide (2014) for details.

Figure 6.7: Precision of Quantile Approximations for SMC and RWMH Algorithms



Notes: N_{eff} for the RWMH-V and SMC quantile approximations. Each dot corresponds to one parameter. The 45 degree line appears in solid.

6.3 Leeper-Plante-Traum Fiscal Policy Model

The final case study is the fiscal DSGE model estimated by Leeper, Plante, and Traum (2010), hereafter LPT. The model is based on a standard real business cycle model with habit, investment adjustment, and variable capital utilization. It also includes a detailed description of fiscal policy. The estimated model is used to track the dynamics of fiscal financing and to assess the role of debt in the determination of spending, taxes, and transfers. Here we present only the portion of the model relevant to the fiscal sector; the full set of loglinearized equations can be found in Section A.2 of the Appendix.

There are three sources of time-varying distortionary taxation (rates) in the model, τ_t^c, τ_t^k, and τ_t^l, levied on consumption, capital and labor income, respectively. Households allocate their income between consumption, c_t (taxed at rate τ_t^c), government bonds, b_t, and capital investment, i_t. Their income is composed of labor income ($w_t l_t$, taxed at rate τ_t^l), utilized-capital income ($R_t^k u_t k_{t-1}$, taxed at rate τ_t^k), riskless government bond income ($R_{t-1} b_{t-1}$), and transfers z_t. Taking the spending and income of households' together, the (flow) budget constraint can be written as:

$$(1+\tau_t^c)c_t + i_t + b_t = (1-\tau_t^l)w_t l_t + (1-\tau_t^k)R_t^k u_t k_{t-1} + R_{t-1}b_{t-1} + z_t. \tag{6.7}$$

The government uses the income from these taxes to finance government spending, G_t. The budget constraint for the government, using capital letters to denote aggregate quantities, is:

$$B_t + \tau_t^k R_t^k u_t K_{t-1} + \tau_t^l w_t L_t + \tau_t^c C_t = R_{t-1}B_{t-1} + G_t + Z_t. \tag{6.8}$$

The level of taxes and transfers are given by fiscal rules, which we describe below. Capital and labor taxes partially depend on output (Y_t), capturing the effects of automatic stabilizers via parameters φ_k and φ_l, and the debt-to-GDP ratio (B_{t-1}) via parameters γ_k and γ_l for capital and labor, respectively. All of rates are driven by exogenous movements to taxes, u_t^c, u_t^k, and u_t^l. In particular, these exogenous movements in one tax category can contemporaneously affect the tax rates in other sectors. The degree of comovement are controlled by the parameters ϕ_{kl}, ϕ_{kc}, and ϕ_{lc}. Letting $\hat{x}_t$ denote the log deviation from steady-state of x_t, one can summarize the tax structure of the model as:

$$\hat{\tau}_t^k = \varphi_k \hat{Y}_t + \gamma_k \hat{B}_{t-1} + \phi_{kl}\hat{u}_t^l + \phi_{kc}\hat{u}_t^c + \hat{u}_t^k, \tag{6.9}$$

$$\hat{\tau}_t^l = \varphi_l \hat{Y}_t + \gamma_l \hat{B}_{t-1} + \phi_{kl}\hat{u}_t^l + \phi_{lc}\hat{u}_t^c + \hat{u}_t^l, \tag{6.10}$$

$$\hat{\tau}_t^c = \phi_{kc}\hat{u}_t^c + \phi_{cl}\hat{u}_t^c + \hat{u}_t^c. \tag{6.11}$$

The exogenous movements in taxes follow AR(1) processes:

$$\hat{u}_t^k = \rho_k \hat{u}_{t-1}^k + \sigma_k \epsilon_t^k, \quad \epsilon_t^k \sim \mathcal{N}(0,1), \tag{6.12}$$

$$\hat{u}_t^l = \rho_l \hat{u}_{t-1}^l + \sigma_l \epsilon_t^l, \quad \epsilon_t^l \sim \mathcal{N}(0,1), \tag{6.13}$$

$$\hat{u}_t^c = \rho_c \hat{u}_{t-1}^c + \sigma_c \epsilon_t^c, \quad \epsilon_t^c \sim \mathcal{N}(0,1). \tag{6.14}$$

On the outlays side, the fiscal rule for government spending, G_t, is a function of current output and the previous period's debt, controlled by the parameters, φ_g and γ_g, respectively. Spending is also affected by an exogenous, $AR(1)$ process, u_t^g. In the log deviations from steady state, the rule is given by:

$$\hat{G}_t = -\varphi_g \hat{Y}_t - \gamma_g \hat{B}_{t-1} + \hat{u}_t^g, \tag{6.15}$$

$$\hat{u}_t^g = \rho_g \hat{u}_{t-1}^g + \sigma_g \epsilon_t^g, \quad \epsilon_t^g \sim \mathcal{N}(0,1). \tag{6.16}$$

The fiscal authority also facilitates lump sum transfers, Z_t, which follows a rule again determined by output (via parameter φ_Z), debt (via parameter γ_z), and an exogenous, $AR(1)$ shock u_t^z. Expressed in log deviations from steady state, the rule is given by:

$$\hat{Z}_t = -\varphi_z \hat{Y}_t - \gamma_z \hat{B}_{t-1} + \hat{u}_t^z, \tag{6.17}$$

$$\hat{u}_t^z = \rho_z \hat{u}_{t-1}^z + \sigma_z \epsilon_t^z, \quad \epsilon_t^z \sim \mathcal{N}(0,1). \tag{6.18}$$

Prior Specifications. LPT use tight (but defensible) priors. Still, there are reasonable arguments for making the prior more diffuse. For example, in their prior, φ_g, the (negative) response of government spending to output, is restricted to be greater than zero, implying counter-cyclical fiscal policy. There is room to relax this. Moreover, many of the tax comovement parameters, for example ϕ_{kc} and ϕ_{lc}, have not be estimated in the past, and so could plausibility be characterized *a priori* with higher uncertainty than in the original estimation. Finally, estimation with diffuse priors allows the analyst to parse the effects of their original priors.

In this spirit, we estimate their model (on slightly different data) with substantially more diffuse priors on a subset of parameters which determine the dynamics of the fiscal sector. The "standard" and "diffuse" prior distributions are displayed in Table 6.5. The γ parameters, which determine the responses of spending, taxes, and transfers to movements (in the previous period's) debt level, are shifted from a gamma distributions centered tightly around 0.4 to uniform distributions on $[0, 5]$. While the γs are still restricted to be nonnegative, consistent with stability of the fiscal rules, there is much more uncertainty about plausible values;

Table 6.5: Fiscal Model: Prior Distributions

	Standard Prior			Diffuse Prior		
	Type	Para (1)	Para (2)	Type	Para (1)	Para (2)
Debt Response Parameters						
γ_g	Gamma	0.4	0.2	Uniform	0	5
γ_{tk}	Gamma	0.4	0.2	Uniform	0	5
γ_{tl}	Gamma	0.4	0.2	Uniform	0	5
γ_z	Gamma	0.4	0.2	Uniform	0	5
Output Response Parameters						
φ_{tk}	Gamma	1.0	0.3	Normal	1.0	1
φ_{tl}	Gamma	0.5	0.25	Normal	0.5	1
φ_g	Gamma	0.07	0.05	Normal	0.07	1
φ_z	Gamma	0.2	0.1	Normal	0.2	1
Exogenous Tax Comovement Parameters						
ϕ_{kl}	Normal	0.25	0.1	Normal	0.25	1
ϕ_{kc}	Normal	0.05	0.1	Normal	0.05	1
ϕ_{lc}	Normal	0.05	0.1	Normal	0.05	1

Notes: Para (1) and Para (2) correspond to the mean and standard deviation of the Beta, Gamma, and Normal distributions and to the upper and lower bounds of the support for the Uniform distribution. For the Inv. Gamma distribution, Para (1) and Para (2) refer to s and ν, where $p(\sigma|\nu, s) \propto \sigma^{-\nu-1}e^{-\nu s^2/2\sigma^2}$.

moreover, the prior is flat, denying the posterior a potential source of curvature. The φ parameters control how much current output affects spending, taxation, and transfers. The parametric assumption of Gamma distributions implies that spending and transfers will be countercyclical, while capital and labor tax rates will be procyclical. These hard restrictions might seem implausible; there is substantial evidence that, for example, government spending is procyclical. Our diffuse priors on these parameter are centered at the same values as their standard counterparts, but with substantially higher variance (between three- and ten-fold increase in prior standard deviation.) The priors on other parameters remain at the values used by LPT, and can be seen in Table 6.6.

Data and Tuning of Algorithm. LPT use US data from 1960Q1 to 2008Q1 on nine

Table 6.6: Fiscal Model: Common Prior Distribution

	Type	Para (1)	Para (2)		Type	Para (1)	Para (2)
Endogenous Propagation Parameters							
γ	Gamma	1.75	0.5	s''	Gamma	5	0.5
κ	Gamma	2.0	0.5	δ_2	Gamma	0.7	0.5
h	Beta	0.5	0.2				
Exogenous Process Parameters							
ρ_a	Beta	0.7	0.2	σ_a	Inv. Gamma	1	4
ρ_b	Beta	0.7	0.2	σ_b	Inv. Gamma	1	4
ρ_l	Beta	0.7	0.2	σ_l	Inv. Gamma	1	4
ρ_i	Beta	0.7	0.2	σ_i	Inv. Gamma	1	4
ρ_g	Beta	0.7	0.2	σ_g	Inv. Gamma	1	4
ρ_{tk}	Beta	0.7	0.2	σ_{tk}	Inv. Gamma	1	4
ρ_{tl}	Beta	0.7	0.2	σ_{tl}	Inv. Gamma	1	4
ρ_{tc}	Beta	0.7	0.2	σ_{tc}	Inv. Gamma	1	4
ρ_z	Beta	0.7	0.2	σ_z	Inv. Gamma	1	4

Notes: Para (1) and Para (2) correspond to the mean and standard deviation of the Beta, Gamma, and Normal distributions and to the upper and lower bounds of the support for the Uniform distribution. For the Inv. Gamma distribution, Para (1) and Para (2) refer to s and ν, where $p(\sigma|\nu, s) \propto \sigma^{-\nu-1}e^{-\nu s^2/2\sigma^2}$.

series to estimate the model: the deviations of log real per capita consumption, investment, hours, government debt, government spending, capital tax revenues, labor tax revenues, consumption tax revenues, and government transfers from independent linear trends. Details on the data construction are available in the Appendix of LPT.

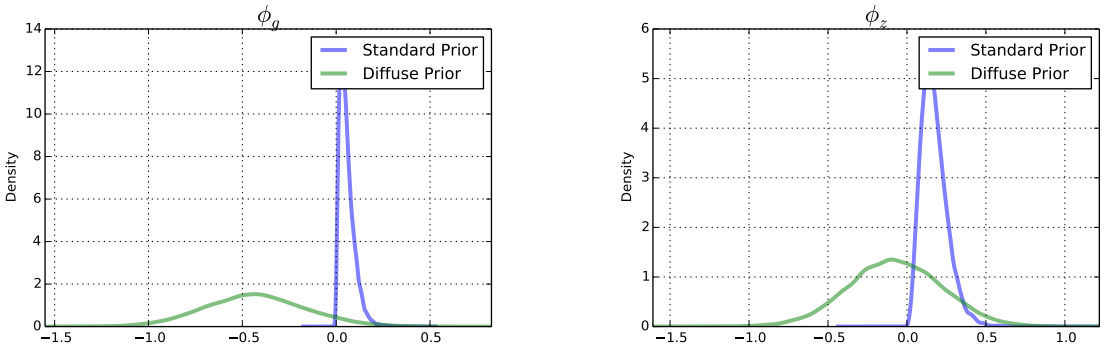
Under the diffuse prior the posterior distribution will be multi-modal. Because we highlighted the difficulty of the RWMH-V algorithm with multimodal posterior surfaces already above, we focus on the substantive results obtained from a single run of the SMC algorithm. The configuration of the algorithm is summarized in Table 6.7.

Results. Table 6.8 summarizes the posterior distribution of the key parameters related

Table 6.7: Configuration of Algorithm for LPT Model

SMC
$N = 6000$
$N_\phi = 500$
$N_{blocks} = 3$, $M = 1$
$\lambda = 4.0$
Run time: 48 minutes (12 cores)

Figure 6.8: Posterior Densities of Output Response Parameters φ_g and φ_z



Notes:

to fiscal policy in the model.[2] The posterior means for the debt-response parameters are more-or-less that same across prior settings, indicating that the prior is not substantially influencing the posterior. On the other hand, the posterior distrubition of the elasticities of taxes, spending, and transfers with respect to output (the φ parameters) are substantially different under the different priors. In particular, the restriction that $\varphi_i > 0$ for $i \in \{tk, tl, g, z\}$, embodied in the LPT prior, is "binding" in the sense that the posterior under the diffuse prior has substantial density for φ_{tl}, φ_g, and $\varphi_z < 0$. Indeed, as shown in Figure 6.8, once this restriction is relaxed, the sign of the posterior for φ_g and φ_z switches.

Figure 6.9 depicts scatter plots of draws from bivariate posterior distributions in the off-diagonal panels and density plots for univariate posteriors of the tax comovement parameters ϕ_{lc}, ϕ_{kc}, and ϕ_{kl}. Under the LPT prior distribution the posteriors appear to be unimodal

[2]The posterior distribution of the other parameters is similar under the two priors and can be seen in Table A-3 in the Appendix.

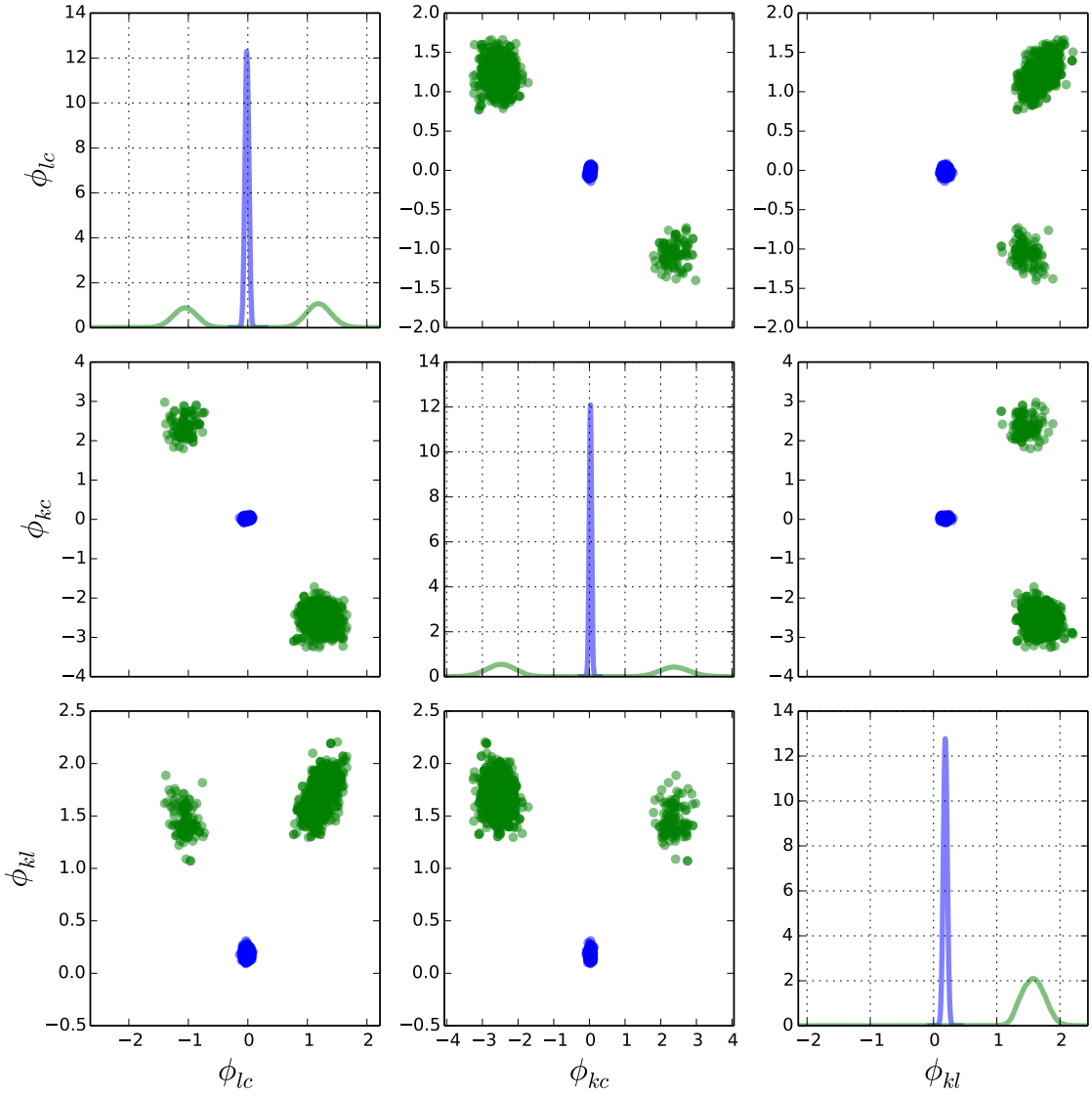
Table 6.8: Fiscal Model: Posterior Moments - Part 1

	Based on LPT Prior		Based on Diff. Prior	
	Mean	[5%, 95%] Int.	Mean	[5%, 95%] Int.
Debt Response Parameters				
γ_g	0.16	[0.07, 0.27]	0.1	[0.01, 0.23]
γ_{tk}	0.39	[0.22, 0.60]	0.38	[0.16, 0.62]
γ_{tl}	0.11	[0.04, 0.21]	0.039	[0.00, 0.11]
γ_z	0.32	[0.17, 0.47]	0.32	[0.14, 0.49]
Output Response Parameters				
φ_{tk}	1.7	[1.18, 2.18]	2.1	[1.44, 2.69]
φ_{tl}	0.29	[0.11, 0.53]	0.12	[-0.33, 0.58]
φ_g	0.057	[0.01, 0.13]	-0.43	[-0.87, 0.03]
φ_z	0.17	[0.06, 0.33]	-0.074	[-0.56, 0.41]
Exogenous Tax Comovement Parameters				
ϕ_{kl}	0.19	[0.14, 0.24]	1.6	[1.29, 1.87]
ϕ_{kc}	0.028	[-0.03, 0.08]	-0.33	[-2.84, 2.73]
ϕ_{lc}	-0.016	[-0.07, 0.04]	0.2	[-1.23, 1.40]
Innovations to Fiscal Rules				
σ_g	3.0	[2.79, 3.30]	2.9	[2.66, 3.18]
σ_{tk}	4.4	[4.01, 4.75]	1.3	[1.08, 1.46]
σ_{tl}	3.0	[2.71, 3.22]	2.0	[1.71, 2.33]
σ_{tc}	4.0	[3.67, 4.33]	1.1	[0.96, 1.35]
σ_z	3.3	[3.07, 3.63]	3.3	[3.07, 3.63]

and concentrated near zero. As the prior distribution is relaxed, the posterior distributions become multimodal. The marginal posterior of ϕ_{lc} has modes near -1 and 1, and the posterior of ϕ_{kc} has modes near -3 and 3. The posterior distributions are not symmetric. For ϕ_{lc} there is more mass in the positive region of the parameter space whereas for ϕ_{kc} most of the posterior mass is in the negative region of the parameter space.

The multimodal posterior for the parameters translates into a multimodal posterior for impulse responses. The responses to a labor income tax shock are depicted in Figure 6.10.

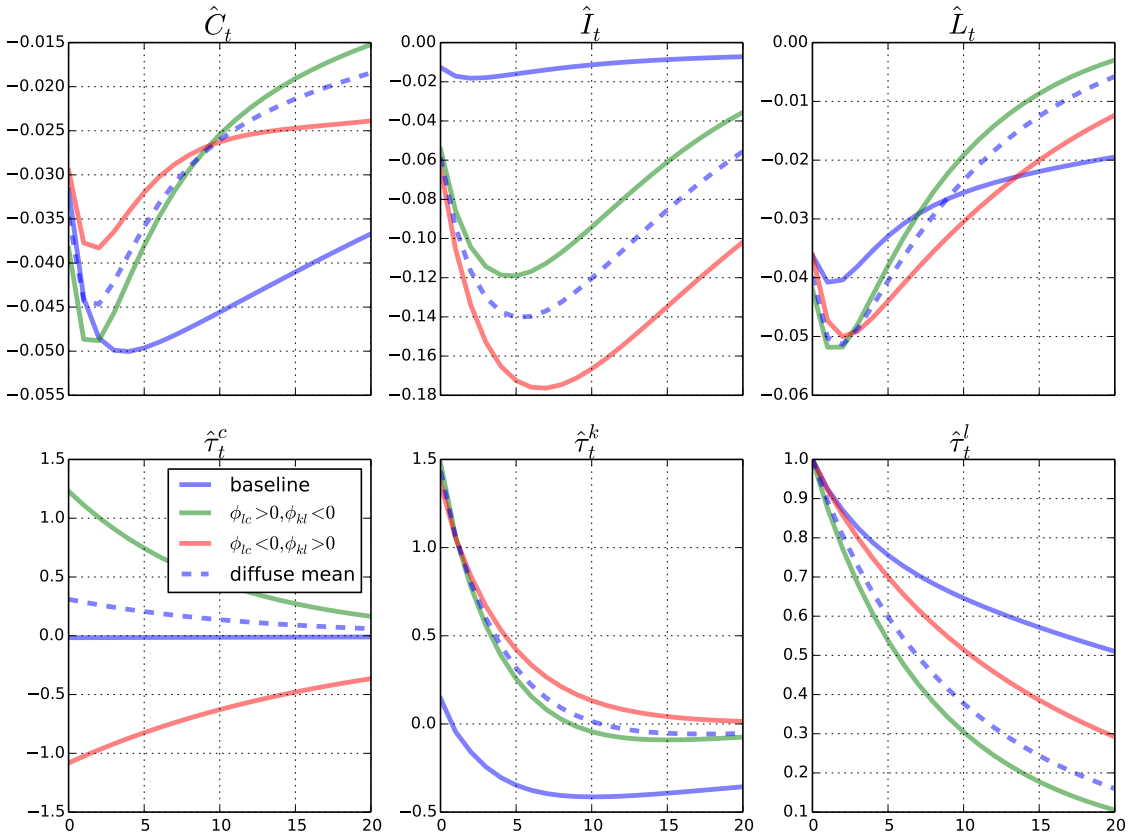
Figure 6.9: Posterior Distribution of Tax Comovement Parameters



Notes: Blue is based on LPT prior; green is based on diffuse prior.

The figure depicts four type of posterior mean responses: the baseline responses obtained from the posterior distribution that is associated with the LPT prior; the unconditional posterior mean responses associated with the diffuse prior; posterior mean responses based on the diffuse prior that condition on $\phi_{lc} > 0, \phi_{kc} < 0$ or $\phi_{lc} < 0, \phi_{kc} > 0$, respectively. The lower right panel shows the response of the labor tax rate $\hat{\tau}_l$. To facilitate the comparison between the four sets of impulse responses, we normalize the labor tax innovation to one percent. If the steady state labor tax rate is 30% then a one percent increase raises the tax

Figure 6.10: Impulse Response to a Labor Tax Innovation



Notes:

rate to 30.3%.

Under the diffuse prior distribution capital taxes increase in response to a labor tax shock, because τ_{kl} is unambiguously positive. Under the LPT prior the capital tax response is more muted and turns negative after one period. While the spillover from the labor tax innovation onto the consumption tax rate is roughly zero on average, under the diffuse prior the response is bimodal: conditional on the $\phi_{lc} > 0$ ($\phi_{lc} < 0$) there is a 1.2% rise (fall) in the consumption tax. In general, the increase in the labor tax lowers the labor supply and the hours worked response is quite similar for all four cases. The increase in capital taxes lowers investment conditional on the diffuse prior distribution. The drop in investment is amplified (dampened) if the consumption tax falls (rises) in response to the labor tax innovation, which creates a bimodal investment response. Falling (rising) consumption taxes create an incentive to allocate more (less) income to consumption and less (more) to investment. In

turn, the consumption response is also bimodal.

Part III

Bayesian Computations for Nonlinear DSGE Models

Chapter 7

Particle Filters

Particle filters are used to compute likelihood functions and track the hidden states in nonlinear state-space models. Nonlinear state-space representations arise when DSGE models are solved by higher-order perturbation methods or projection methods. For detailed descriptions and assessments of nonlinear solution techniques we refer the reader to the existing literature, e.g., Judd (1998) and Aruoba, Fernández-Villaverde, and Rubio-Ramírez (2006). Our starting point is a state-space representation of the form

$$\begin{aligned} y_t &= \Psi(s_t;\theta) + u_t, \quad u_t \sim F_u(\cdot;\theta) \\ s_t &= \Phi(s_{t-1},\epsilon_t;\theta), \quad \epsilon_t \sim F_\epsilon(\cdot;\theta). \end{aligned} \tag{7.1}$$

Just as in the linear case, the functions $\Psi(s_t;\theta)$ and $\Phi(s_{t-1},\epsilon_t;\theta)$ are generated numerically by the solution method. We require that the measurement error u_t in the measurement equation is additively separable and that the probability density function $p(u_t|\theta)$ can be evaluated analytically. In many applications, $u_t \sim N(0,\Sigma_u)$. While the exposition of the algorithms in this chapter focuses on the linear state-space model (7.1), the numerical illustrations and empirical applications are based on the linear Gaussian model

$$\begin{aligned} y_t &= \Psi_0(\theta) + \Psi_1(\theta)t + \Psi_2(\theta)s_t + u_t, \quad u_t \sim N(0,\Sigma_u), \\ s_t &= \Phi_1(\theta)s_{t-1} + \Phi_\epsilon(\theta)\epsilon_t, \quad \epsilon_t \sim N(0,\Sigma_\epsilon) \end{aligned} \tag{7.2}$$

obtained by solving log-linearized DSGE models. For model (7.2) the Kalman filter described in Table 2.1 delivers the exact distributions $p(y_t|Y_{1:t-1},\theta)$ and $p(s_t|Y_{1:t},\theta)$ against which the accuracy of the particle filter approximation can be evaluated.

There exists a large literature on particle filters. Surveys and tutorials can be found, for instance, in Arulampalam, Maskell, Gordon, and Clapp (2002), Cappé, Godsill, and Moulines

(2007), Doucet and Johansen (2011), Creal (2012). These papers provide detailed references to the literature. The basic bootstrap particle filtering algorithm is remarkably straightforward, but may perform quite poorly in practice. Thus, much of the literature focuses on refinements of the bootstrap filter that increases the efficiency of the algorithm, see, for instance, Doucet, de Freitas, and Gordon (2001). Textbook treatments of the statistical theory underlying particle filters can be found in Cappé, Moulines, and Ryden (2005), Liu (2001), and Del Moral (2013). In the remainder of this chapter, we discuss various versions of the particle filter, including the bootstrap filter, conditionally optimal filters, filters for conditional linear models, and the auxiliary particle filter. These filters are then applied to the small-scale New Keynesian DSGE model and the SW model with diffuse prior. Throughout this chapter we condition on a fixed vector of parameter values θ.

7.1 The Bootstrap Particle Filter

We will begin with a version of the particle filter in which the particles representing the hidden state vector s_t are propagated by iterating the state-transition equation in (7.1) forward. This version of the particle filter is due to Gordon, Salmond, and Smith (1993) and called the bootstrap particle filter. As in Algorithm 8, we use the sequence $\{\rho_t\}_{t=1}^T$ to indicate whether the particles are resampled in period t.

Algorithm 11 (Bootstrap Particle Filter)

1. **Initialization.** *Draw the initial particles from the distribution* $s_0^j \overset{iid}{\sim} p(s_0)$ *and set* $W_0^j = 1$, $j = 1, \ldots, M$.

2. **Recursion.** *For* $t = 1, \ldots, T$*:*

   (a) **Forecasting** s_t**.** *Propagate the period* $t-1$ *particles* $\{s_{t-1}^j, W_{t-1}^j\}$ *by iterating the state-transition equation forward:*

   $$\tilde{s}_t^j = \Phi(s_{t-1}^j, \epsilon_t^j; \theta), \quad \epsilon_t^j \sim F_\epsilon(\cdot; \theta). \tag{7.3}$$

   An approximation of $\mathbb{E}[h(s_t)|Y_{1:t-1}, \theta]$ *is given by*

   $$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^j) W_{t-1}^j. \tag{7.4}$$

(b) **Forecasting** y_t**.** *Define the incremental weights*

$$\tilde{w}_t^j = p(y_t|\tilde{s}_t^j, \theta). \tag{7.5}$$

The predictive density $p(y_t|Y_{1:t-1}, \theta)$ *can be approximated by*

$$\hat{p}(y_t|Y_{1:t-1}, \theta) = \frac{1}{M}\sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \tag{7.6}$$

(c) **Updating.** *Define the normalized weights*

$$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j}. \tag{7.7}$$

An approximation of $\mathbb{E}[h(s_t)|Y_{1:t}, \theta]$ *is given by*

$$\tilde{h}_{t,M} = \frac{1}{M}\sum_{j=1}^{M} h(\tilde{s}_t^j)\tilde{W}_t^j. \tag{7.8}$$

(d) **Selection.** Case (i): *If* $\rho_t = 1$ *resample the particles via multinomial resampling. Let* $\{s_t^j\}_{j=1}^M$ *denote* M *iid draws from a multinomial distribution characterized by support points and weights* $\{\tilde{s}_t^j, \tilde{W}_t^j\}$ *and set* $W_t^j = 1$.
Case (ii): *If* $\rho_t = 0$*, let* $s_t^j = \tilde{s}_t^j$ *and* $W_t^j = \tilde{W}_t^j$.
An approximation of $\mathbb{E}[h(s_t)|Y_{1:t}, \theta]$ *is given by*

$$\bar{h}_{t,M} = \frac{1}{M}\sum_{j=1}^{M} h(s_t^j)W_t^j. \tag{7.9}$$

3. **Likelihood Approximation.** *The approximation of the log likelihood function is given by*

$$\ln \hat{p}(Y_{1:T}|\theta) = \sum_{t=1}^{T} \ln\left(\frac{1}{M}\sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j\right). \tag{7.10}$$

The particle filter algorithm closely follows the steps of the generic filter in Algorithm 1. The convergence theory underlying the particle filter is similar to the theory sketched in Section 5.1.4 for the SMC sampler. To simplify the notation, we will drop the parameter vector θ from the conditioning set. Starting point is a SLLN and a CLT for period $t-1$:

$$\begin{aligned} \bar{h}_{t-1,M} &\xrightarrow{a.s.} \mathbb{E}[h(s_{t-1})|Y_{1:t-1}], \\ \sqrt{M}\big(\bar{h}_{t-1,M} - \mathbb{E}[h(s_{t-1})|Y_{1:t-1}]\big) &\Longrightarrow N\big(0, \Omega_{t-1}(h)\big). \end{aligned} \tag{7.11}$$

For period $t-1=0$ this can generally be achieved by directly sampling from the initial distribution $p(s_0)$. We briefly sketch the convergence arguments for Steps 2(a) to 2(d). A rigorous proof would involve verifying the existence of moments required by the SLLN and CLT and a careful characterization of the asymptotic covariance matrices.

Forecasting Steps. The forward iteration of the state-transition equation amounts to drawing s_t from a conditional density $g_t(s_t|s_{t-1}^j)$. In Algorithm 11 this density is given by

$$g_t(s_t|s_{t-1}^j) = p(s_t|s_{t-1}^j).$$

We denote expectations under this density as $\mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[h]$ and decompose

$$\begin{aligned}\hat{h}_{t|t-1} - \mathbb{E}[h(s_t)|Y_{1:t-1}] &= \frac{1}{M}\sum_{j=1}^{M}\Big(h(\tilde{s}_t^j) - \mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[h]\Big)W_{t-1}^j \\ &\quad + \frac{1}{M}\sum_{j=1}^{M}\Big(\mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[h]W_{t-1}^j - \mathbb{E}[h(s_t)|Y_{1:t-1}]\Big) \\ &= I + II,\end{aligned} \tag{7.12}$$

say. This decomposition is similar to the decomposition (5.30) used in the analysis of the mutation step of the SMC algorithm.

Conditional on $\{s_{t-1}^j, W_{t-1}^j\}_{i=1}^N$ the weights W_{t-1}^j are known and the summands in term I form a triangular array of mean-zero random variables that within each row are independently but not identically distributed. Provided the required moment bounds for $h(\tilde{s}_t^j)W_{t-1}^j$ are satisfied, I converges to zero almost surely and satisfies a CLT. Term II also converges to zero because

$$\begin{aligned}\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[h]W_{t-1}^j &\xrightarrow{a.s.} \mathbb{E}\big[\mathbb{E}_{g_t(\cdot|s_{t-1})}[h]\,\big|\,Y_{1:t-1}\big] \\ &= \int\left[\int h(s_t)p(s_t|s_{t-1})ds_t\right]p(s_{t-1}|Y_{1:t-1})ds_{t-1} \\ &= \mathbb{E}[h(s_t)|Y_{1:t-1}]\end{aligned} \tag{7.13}$$

Thus, under suitable regularity conditions

$$\hat{h}_{t,M} \xrightarrow{a.s.} \mathbb{E}[h(s_t)|Y_{1:t-1}], \quad \sqrt{M}\big(\hat{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t-1}]\big) \Longrightarrow N\big(0, \hat{\Omega}_t(h)\big). \tag{7.14}$$

The convergence of the predictive density approximation $\hat{p}(y_t|Y_{1:t-1})$ to $p(y_t|Y_{1:t-1})$ in Step 2(b) follows directly from (7.14) by setting $h(s_t) = p(y_t|s_t)$. For the Gaussian state-space

model (7.2) the incremental weights take the form

$$\tilde{w}_t^j = p(y_t|\tilde{s}_t^j) = (2\pi)^{-n/2}|\Sigma_u|^{-1/2}\exp\Big\{-\frac{1}{2}(y_t-\Psi_0-\Psi_1 t-\Psi_2\tilde{s}_t^j)'\Sigma_u^{-1} \times(y_t-\Psi_0-\Psi_1 t-\Psi_2\tilde{s}_t^j)\Big\}, \tag{7.15}$$

where n here denotes the dimension of y_t.

Updating and Selection Steps. The goal of the updating step is to approximate posterior expectations of the form

$$\mathbb{E}[h(s_t)|Y_{1:t}] = \frac{\int h(s_t)p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t}{\int p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t} \approx \frac{\frac{1}{M}\sum_{j=1}^M h(\tilde{s}_t^j)\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j} = \tilde{h}_{t,M}. \tag{7.16}$$

The Monte Carlo approximation of $\mathbb{E}[h(s_t)|Y_{1:t}]$ has the same form as the Monte Carlo approximation of $\tilde{h}_{n,M}$ in (5.23) in the correction step of the SMC Algorithm 8 and its convergence can be analyzed in a similar manner. Defining the normalized incremental weights

$$v_t(s_t) = \frac{p(y_t|s_t)}{\int p(y_t|s_t)p(s_t|Y_{1:t})ds_t}, \tag{7.17}$$

under suitable regularity conditions the Monte Carlo approximation satisfies a CLT of the form

$$\sqrt{M}\big(\tilde{h}_{t,M}-\mathbb{E}[h(s_t)|Y_{1:t}]\big) \Longrightarrow N\big(0,\tilde{\Omega}_t(h)\big), \quad \tilde{\Omega}_t(h) = \hat{\Omega}_t\big(v_t(s_t)(h(s_t)-\mathbb{E}[h(s_t)|Y_{1:t}])\big). \tag{7.18}$$

Finally, the selection step in Algorithm 11 is identical to the selection step in Algorithm 8 and it adds some additional noise to the approximation. If $\rho_t = 1$, then

$$\sqrt{M}\big(\bar{h}_{t,M}-\mathbb{E}[h(s_t)|Y_{1:t}]\big) \Longrightarrow N\big(0,\Omega_t(h)\big), \quad \Omega_t(h) = \tilde{\Omega}_t(h) + \mathbb{V}[h(s_t)|Y_{1:t}]. \tag{7.19}$$

As discussed in the context of the SMC algorithm, the asymptotic covariance matrix $\tilde{\Omega}_t(h)$ critically depends on the distribution of the particle weights. As for the SMC algorithm, we can define an effective sample size (in terms of number of particles) as

$$\widehat{ESS}_t = N\Big/\left(\frac{1}{M}\sum_{j=1}^M(\tilde{W}_t^j)^2\right). \tag{7.20}$$

and replace the deterministic sequence $\{\rho_t\}_{t=1}^T$ by an adaptively chosen sequence $\{\hat{\rho}_t\}_{t=1}^T$, for which $\hat{\rho}_t = 1$ whenever $\widehat{ESS}_t$ falls below a threshold.

The Role of Measurement Errors. Many DSGE models, e.g., the ones considered in this book, do not assume that the observables y_t are measured with error. Instead, the number

of structural shocks is chosen to be equal to the number of observables, which means that the likelihood function $p(Y_{1:T})$ is nondegenerate. The Kalman filter iterations in Table 2.1 are well defined even if the measurement error covariance matrix Σ_u in the linear Gaussian state space model (7.2) is equal to zero, provided that the number of shocks ϵ_t is not smaller than the number of observables and the forecast error covariance matrix $F_{t|t-1}$ is invertible.

For the particle filter, the incremental weights (7.15) are degenerate if $\Sigma_u = 0$ because the conditional distribution of $y_t|s_t$ is a pointmass. For a particle j, this point mass is located at $y_t^j = \Psi(\tilde{s}_t^j; \theta)$. If in the forecasting step the innovation ϵ_t^j is drawn from a continuous distribution and the state transition equation $\Phi(s_{t-1}, \epsilon_t; \theta)$ is a smooth function of the lagged state and the innovation ϵ_t, then the probability that $y_t^j = y_t$ is zero, which means that $\tilde{w}_t^j = 0$ for all j and the particles vanish after one iteration. The intuition for this result is straightforward. The incremental weights are large for particles j for which $y_t^j = \Psi(\tilde{s}_t^j; \theta)$ is close to the actual y_t. Under Gaussian measurement errors, the metric for closeness is given by Σ_u^{-1}. Thus, all else equal, decreasing the measurement error variance Σ_u increases the discrepancy between y_t^j and y_t and therefore the variance of the particle weights.

Consider the following stylized examples (we are omitting the j superscripts). Suppose that y_t is scalar, the measurement errors are distributed according to $u_t \sim N(0, \sigma_u^2)$, $W_{t-1} = 1$, and let $\delta = y_t - \Psi(s_t; \theta)$. Suppose that in population the δ is distributed according to a $N(0,1)$. In this case $v_t(s_t)$ in (7.17) can be viewed as a population approximation of the normalized weights $\tilde{W}_t$ constructed in the updating step (note that the denominator of these two objects is slightly different):

$$\tilde{W}_t(\delta) \approx v_t(\delta) = \frac{\exp\left\{-\frac{1}{2\sigma_u^2}\delta^2\right\}}{(2\pi)^{-1/2}\int \exp\left\{-\frac{1}{2}\left(1+\frac{1}{\sigma_u^2}\right)\delta^2\right\} d\delta} = \left(1+\frac{1}{\sigma_u^2}\right)^{1/2} \exp\left\{-\frac{1}{2\sigma_u^2}\delta^2\right\}.$$

The asymptotic covariance matrix $\tilde{\Omega}_t(h)$ in (7.18) which captures the accuracy of $\tilde{h}_{t,M}$ as well as the heuristic effective sample size measure defined in (7.20) depend on the variance of the particle weights, which in population is given by

$$\int v_t^2(\delta) d\delta = \frac{1+1/\sigma_u^2}{\sqrt{1+2/\sigma_u^2}} = \frac{1}{\sigma_u}\frac{1+\sigma_u^2}{\sqrt{2+\sigma_u^2}} \longrightarrow \infty \quad \text{as} \quad \sigma_u \longrightarrow 0.$$

Thus, a decrease in the measurement error variance raises the variance of the particle weights and thereby decreases the effective sample size. More importantly, the increasing dispersion of the weights translates into an increase in the limit covariance matrix $\tilde{\Omega}_t(h)$ and a deterioration of the Monte Carlo approximations generated by the particle filter. In sum, all else equal, the smaller the measurement error variance, the less accurate the particle filter.

7.2 Sequential Importance Sampling and Resampling

In the basic version of the particle filter the time t particles were generated by simulating the state transition equation forward. However, the naive forward simulation ignores information contained in the current observation y_t and may lead to a very uneven distribution of particle weights, in particular if the measurement error variance is small or if the model has difficulties explaining the period t observation in the sense that for most particles $\tilde{s}_t^j$ the actual observation y_t lies far in the tails of the model-implied distribution of $y_t|(\tilde{s}_t^j, \theta)$. The particle filter can be generalized by allowing $\tilde{s}_t^j$ in the forecasting step to be drawn from a generic importance sampling density $g_t(\cdot|s_{t-1}^j, \theta)$, which is why particle filters are also called sequential importance sampling with resampling (SISR) algorithms.

Algorithm 12 (Sequential Importance Sampling with Resampling)

1. **Initialization.** *(Same as Algorithm 11)*

2. **Recursion.** *For $t = 1, \ldots, T$:*

   (a) **Forecasting s_t.** *Draw $\tilde{s}_t^j$ from density $g_t(\tilde{s}_t|s_{t-1}^j, \theta)$ and define the importance weights*

   $$\omega_t^j = \frac{p(\tilde{s}_t^j|s_{t-1}^j, \theta)}{g_t(\tilde{s}_t^j|s_{t-1}^j, \theta)}. \tag{7.21}$$

   An approximation of $\mathbb{E}[h(s_t)|Y_{1:t-1}, \theta]$ is given by

   $$\hat{h}_{t,M} = \frac{1}{M}\sum_{j=1}^{M} h(\tilde{s}_t^j)\omega_t^j W_{t-1}^j. \tag{7.22}$$

   (b) **Forecasting y_t.** *Define the incremental weights*

   $$\tilde{w}_t^j = p(y_t|\tilde{s}_t^j, \theta)\omega_t^j. \tag{7.23}$$

   The predictive density $p(y_t|Y_{1:t-1}, \theta)$ can be approximated by

   $$\hat{p}(y_t|Y_{1:t-1}, \theta) = \frac{1}{M}\sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \tag{7.24}$$

   (c) **Updating.** *(Same as Algorithm 11)*

   (d) **Selection.** *(Same as Algorithm 11)*

3. **Likelihood Approximation.** *(Same as Algorithm 11).*

The only difference between Algorithms 11 and 12 is the introduction of the importance weights ω_t^j which appear in (7.22) as well as the definition of the incremental weights $\tilde{w}_t^j$ in (7.23). It can be verified that the introduction of the importance weights guarantees the the Monte Carlo averages converge to the desired limits. To assess the convergence of $\hat{h}_{t,M}$ replace in the decomposition (7.12) $h(\tilde{s}_t^j)$ by $h(\tilde{s}_t^j)\omega_t^j$ and $\mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[h]$ by $\mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[h\omega]$, respectively. Then note that

$$\begin{aligned}\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[h\omega]W_{t-1}^j &\xrightarrow{a.s.} \mathbb{E}\big[\mathbb{E}_{g_t(\cdot|s_{t-1})}[h\omega]\,\big|\,Y_{1:t-1}\big] \\ &= \int\left[\int h(s_t)\frac{p(s_t|s_{t-1})}{g_t(s_t|s_{t-1})}g_t(s_t|s_{t-1})ds_t\right]p(s_{t-1}|Y_{1:t-1})ds_{t-1} \\ &= \mathbb{E}[h(s_t)|Y_{1:t-1}],\end{aligned} \tag{7.25}$$

as desired.

The main goal of replacing the forward iteration of the state-transition equation by an importance sampling step is to improve the accuracy of $\hat{p}(y_t|Y_{1:t-1},\theta)$ and $\tilde{h}_{t,M}$. Consider

$$\frac{1}{M}\sum_{j=1}^{M}h(\tilde{s}_t^j)\tilde{w}_t^jW_{t-1}^j = \frac{1}{M}\sum_{j=1}^{M}h(\tilde{s}_t^j)\frac{p(y_t|\tilde{s}_t^j)p(\tilde{s}_t^j|s_{t-1}^j)}{g_t(\tilde{s}_t^j|s_{t-1}^j)}W_{t-1}^j, \tag{7.26}$$

which for $h(\cdot)=1$ delivers $\hat{p}(y_t|Y_{1:t-1},\theta)$ and also appears in the definition of $\tilde{h}_{t,M}$. Choosing a suitable importance density $g_t(\tilde{s}_t^j|s_{t-1}^j)$ that is a function of the time t observation y_t can drastically reduce the variance of the incremental weights $\tilde{w}_t^j$ and improve the accuracy of the Monte Carlo approximations. Such filters are called adapted particle filters. We will subsequently discuss various choices of $g_t(\tilde{s}_t^j|s_{t-1}^j)$.

7.3 Implementation Issues

The implementation of Algorithm 12 requires the evaluation of the density $p(s_t|s_{t-1}^j)$. In a nonlinear DSGE model the innovations ϵ_t typically enter the state transition equation $s_t = \Phi(s_{t-1}^j,\epsilon_t)$ in a non-additive form, which makes it difficult to compute $p(s_t|s_{t-1}^j)$. We show in Section 7.3.1 that if the proposal distribution $g_t(s_t|s_{t-1}^j)$ is implicitly generated by iterating the state-transition equation forward based on a draw $\tilde{\epsilon}_t^j$ from $g_t^\epsilon(s_{t-1}^j)$ then the computation of the importance weights ω_t^j simplifies considerably and does not require the evaluation of conditional densities of s_t. Section 7.3.2 provides further discussion of filtering for models that do not have measurement errors.

7.3.1 Nonlinear and Partially Deterministic State Transitions

The implementation of Algorithm 12 requires the evaluation of the density $p(s_t|s_{t-1})$. Two difficulties arise in nonlinear DSGE model applications: first, the density is singular because some state variables, e.g. the capital stock, may evolve according to a deterministic law of motion. Second, if the state-transition equation is nonlinear and the innovations do not enter in an additively separable way, it may be difficult to evaluate the density $p(s_t|s_{t-1}, \theta)$ because of a complicated change of variables. For illustrative purposes, consider a modified version of the simple state space model of Section 4.3 with state transition equations:

$$s_{1,t} = \Phi_1(s_{t-1}, \epsilon_t), \quad s_{2,t} = \Phi_2(s_{t-1}), \quad \epsilon_t \sim N(0,1). \tag{7.27}$$

Here the transition for the state $s_{2,t}$ (think of the capital accumulation equation in a DSGE model) is deterministic. Thus the joint distribution of $s_{1,t}$ and $s_{2,t}$ is a mixture of a continuous and a discrete distribution with a pointmass at $s_{2,t} = \Phi_2(s_{t-1})$.

Now suppose we define the extended state vector $\varsigma_t = [s_t', \epsilon_t']'$ and augment the state transitions in (7.27) by the identity $\epsilon_t = \epsilon_t$. Using the independence of the innovation ϵ_t from the lagged states ς_{t-1}, we can factorize the density $p(\varsigma_t|\varsigma_{t-1})$ as

$$p(\varsigma_t|\varsigma_{t-1}) = p^\epsilon(\epsilon_t)p(s_{1,t}|s_{t-1}, \epsilon_t)p(s_{2,t}|s_{t-1}). \tag{7.28}$$

Note that $p(s_{1,t}|s_{t-1}, \epsilon_t)$ and $p(s_{2,t}|s_{t-1})$ are pointmasses at $s_{1,t} = \Phi_1(s_{t-1}, \epsilon_t)$ and $s_{2,t} = \Phi_2(s_{t-1})$, respectively. The easiest way of designing an importance distribution $g_t(\varsigma_t|\varsigma_{t-1})$ that has support in the subspace of the state space that satisfies (7.27) is to sample an innovation ϵ_t and iterate the state-transition equation forward. Let $g_t^\epsilon(\epsilon_t|s_{t-1})$ denote the importance density for ϵ_t. Then

$$g_t(\varsigma_t|\varsigma_{t-1}) = g_t^\epsilon(\epsilon_t|s_{t-1})p(s_{1,t}|s_{t-1}, \epsilon_t)p(s_{2,t}|s_{t-1}). \tag{7.29}$$

In turn,

$$\omega_t^j = \frac{p(\tilde{\varsigma}_t^j|\varsigma_{t-1}^j)}{g_t(\tilde{\varsigma}_t^j|\varsigma_{t-1}^j)} = \frac{p^\epsilon(\tilde{\epsilon}_t^j)p(\tilde{s}_{1,t}^j|s_{t-1}^j, \tilde{\epsilon}_t^j)p(\tilde{s}_{2,t}^j|s_{t-1}^j)}{g_t^\epsilon(\tilde{\epsilon}_t^j|s_{t-1}^j)p(\tilde{s}_{1,t}^j|s_{t-1}^j, \tilde{\epsilon}_t^j)p(\tilde{s}_{2,t}^j|s_{t-1}^j)} = \frac{p^\epsilon(\tilde{\epsilon}_t^j)}{g_t^\epsilon(\tilde{\epsilon}_t^j|s_{t-1}^j)}. \tag{7.30}$$

Thus, the computation of ω_t^j only requires the evaluation of the densities for ϵ_t.

The derivation of (7.30) may appear a bit obscure because it involves the factorization of a joint density for a degenerate probability distribution. The reader may wonder why the Jacobian term that would arise under a careful change-of-variables argument does not

appear in (7.30). Notice that we are ultimately using (7.30) in an importance sampling approximation of an integral. The key insight (simplifying the notation considerably) is that if $s = \Phi(\epsilon)$ then

$$\int h(s)p^s(s)ds = \int h(\Phi(\epsilon))p^\epsilon(\epsilon)d\epsilon. \tag{7.31}$$

According to the change-of-variable formula the relationship between the densities $p^s(\cdot)$ and $p^\epsilon(\cdot)$ is the following. Define B_j, $j = 1, \ldots, J$, as a set of partitions of the domain of ϵ such that $\Phi(s)$ is monotone in ϵ. Let $\Phi_j^{-1}(s)$ be the inverse with respect to ϵ on B_j and assume it has continuous derivatives. Then

$$p^s(s) = \sum_{j=1}^{J} p^\epsilon\big(\Phi_j^{-1}(s)\big) \left| \frac{\partial}{\partial s}\Phi_j^{-1}(s) \right|$$

and we can write

$$\int h(s)p^s(s)ds = \sum_{j=1}^{J} \int h(s)p^\epsilon\big(\Phi_j^{-1}(s)\big) \left| \frac{\partial}{\partial s}\Phi_j^{-1}(s) \right|.$$

Setting $\epsilon_j = \Phi_j^{-1}(s)$, $s = \Phi(\epsilon_j)$, and noting that

$$\int h(\Phi(\epsilon))p^\epsilon(\epsilon)d\epsilon = \sum_{j=1}^{J} \int_{A_j} h\big(\Phi(\epsilon_j)\big)p^\epsilon(\epsilon_j)d\epsilon_j$$

yields the desired result in (7.31). We can now change the forecasting step 2.(a) of Algorithm 12 as follows:

Algorithm 13 (Generalized Bootstrap Particle Filter)

Replace Step 2.(a) in Algorithm 12 by:

2.(a)' **Forecasting** s_t**.** *Draw* $\tilde{\epsilon}_t^j$ *from density* $g_t^\epsilon(\tilde{\epsilon}_t|s_{t-1})$*, let* $\tilde{s}_t^j = \Phi(s_{t-1}, \tilde{\epsilon}_{t-1}^j)$*. The importance weights* ω_t^j *are given by (7.30).*

The importance sampling distribution $g_t^\epsilon(\epsilon_t|s_{t-1})$ can be constructed by applying the methods described previously to a version of the DSGE model with extended state space ς_t.

7.3.2 Degenerate Measurement Error Distributions

We saw in Section 7.1 that the bootstrap particle filter deteriorates as the measurement error variance decreases. If the measurement error variance $\Sigma_u = 0$, then only particles that can exactly predict the current-period observation will get non-zero weight. Under a continuous distribution of the innovations ϵ_t the probability of generating such particles in the forecasting step is zero. Our discussion of the conditionally-optimal importance distribution suggests that in the absence of measurement errors, we should solve the system of equations

$$y_t = \Psi\big(\Phi(s_{t-1}^j, \tilde{\epsilon}_t^j)\big), \tag{7.32}$$

to determine ϵ_t^j as a function of s_{t-1}^j and the current observation y_t. We then can define

$$\omega_t^j = p^\epsilon(\tilde{\epsilon}_t^j) \quad \text{and} \quad \tilde{s}_t^j = \Phi(s_{t-1}^j, \tilde{\epsilon}_t^j). \tag{7.33}$$

In a nonlinear state-space system, e.g., one that arises from a higher-order perturbation solution there maybe be multiple solutions to the system even if the dimension of y_t and ϵ_t are equal. Let the solutions be denoted by $\bar{\epsilon}_t^j(k)$, $k = 1, \ldots, K$. The t subscript and j superscript indicate that the solutions depend on y_t and s_{t-1}^j. The importance distribution represented by the density $g_t^\epsilon(\tilde{\epsilon}_t^j | s_{t-1}^j)$ in (7.30) is now a multinomial distribution of the form

$$\mathbb{P}\{\tilde{\epsilon}_t^j = \bar{\epsilon}_t^i(k)\} = \frac{p^\epsilon(\bar{\epsilon}_t^j(j))}{\sum_{k=1}^K p^\epsilon(\bar{\epsilon}_t^j(k))}, \quad k = 1, \ldots, K, \tag{7.34}$$

which leads to

$$\omega_t^j = \sum_{k=1}^K p^\epsilon(\bar{\epsilon}_t^j(k)). \tag{7.35}$$

By construction, $p(y_t|\tilde{s}_t^j)$ corresponds to a pointmass at y_t for each particle j. Thus, we can define the incremental weight $\tilde{w}_t^j$ in (7.23) simply as $\tilde{w}_t^j = \omega_t^j$.

There are two computational challenges. First, one has to find *all* the (real) solutions to a nonlinear system of equations. For instance, if the DSGE model has been solved with a second-order perturbation method, then one has to solve a system of quadratic equations for each particle j to determine the $\bar{\epsilon}_t^j(k)$'s. (*** reference to Foerster, Rubio-Ramirez, Waggoner, and Zha? ***). The second computational problem can be illustrated in the context of the simple state space model presented in Chapter 4.3:

$$y_t = s_{1,t} + s_{2,t}, \quad s_{1,t} = \phi_2 s_{1,t-1} + \epsilon_t, \quad s_{2,t} = \phi_3 s_{1,t-1} + \phi_2 s_{1,t-1}.$$

Note that due to the absence of measurement errors, it is possible to recursively solve for the entire sequence of particles $s^j_{1:T}$ conditional on the initial draws $s^j_0 = (s^j_{1,0}, s^j_{2,0})$ and the observations $Y_{1:T}$. The particles will be reweighted based on $p^\epsilon(\tilde{\epsilon}^j_t)$ which captures the likelihood of observation y_t conditional on s^j_{t-1}. The resampling step of the filter duplicates the particles for which $p^\epsilon(\tilde{\epsilon}^j_t)$ is large. But unlike in the case of a model with measurement errors, the duplicate particles do not mutate in the subsequent iteration, because if two particles i and j are identical in period τ, i.e., $s^j_\tau = s^j_\tau$, then $s^j_t = s^j_t$ for $t > \tau$. Thus, the degeneracy problem does not manifest itself in an uneven distribution of particles. Instead, it is reflected by the fact that the particle values are mostly identical. This will lead to an imprecise approximation of the likelihood function, which is not surprising as the algorithm essentially approximates the integral $\int p(Y_{1:T}|s_0)p(s_0)ds_0$ by sampling s^j_0 from $p(s_0)$ and then evaluates the Monte Carlo average $\frac{1}{M}\sum_{j=1}^M p(Y_{1:T}|s^j_0)$.

7.4 Improving the Performance of Particle Filters

There exists a large literature on the implementation and the improvement of the particle filters in Algorithms 11 and 12. Detailed references to this literature are provided, for instance, in Doucet, de Freitas, and Gordon (2001), Cappé, Godsill, and Moulines (2007), Doucet and Johansen (2011), Creal (2012). We will focus in this section on a few key issues that are central to DSGE model applications. First, we will provide some discussion on how to choose the proposal distribution $g_t(\tilde{s}^j_t|s^j_{t-1})$ for Algorithm 12. Second, we will consider specific features of state-space representations derived from DSGE models, including deterministic state transitions and the potential absence of measurement errors. Finally, we consider alternative versions of the particle filter, including the auxiliary particle filter, a filter with resample-move step, and a filter for conditionally linear models.

7.4.1 Conditionally-Optimal Importance Distribution

The conditionally-optimal distribution, e.g., Liu and Chen (1998), is defined as the distribution that minimizes the Monte Carlo variation of the importance weights. However, this notion of optimality conditions on the current observation y_t as well as the $t-1$ particle s^j_{t-1}. Given (y_t, s^j_{t-1}) the weights $\tilde{w}^j_t$ are constant (as a function of $\tilde{s}_t$) if

$$g_t(\tilde{s}_t|s^j_{t-1}) = p(\tilde{s}_t|y_t, s^j_{t-1}), \tag{7.36}$$

that is, $\tilde{s}_t$ is sampled from the posterior distribution of the period t state given (y_t, s_{t-1}^j). In this case

$$\tilde{w}_t^j = \int p(y_t|s_t)p(s_t|s_{t-1}^j)ds_t. \tag{7.37}$$

For most DSGE model applications it is not possible to sample directly from $p(\tilde{s}_t|y_t, s_{t-1}^j)$. One notable exception is the case in which the DSGE model takes the form of the linear Gaussian state-space model (7.2). In this case we can obtain $p(\tilde{s}_t|y_t, s_{t-1}^j)$ from the Kalman filter updating step described in Table 2.1. Let

$$\begin{array}{lcllcl} \bar{s}_{t|t-1}^j & = & \Phi_1 s_{t-1}^j & P_{t|t-1} & = & \Phi_\epsilon \Sigma_\epsilon \Phi_\epsilon' \\ \bar{y}_{t|t-1}^j & = & \Psi_0 + \Psi_1 t + \Psi_2 \bar{s}_{t|t-1}^j & F_{t|t-1} & = & \Psi_2 P_{t|t-1} \Psi_2' + \Sigma_u \\ \bar{s}_{t|t}^j & = & \bar{s}_{t|t-1}^j + P_{t|t-1}\Psi_2' F_{t|t-1}^{-1}(y_t - \bar{y}_{t|t-1}) & P_{t|t} & = & P_{t|t-1} - P_{t|t-1}\Psi_2' F_{t|t-1}^{-1}\Psi_2 P_{t|t-1}. \end{array}$$

The conditionally optimal proposal distribution is given by

$$\tilde{s}_t|(s_{t-1}^j, y_t) \sim N\big(\bar{s}_{t|t}^j, P_{t|t}\big). \tag{7.38}$$

In our numerical illustrations we will use (7.38) as a benchmark against which we evaluate the accuracy of feasible alternatives. If it is not possible to sample directly from $p(\tilde{s}_t|y_t, s_{t-1}^j)$, one could use an accept-reject algorithm as discussed in Künsch (2005). However, for this approach to work efficiently, the user needs to find a good proposal distribution within the accept-reject algorithm.

7.4.2 Approximately Conditionally-Optimal Distributions

If sampling from the conditionally-optimal importance distribution is infeasible or computationally too costly, then one could try to sample from an approximately conditionally-optimal importance distribution. For instance, if the DSGE model nonlinearity arises from a higher-order perturbation solution and the nonlinearities are not too strong, then an approximately conditionally-optimal importance distribution could be obtained by applying the one-step Kalman filter updating in (7.38) to the first-order approximation of the DSGE model. More generally, as suggested in Guo, Wang, and Chen (2005), one could use the updating steps of a conventional nonlinear filter, such as an extended Kalman filter, unscented Kalman filter, or a Gaussian quadrature filter, to construct an efficient proposal distribution. Approximate filters for nonlinear DSGE models have been developed by Andreasen (2013) and Kollmann (2014).

7.4.3 Conditional Linear Gaussian Models

Certain DSGE models have a conditional linear structure that can be exploited to improve the efficiency of the particle filter. These models include the class of Markov switching linear rational expectations (MS-LRE) models analyzed in Farmer, Waggoner, and Zha (2009) as well as models that are linear conditional on exogenous stochastic volatility processes, e.g., the linearized DSGE model with heteroskedastic structural shocks estimated by Justiniano and Primiceri (2008) and the long-run risks model studied in Schorfheide, Song, and Yaron (2014).

For concreteness, consider an MS-LRE model obtained by replacing the fixed target-inflation rate π^* in the monetary policy rule (1.23) with a time-varying process $\pi_t^*(\varsigma_t)$ of the form

$$\pi_t^* = \varsigma_t \pi_L^* + (1-\varsigma_t)\pi_H^*, \quad \mathbb{P}\{\varsigma_t = l | \varsigma_{t-1} = l\} = \eta_{ll}, \quad l \in \{0,1\}. \tag{7.39}$$

This model was estimated in Schorfheide (2005).[1] Log-linearizing the model with Markov-switching target inflation rate leads to a MS-LRE similar to (2.1), except that the log-linearized monetary policy rule now contains an intercept that depends on ς_t. The solution to an MS-LRE model can be expressed as

$$\begin{aligned} y_t &= \Psi_0(\varsigma_t) + \Psi_1(\varsigma_t)t + \Psi_2(\varsigma_t)s_t + u_t, \quad u_t \sim N(0, \Sigma_u), \\ s_t &= \Phi_0(\varsigma_t) + \Phi_1(\varsigma_t)s_{t-1} + \Phi_\epsilon(\varsigma_t)\epsilon_t, \quad \epsilon_t \sim N(0, \Sigma_\epsilon), \end{aligned} \tag{7.40}$$

where ς_t follows a discrete Markov-switching process. In (7.40) we allow for Markov switches in all coefficient matrices, which may arise if not only the intercepts but also the slope coefficients of the linear rational expectations system depend on ς_t as, for instance, in Davig and Leeper (2007) and Bianchi (2013). Solution methods for general MS-LRE models are provided by Farmer, Waggoner, and Zha (2009).

The state-space representation in (7.40) is linear conditional on ς_t. In slight abuse of notation, we abbreviate the transition kernel for ς_t by $p(\varsigma_t|\varsigma_{t-1})$ and use the notation $p(\varsigma_t|Y_{1:t})$ for the distribution of ς_t given $Y_{1:t}$. The joint distribution of (ς_t, s_t) can be factorized as follows:

$$p(\varsigma_t, s_t|Y_{1:t}) = p(\varsigma_t|Y_{1:t})p(s_t|\varsigma_t, Y_{1:t}). \tag{7.41}$$

In conditionally linear Gaussian state-space models, the distribution $p(s_t|\varsigma_t, Y_{1:t})$ is normal:

$$s_t|(\varsigma_t, Y_{1:t}) \sim N\big(\bar{s}_{t|t}(\varsigma_t), P_{t|t}(\varsigma)_t\big). \tag{7.42}$$

[1] A richer DSGE model with a Markov switching target inflation rate was subsequently estimated by Liu, Waggoner, and Zha (2011).

We abbreviate the density of $s_t|(\varsigma_t, Y_{1:t})$ by $p_N\big(s_t|\bar{s}_{t|t}(\varsigma_t), P_{t|t}(\varsigma_t)\big)$, where the N subscript emphasizes the conditional Normal distribution. Because the vector of means $\bar{s}_{t|t}(\varsigma_t)$ and the covariance matrix $P_{t|t}(\varsigma)_t$ are sufficient statistics for the conditional distribution of s_t, we approximate $(\varsigma_t, s_t)|Y_{1:t}$ by the the quadruplets $\{\varsigma_t^j, \bar{s}_{t|t}^j, P_{t|t}^j, W_t^j\}_{i=1}^N$. The swarm of particles approximates

$$\begin{aligned}\int h(\varsigma_t, s_t)p(\varsigma_t, s_t, Y_{1:t})d(\varsigma_t, s_t) &= \int \left[\int h(\varsigma_t, s_t)p(s_t|\varsigma_t, Y_{1:t})ds_t\right] p(\varsigma_t|Y_{1:t})d\varsigma_t \\ &\approx \frac{1}{M}\sum_{j=1}^{M}\left[\int h(\varsigma_t^j, s_t^j)p_N\big(s_t|\bar{s}_{t|t}^j, P_{t|t}^j\big)ds_t\right] W_t^j.\end{aligned} \tag{7.43}$$

The Monte Carlo approximation in (7.43) is constructed by first integrating out $s_t|\varsigma_t^j$ based on the conditional Normal distribution which for many functions $h(\varsigma_t, s_t)$ can be done analytically, and then use Monte Carlo averaging to integrate out ς_t. This approach is called Rao-Blackwellization. It is a variance reduction technique that exploits the following inequality:

$$\mathbb{V}[h(s_t, \varsigma_t)] = \mathbb{E}\big[\mathbb{V}[h(s_t, \varsigma_t)|\varsigma_t]\big] + \mathbb{V}\big[\mathbb{E}[h(s_t, \varsigma_t)|\varsigma_t]\big] \geq \mathbb{V}\big[\mathbb{E}[h(s_t, \varsigma_t)|\varsigma_t]\big]. \tag{7.44}$$

Algorithm 12 can be easily be modified to exploit the conditionally Gaussian structure and integrate out $s_t|\varsigma_t$ analytically. The modification is due to Chen and Liu (2000) who referred to the resulting algorithm as mixture Kalman filter (see also Liu, Chen, and Logvinenko (2001)). Suppose that $\{\varsigma_{t-1}^j, \bar{s}_{t-1|t-1}^j, P_{t-1|t-1}^j, W_{t-1}^j\}$ satisfies (7.43). To forecast the states in period t, generate $\tilde{\varsigma}_t^j$ from the importance sampling distribution $g_t(\tilde{\varsigma}_t|\varsigma_{t-1}^j)$ and define the importance weights: Define the importance weights

$$\omega_t^j = \frac{p(\tilde{\varsigma}_t^j|\varsigma_{t-1}^j)}{g_t(\tilde{\varsigma}_t^j|\varsigma_{t-1}^j)}. \tag{7.45}$$

The Kalman filter forecasting step can be used to compute the conditional mean and variances of s_t and y_t given $\tilde{\varsigma}_t^j$:

$$\begin{aligned}\tilde{s}_{t|t-1}^j &= \Phi_0(\tilde{\varsigma}_t^j) + \Phi_1(\tilde{\varsigma}_t^j)s_{t-1}^j \\ P_{t|t-1}^j &= \Phi_\epsilon(\tilde{\varsigma}_t^j)\Sigma_\epsilon(\tilde{\varsigma}_t^j)\Phi_\epsilon(\tilde{\varsigma}_t^j)' \\ \tilde{y}_{t|t-1}^j &= \Psi_0(\tilde{\varsigma}_t^j) + \Psi_1(\tilde{\varsigma}_t^j)t + \Psi_2(\tilde{\varsigma}_t^j)\tilde{s}_{t|t-1}^j \\ F_{t|t-1}^j &= \Psi_2(\tilde{\varsigma}_t^j)P_{t|t-1}^j\Psi_2(\tilde{\varsigma}_t^j)' + \Sigma_u.\end{aligned} \tag{7.46}$$

Then,

$$\begin{aligned}\int h(\varsigma_t, s_t) p(\varsigma_t, s_t | Y_{1:t-1} d(\varsigma_t, s_t) &= \int \left[\int h(\varsigma_t, s_t) p(s_t | \varsigma_t, Y_{1:t-1}) ds_t \right] p(\varsigma_t | Y_{1:t-1}) d\varsigma_t \quad (7.47) \\ &\approx \frac{1}{M} \sum_{j=1}^{M} \left[\int h(\varsigma_t^j, s_t^j) p_N\big(s_t | \tilde{s}_{t|t-1}^j, P_{t|t-1}^j\big) ds_t \right] \omega_t^j W_{t-1}^j\end{aligned}$$

The likelihood approximation is based on the incremental weights $\tilde{w}_t^j$ which are given by

$$\tilde{w}_t^j = p_N\big(y_t | \tilde{y}_{t|t-1}^j, F_{t|t-1}^j\big) \omega_t^j. \quad (7.48)$$

The mean $\tilde{y}_{t|t-1}^j$ and variance $F_{t|t-1}^j$ of the regime-conditional predictive distribution were defined in (7.46). Conditional on $\tilde{\varsigma}_t^j$ we can use the Kalman filter once more to update the information about s_t in view of the current observation y_t:

$$\begin{aligned}\tilde{s}_{t|t}^j &= \tilde{s}_{t|t-1}^j + P_{t|t-1}^j \Psi_2(\tilde{\varsigma}_t^j)' \big(F_{t|t-1}^j\big)^{-1} (y_t - \bar{y}_{t|t-1}^j) \\ \tilde{P}_{t|t}^j &= P_{t|t-1}^j - P_{t|t-1}^j \Psi_2(\tilde{\varsigma}_t^j)' \big(F_{t|t-1}^j\big)^{-1} \Psi_2(\tilde{\varsigma}_t^j) P_{t|t-1}^j.\end{aligned} \quad (7.49)$$

Overall, this leads to the following algorithm:

Algorithm 14 (SISR For Conditionally Linear Models)

1. **Initialization.** *(Same as Algorithm 11)*

2. **Recursion.** *For $t = 1, \ldots, T$:*

   (a) **Forecasting s_t.** *Draw $\tilde{\varsigma}_t^j$ from density $g_t(\tilde{\varsigma}_t | \varsigma_{t-1}^j, \theta)$, calculate the importance weights ω_t^j in (7.45), and compute $\tilde{s}_{t|t-1}^j$ and $P_{t|t-1}^j$ according to (7.46). An approximation of $\mathbb{E}[h(s_t, \varsigma_t) | Y_{1:t-1}, \theta]$ is given by (7.47).*

   (b) **Forecasting y_t.** *Compute the incremental weights $\tilde{w}_t^j$ according to (7.48). The predictive density $p(y_t | Y_{1:t-1}, \theta)$ can be approximated by*

$$\hat{p}(y_t | Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j. \quad (7.50)$$

   (c) **Updating.** *Define the normalized weights*

$$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^j W_{t-1}^j} \quad (7.51)$$

   and compute $\tilde{s}_{t|t}^j$ and $\tilde{P}_{t|t}^j$ according to (7.49). An approximation of $\mathbb{E}[h(\varsigma_t, s_t) | Y_{1:t}, \theta]$ can be obtained by $\{\tilde{\varsigma}_t^j, \tilde{s}_{t|t}^j, \tilde{P}_{t|t}^j, \tilde{W}_t^j\}$ according to (7.43).

(d) **Selection.** *Resample* $\{\tilde{\varsigma}_t^j, \tilde{s}_{t|t}^j, \tilde{P}_{t|t}^j, \tilde{W}_t^j\}$ *to obtain* $\{\varsigma_t^j, s_{t|t}^j, P_{t|t}^j, W_t^j\}$. *(Similar to Algorithm 11)*

3. **Likelihood Approximation.** *(Same as Algorithm 11).*

7.4.4 Resample-Move Steps

The resampling step of Algorithms 11 and 12 is designed to equalize the distribution of particle weights and avoid a degenerate particle distribution. However, as the discussion of the model without measurement errors in Section 7.3.2 highlighted, it is possible that an even distribution of particle weights coincides with (nearly) identical particle values (impoverishment), which leads to potentially very inaccurate Monte Carlo approximations. While the sampling of $\tilde{s}_t^j$ from the proposal distribution $g_t(\tilde{s}_t|s_{t-1}^j)$ leads to some diversity of the period t particles even if the $t-1$ particle values are identical, a mutation step that "jitters" the particle values after resampling may help to increase the diversity of particle values and improve the accuracy of the filter. This "jittering" is comparable to the mutation step in the SMC Algorithm 8, used for posterior inference on the model parameter vector θ. Thus, the resampling step of the particle filter is augmented by a "move" step as in Berzuini and Gilks (2001).

The resample-move algorithm presented below is a special case of the algorithm described in Doucet and Johansen (2011). To understand how a particle filter with resample-move step works, it is useful to abstract from the resampling first and to introduce the mutation right after the updating in Step 2.(c) of Algorithm 12. The particle mutation is based on a Markov transition kernel, which we denote by $K_{(y_t,s_{t-1})}(s_t|\tilde{s}_t)$. The transition kernel transforms the particle $\tilde{s}_t^j$ into a particle s_t^j by sampling from a conditional distribution of s_t given $\tilde{s}_t$. The transition kernel depends on the current observation y_t as well as the period $t-1$ value of the state s_{t-1}, which is indicated by the subscript.

We generate the transition kernel from a sequence of MH steps. To simplify the exposition we focus on the case of a single MH step:

Algorithm 15 (Mutation Step for Resample-Move Algorithm)

1. Draw ς_t *from a density* $q_t(\varsigma_t|\tilde{s}_t^j)$.

2. Set $s_t^j = \varsigma_t$ *with probability*

$$\alpha_t(\varsigma_t|\tilde{s}_t^j) = \min\left\{1, \frac{p(y_t|\varsigma_t)p(\varsigma_t|s_{t-1}^j)/q_t(\varsigma_t|\tilde{s}_t^j)}{p(y_t|\tilde{s}_t)p(\tilde{s}_t|s_{t-1}^j)/q_t(\tilde{s}_t^j|\varsigma_t)}\right\}$$

and $s_t^j = \tilde{s}_t^j$ *otherwise.*

By construction, the transition kernel satisfies an important invariance property, which can be established using the same steps as in Chapter 3.4.2. Write

$$K_{(y_t,s_{t-1})}(s_t|\tilde{s}_t) = u_t(s_t|\tilde{s}_t) + r_t(\tilde{s}_t)\delta_{\tilde{s}_t}(s_t), \tag{7.52}$$

where

$$u_t(s_t|\tilde{s}_t) = \alpha_t(s_t|\tilde{s}_t)q_t(s_t|\tilde{s}_t), \quad r_t(\tilde{s}_t) = 1 - \int u_t(s_t|\tilde{s}_t)ds_t \tag{7.53}$$

One can verify the reversibility property

$$p(y_t|\tilde{s}_t)p(\tilde{s}_t|s_{t-1})u_t(s_t|\tilde{s}_t) = p(y_t|s_t)p(s_t|s_{t-1})u_t(\tilde{s}_t|s_t). \tag{7.54}$$

Using the reversibility result, we obtain the following invariance property

$$\begin{aligned}
&\int K_{(y_t,s_{t-1})}(s_t|\tilde{s}_t)p(y_t|\tilde{s}_t)p(\tilde{s}_t|s_{t-1})d\tilde{s}_t \qquad (7.55)\\
&= \int p(y_t|\tilde{s}_t)p(\tilde{s}_t|s_{t-1})u_t(s_t|\tilde{s}_t)d\tilde{s}_t + \int p(y_t|\tilde{s}_t)p(\tilde{s}_t|s_{t-1})r_t(\tilde{s}_t)\delta_{\tilde{s}_t}(s_t)d\tilde{s}_t\\
&= \int p(y_t|s_t)p(s_t|s_{t-1})u_t(\tilde{s}_t|s_t)d\tilde{s}_t + p(y_t|s_t)p(s_t|s_{t-1})r_t(s_t)\\
&= p(y_t|s_t)p(s_t|s_{t-1}).
\end{aligned}$$

The second equality follows from (7.54) and the last equality follows from the definition of the rejection probability $r_t(\cdot)$ in (7.53). Suppose that particle values are sampled according to

$$\tilde{s}_t^j \sim g_t(\tilde{s}_t^j|s_{t-1}) \quad \text{and} \quad s_t^j \sim K_{(y_t,s_{t-1}^j)}(s_t^j|\tilde{s}_t^j). \tag{7.56}$$

Then, we obtain the following approximation:

$$\begin{aligned}
&\int_{s_{t-1}}\int_{s_t} h(s_t)p(y_t|s_t)p(s_t|s_{t-1})p(s_{t-1}|Y_{1:t-1})ds_t ds_{t-1} \qquad (7.57)\\
&= \int_{s_{t-1}}\int_{\tilde{s}_t}\int_{s_t} h(s_t)K_{(y_t,s_{t-1})}(s_t|\tilde{s}_t)\frac{p(y_t|\tilde{s}_t)p(\tilde{s}_t|s_{t-1})}{g_t(\tilde{s}_t|s_{t-1})}g_t(\tilde{s}_t|s_{t-1})ds_t d\tilde{s}_t ds_{t-1}\\
&\approx \frac{1}{M}\sum_{j=1}^{M} h(s_t^j)\tilde{w}_t^j W_{t-1}^j, \quad \text{where} \quad \tilde{w}_t^j = \frac{p(y_t|\tilde{s}_t^j)p(\tilde{s}_t^j|s_{t-1}^j)}{g_t(\tilde{s}_t^j|s_{t-1}^j)}
\end{aligned}$$

To complete the heuristic derivations for the resample-move algorithm, notice that we can introduce a resampling step before the mutation step in which we generate draws $(\hat{s}_t^j, \hat{s}_{t-1}^j)$ from a multinomial distribution characterized by the support points and weights $\left\{(\tilde{s}_t^j, s_{t-1}^j), \tilde{W}_t^j\right\}$ with $\tilde{W}_t^j \propto \tilde{w}_t^j W_{t-1}^j$. Resampling before an MCMC step will always lead to greater sample diversity than performing the steps in the other order. After the resampling we can set the weights $W_t^j = 1$ and draw $s_t^j \sim K_{(y_t, \hat{s}_{t-1}^j)}(s_t|\hat{s}_t^j)$, which leads to the following approximation:

$$\int_{s_{t-1}} \int_{s_t} h(s_t) p(y_t|s_t) p(s_t|s_{t-1}) p(s_{t-1}|Y_{1:t-1}) ds_t ds_{t-1} \approx \frac{1}{M} \sum_{j=1}^{M} h(s_t^j) W_t^j. \tag{7.58}$$

The sequential importance sampling algorithm with resample-move step can be summarized as follows:

Algorithm 16 (Sequential Importance Sampling with Resample-Move Step)

Replace Step 2.(d) of Algorithm 12 by:

2.(d)' **Resample-Move Step:**

(i) Resample the particles via multinomial ressampling. Let $\{(\hat{s}_t^j, \hat{s}_{t-1}^j\}_{i=1}^N$ *denote* N *iid draws from a multinomial distribution characterized by support points and weights* $\left\{\{(\tilde{s}_t^j, s_{t-1}^j), \tilde{W}_t^j\right\}$ *and set* $W_t^j = 1$.

(ii) Use Algorithm 15 to generate draws s_t^j *from Markov-transition kernel* $K_{(y_t, \hat{s}_{t-1}^j)}(s_t|\hat{s}_t^j)$.

7.4.5 Auxiliary Particle Filter

The auxiliary particle filter is due to Pitt and Shephard (1999) (see also Pitt and Shephard (2001)). The original version of the auxiliary particle filter contained two resampling steps. However, subsequent research has shown that a single resampling step is preferable. Our description of the algorithm follows Doucet and Johansen (2011) and has the same structure as Algorithm 12. Let $\tilde{p}(y_t|s_{t-1})$ be an approximation of the one-step-ahead predictive density $p(y_t|s_{t-1})$. The density $\tilde{p}(y_t|s_{t-1})$ can be obtained, for instance, by iterating the state-transition equation forward (based on the s_{t-1}^j's and draws of ϵ_t), averaging the simulated s_t's to form an $\bar{s}_{t|t}^j$ and using a modified version of the measurement equation to form a fat-tailed density $\tilde{p}(y_t|\bar{s}_{t|t}^j)$.

The auxiliary particle filter is based on two sets of weights. The first set of weights tracks an auxiliary posterior distribution $\tilde{p}(s_{t-1}|Y_{1:t})$ defined as

$$\tilde{p}(s_{t-1}|Y_{1:t}) = \frac{\tilde{p}(y_t|s_{t-1})p(s_{t-1}|Y_{1:t-1})}{\tilde{p}(y_t|Y_{1:t-1})}, \tag{7.59}$$

where the auxiliary marginal data density is defined as

$$\tilde{p}(y_t|Y_{1:t-1}) = \int \tilde{p}(y_t|s_{t-1})p(s_{t-1}|Y_{1:t-1})ds_{t-1}. \tag{7.60}$$

We will begin the derivations with the assumption that the $t-1$ particle swarm $\{s_{t-1}^j, W_{t-1}^j\}_{j=1}^M$ approximates the auxiliary posterior distribution

$$\frac{1}{M}\sum_{j=1}^{M} h(s_{t-1}^j)W_{t-1}^j \approx \int h(s_{t-1})\tilde{p}(s_{t-1}|Y_{1:t})ds_{t-1} \tag{7.61}$$

and then manipulate the particle swarm to obtain an approximation of $\tilde{p}(s_t|Y_{1:t=1})$. A second set of weights is introduced subsequently to approximate the posterior of interest $p(s_t|Y_{1:t})$.

Suppose that the time t particles $\tilde{s}_t^j$ are sampled from the importance density $g_t(\tilde{s}_t|s_{t-1}^j)$. Now define the incremental weights

$$\tilde{w}_t^j = p(y_t|\tilde{s}_t^j)\frac{p(\tilde{s}_t^j|s_{t-1}^j)}{g_t(\tilde{s}_t^j|s_{t-1}^j)}\frac{\tilde{p}(y_{t+1}|\tilde{s}_t^j)}{\tilde{p}(y_t|\tilde{s}_{t-1}^j)}. \tag{7.62}$$

These weights will replace the incremental weights in Equation (7.23) of Algorithm 12. Using the definitions of $\tilde{p}(s_{t-1}|Y_{1:t})$ and $\tilde{w}_t^j$ in (7.59) and (7.62), respectively, we deduce that

$$\begin{aligned}
&\frac{1}{M}\sum_{j=1}^{M} h(s_t^j)\tilde{w}_t^j W_{t-1}^j \qquad (7.63)\\
&\approx \int\int h(s_t)p(y_t|s_t)\frac{p(s_t|s_{t-1})\tilde{p}(y_{t+1}|s_t)}{g_t(s_t|s_{t-1})\tilde{p}(y_t|s_{t-1})}g_t(s_t|s_{t-1})\tilde{p}(s_{t-1}|Y_{1:t})ds_{t-1}ds_t\\
&= \frac{1}{\tilde{p}(y_t|Y_{1:t-1})}\int h(s_t)p(y_t|s_t)\tilde{p}(y_{t+1}|s_t)\left[\int p(s_t|s_{t-1})p(s_{t-1}|Y_{1:t-1})ds_{t-1}\right]ds_t.\\
&= \frac{1}{\tilde{p}(y_t|Y_{1:t-1})}\int h(s_t)\tilde{p}(y_{t+1}|s_t)p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t\\
&= \frac{p(y_t|Y_{1:t-1})}{\tilde{p}(y_t|Y_{1:t-1})}\int h(s_t)\tilde{p}(y_{t+1}|s_t)p(s_t|Y_{1:t})ds_t.
\end{aligned}$$

The first equality follows from (7.59) and the third equality utilizes Bayes Theorem to obtain a posterior for s_t given $(y_t, Y_{1:t-1})$. The normalization factor in front of the last integral is a nuisance, but it cancels once we take ratios. Define (this derivation ignores the distinction

between the updated normalized weights $\tilde{W}_t^j$ and the weights W_t^j after the resampling step in the particle filter algorithms)

$$W_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j} \tag{7.64}$$

and notice that

$$\frac{1}{M}\sum_{j=1}^M h(s_t^j) W_t^j \approx \frac{\int h(s_t)\tilde{p}(y_{t+1}|s_t)p(s_t|Y_{1:t})ds_t}{\int \tilde{p}(y_{t+1}|s_t)p(s_t|Y_{1:t})ds_t} = \int h(s_t)\tilde{p}(s_t|Y_{1:t+1})ds_t, \tag{7.65}$$

which is the time t version of (7.61).

A second set of weights is necessary, because unlike in the original Algorithm 12 the particle swarm $\{s_t^j, W_t^j\}$ does not deliver approximations to the objects of interest, which are the predictive likelihood $p(y_t|Y_{1:t-1})$ and $p(s_t|Y_{1:t})$. Dividing $\tilde{w}_t^j$ in (7.62) by $\tilde{p}(y_{t+1}|\tilde{s}_t^j)$ yields the alternative weights:

$$\bar{w}_t^j = p(y_t|\tilde{s}_t^j)\frac{p(\tilde{s}_t^j|s_{t-1}^j,\theta)}{g_t(\tilde{s}_t^j|s_{t-1}^j,\theta)}\frac{1}{\tilde{p}(y_t|\tilde{s}_{t-1}^j)}, \quad \bar{W}_t^j = \frac{\bar{w}_t^j W_{t-1}^j}{\frac{1}{M}\sum_{j=1}^M \bar{w}_t^j W_{t-1}^j}. \tag{7.66}$$

Using the same steps as in (7.63) one can verify that

$$\frac{1}{M}\sum_{j=1}^M h(\tilde{s}_t^j)\bar{w}_t^j W_{t-1}^j \approx \frac{1}{\tilde{p}(y_t|Y_{1:t-1})}\int h(s_t)p(y_t|s_t)p(s_t|Y_{1:t})ds_t. \tag{7.67}$$

It follows immediately that the posterior of $s_t|Y_{1:t}$ can be approximated according to:

$$\int h(s_t)p(s_t|Y_{1:t})ds_t \approx \frac{1}{M}\sum_{j=1}^M h(s_t^j)\bar{W}_t^j. \tag{7.68}$$

The factor $1/\tilde{p}(y_t|Y_{1:t-1})$ cancels due to the definition of $\bar{W}_t^j$ in (7.66) as a ratio. The approximation of the likelihood increment, on the other hand requires an estimate of $\tilde{p}(y_t|Y_{1:t-1})$. Such an estimate can be obtained from:

$$\begin{aligned}\frac{1}{M}\sum_{j=1}^M \frac{1}{\tilde{p}(y_t|s_{t-1}^j)}W_{t-1}^j &\approx \int \frac{1}{\tilde{p}(y_t|s_{t-1})}\frac{\tilde{p}(y_t|s_{t-1})p(s_{t-1}|Y_{1:t-1})}{\tilde{p}(y_t|Y_{1:t-1})}ds_{t-1} \\ &= \frac{1}{\tilde{p}(y_t|Y_{1:t-1})}.\end{aligned} \tag{7.69}$$

Here we used (7.59) and (7.61). Thus, we obtain

$$p(y_t|Y_{1:t-1}) \approx \left(\frac{1}{M}\sum_{j=1}^M \bar{w}_t^j W_{t-1}^j\right)\left(\frac{1}{M}\sum_{j=1}^M \frac{1}{\tilde{p}(y_t|s_{t-1}^j)}W_{t-1}^j\right)^{-1}. \tag{7.70}$$

The potential advantage of the auxiliary particle filter is that the incremental weights $\bar{w}_t^j$ are more stable than in Algorithms 11 and 12. *** this seems advantageous for filtering the state, but not so much for the likelihood function - because the likelihood function requires to average $1/\tilde{p}(y_t|s_{t-1}^j)$. However, also notice that

$$\frac{\tilde{w}_{t-1}^j}{\tilde{p}(y_t|s_{t-1}^j)} = \frac{p(y_{t-1}|\tilde{s}_{t-1}^j)p(\tilde{s}_t^j|s_{t-1}^j)}{\tilde{p}(y_{t-1}|\tilde{s}_{t-2}^j)g_t(\tilde{s}_t^j|s_{t-1}^j)}$$

Flesh this out some more *** Consider the proposal distribution of the bootstrap filter, which is given by $g_t(s_t|s_{t-1}) = p(s_t|s_{t-1})$. In this case $\bar{w}_t^j = p(y_t|\tilde{s}_t^j)/\tilde{p}(y_t|s_{t-1}^j)$ which potentially has much lower variance than $p(y_t|\tilde{s}_t^j)$. As in any importance sampling approximation, it is important that the density $\tilde{p}(y_t|s_{t-1}^j)$ has fatter tails than $p(y_t|\tilde{s}_t^j)$. The auxiliary particle filter can be summarized as follows:

Algorithm 17 (Auxiliary Particle Filter)

1. **Initialization.** *Draw the initial particles from the distribution* $s_0^j \overset{iid}{\sim} p(s_0)$ *and set* $\tilde{w}_0^j = \tilde{p}(y_1|s_0^j)$ *and* $W_0^j = \tilde{w}_0^j / \frac{1}{M}\sum_{j=1}^M \tilde{w}_0^j$.

2. **Recursion.** *For* $t = 1, \ldots, T$*:*

   (a) **Importance Sampling.** *Draw* $\tilde{s}_t^j$ *from density* $g_t(\tilde{s}_t|s_{t-1}^j, \theta)$ *and compute the incremental weights* $\tilde{w}_t^j$ *defined in (7.62) and* $\bar{w}_t^j$ *in (7.66). Also compute the normalized weights* $\tilde{W}_t^j \propto \tilde{w}_t^j$ *and* $\bar{W}_t^j \propto \bar{w}_t^j$.

   (b) **Forecasting** y_t**.** *The predictive density* $p(y_t|Y_{1:t-1}, \theta)$ *can be approximated by (7.70).*

   (c) **Updating.** *An approximation of* $\mathbb{E}[h(s_t)|Y_{1:t}, \theta]$ *is given by (7.68).*

   (d) **Selection.** *(Same as Algorithm 11)*

3. **Likelihood Approximation.** *The approximation of the log likelihood function is given by*

$$\ln \hat{p}(Y_{1:T}|\theta) = \sum_{t=1}^{T} \left[\ln \left(\frac{1}{M} \sum_{j=1}^{M} \bar{w}_t^j W_{t-1}^j \right) - \ln \left(\frac{1}{M} \sum_{j=1}^{M} \frac{1}{\tilde{p}(y_t|s_{t-1}^j)} W_{t-1}^j \right) \right]. \quad (7.71)$$

Table 7.1: Parameter Values For Likelihood Evaluation

Parameter	θ^m	θ^l	Parameter	θ^m	θ^l
τ	2.09	3.26	κ	0.98	0.89
ψ_1	2.25	1.88	ψ_2	0.65	0.53
ρ_r	0.81	0.76	ρ_g	0.98	0.98
ρ_z	0.93	0.89	$r^{(A)}$	0.34	0.19
$\pi^{(A)}$	3.16	3.29	$\gamma^{(Q)}$	0.51	0.73
σ_r	0.19	0.20	σ_g	0.65	0.58
σ_z	0.24	0.29	$\ln p(Y\|\theta)$	-306.5	-313.4

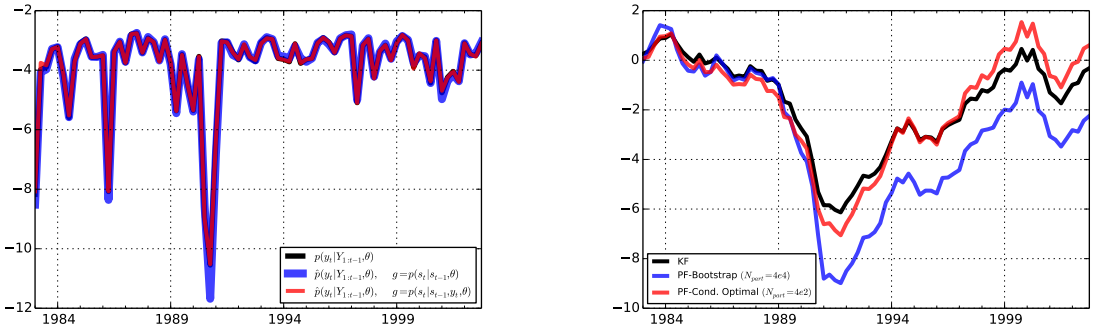
7.5 Application to the Small-Scale New Keynesian Model

To illustrate the particle filtering techniques, we will use the bootstrap PF, the conditionally-optimal PF, and the auxiliary PF to evaluate the likelihood function associated with the small-scale New Keynesian DSGE model. We do so for two parameter vectors, which are denoted by θ^m and θ^l and tabulated in Table 7.1. The value θ^m is chosen by searching among the posterior draws (see Chapter 4.2) for θ for the value associated with the highest likelihood. Note that this value is neither the posterior mode (because the mode is determined by the product of likelihood and prior) nor necessarily the maximum of the likelihood function (because the posterior sampler does not necessarily visit the area of the parameter space in which the likelihood function is maximized). The log likelihood at θ^m is -306.49. The second parameter value θ^l is associated with a lower log likelihood value of -313.36. To put the likelihood differentials into perspective, twice the log likelihood differential, 13.8, can be compared to the 5% χ^2 critical value for the hypothesis $\theta = \theta^m$ versus $\theta = \theta^l$ is 22.4. Thus, while the data prefer θ^m, they do not provide overwhelming evidence against θ^m.

The particle filters generate approximations of the period t contribution of the likelihood function, $\hat{p}(y_t|Y_{1:t-1}, \theta)$ and the distribution of the filtered states, $\hat{p}(s_t|Y_{1:t}, \theta)$. Because we are using the linearized version of the small-scale DSGE model, we can compare the approximations $\hat{p}(\cdot)$ to the exact densities $p(\cdot)$ obtained from the Kalman filter. We begin with a single run of the filters over the estimation period 1983:I to 2002:IV and compare the output of the Kalman filter, the bootstrap PF, and the conditionally-optimal PF. To facilitate the use of particle filters we augment the measurement equation of the DSGE model by independent measurement errors, whose standard deviations we set to be 20% of the standard deviation

Figure 7.1: Likelihood Approximation and Filtered State

$\ln \hat{p}(y_t|Y_{1:t-1}, \theta^m)$ vs. $\ln p(y_t|Y_{1:t-1}, \theta^m)$ $\quad$ $\widehat{\mathbb{E}}[\hat{g}_t|Y_{1:t}, \theta^m]$ vs. $\mathbb{E}[g_t|Y_{1:t}, \theta^m]$



Notes: The results depicted in the figure are based on a single run of the bootstrap PF, the conditionally-optimal PF, and the Kalman filter.

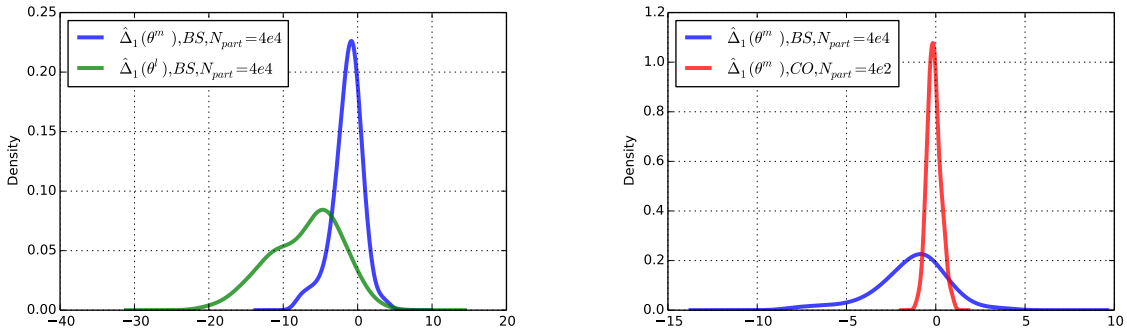
of the observables.[2] We use 40,000 particles for the bootstrap PF and 400 particles for the conditionally-optimal PF.

Throughout this chapter, the bootstrap PF can be viewed as providing a lower bound on the accuracy of particle-filter-based likelihood approximation, because this filter can easily be implemented in DSGE model applications provided that the measurement equation contains measurement errors. The conditionally-optimal filter typically not implementable for nonlinear DSGE models, but an approximate version that utilizes some other nonlinear filter to generate the proposal distribution may be implementable. Thus, we view it as a realistic upper bound on the accuracy that can be achieved with particle filters in DSGE model applications.

Figure 7.1 depicts the sequence of log-likelihood increments as well as the mean $\mathbb{E}[\hat{g}_t|Y_{1:t}]$, where $\hat{g}_t$ is the exogenous government spending in percentage deviations from its steady state. The log-likelihood approximation generated by the conditionally-optimal PF is visually indistinguishable from the exact log-likelihood. The bootstrap PF approximation deviates from the actual log likelihood more strongly, in particular in periods in which the likelihood is low, e.g., around 1986 and 1991. The results for the filtered demand shock are similar: the approximation of $\mathbb{E}[g_t|Y_{1:t}, \theta^m]$ obtained from the conditionally-optimal PF is fairly accurate, whereas there is a substantial discrepancy between the estimated path of $\hat{g}_t$

[2]The measurement error standard deviations are 0.1160 for output growth, 0.2942 for inflation, and 0.4476 for the interest rates

Figure 7.2: Distribution of Log-Likelihood Approximation Errors



Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y|\theta) - \ln p(Y|\theta)$ based on 100 runs of the PF.

produced by the Kalman filter and the bootstrap PF. A gap of about 2 percentages opens up in 1991, at the same time when the log likelihood drops below -10. Due to the persistence in the $\hat{g}_t$ process ($\rho_g = 0.98$), the gap does not close for the remainder of the sample.

To assess the accuracy of the particle filter approximation of the likelihood function we will now run the filters repeatedly and examine the sampling properties of the discrepancy

$$\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta) - \ln p(Y_{1:\theta}). \tag{7.72}$$

The results are depicted in Figure 7.2. The left panel compares the accuracy of the bootstrap filter for θ^m and θ^l. Conditional on θ^m most of the simulated values for $\hat{\Delta}_1$ fall in the range from -8 to 5 log-likelihood units. At θ^l the dispersion of $\hat{\Delta}_1$ is much larger and more skewed toward the left, encompassing values from -20 to 5. The deterioration of fit is associated with a deterioration in the approximation accuracy. This is not surprising because the bootstrap PF generates proposal draws for s_t through forward simulation of the state-transition equation. The worse the fit of the model, the greater the mismatch between the proposal distribution and the target posterior distribution of s_t.

The right panel of Figure 7.2 compares the distribution of $\hat{\Delta}_1$ for the bootstrap and the conditionally-optimal PF. The latter is a lot more precise than the former and the empirical distribution of $\hat{\Delta}_1$ is tightly centered around zero. The biases and standard deviations of $\hat{\Delta}_1$ for the two filters are summarized in Table 7.2. Conditional on θ^m, the standard deviation of $\hat{\Delta}_1$ is about six times larger for the bootstrap PF than for the conditionally-optimal PF. Changing the parameter to θ^l increases the standard deviation by a factor of 2.3 (1.4) for the

Table 7.2: Summary Statistics for Particle Filters

	Bootstrap	Cond. Opt.	Auxiliary
Number of Particles M	40,000	400	40,000
Number of Repetitions	100	100	100
High Posterior Density: $\theta = \theta^m$			
Bias $\hat{\Delta}_1$	-1.39	-0.10	-2.83
StdD $\hat{\Delta}_1$	2.03	0.37	1.87
Bias $\hat{\Delta}_2$	0.32	-0.03	-0.74
Low Posterior Density: $\theta = \theta^l$			
Bias $\hat{\Delta}_1$	-7.01	-0.11	-6.44
StdD $\hat{\Delta}_1$	4.68	0.44	4.19
Bias $\hat{\Delta}_2$	-0.70	-0.02	-0.50

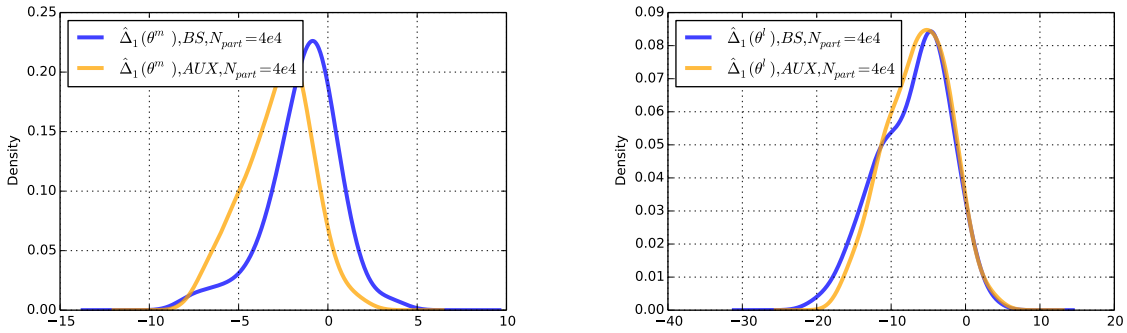
Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (7.72) and (7.73).

bootstrap PF (conditionally-optimal PF). Thus, the bootstrap PF is much more sensitive to the fit of the model specification that the conditionally-optimal PF.

As an alternative to the bootstrap PF and the conditionally-optimal PF we also consider the auxiliary PF. The distribution of log-likelihood approximation errors $\hat{\Delta}_1$ is plotted in Figure 7.3. Visually, results from the auxiliary PF and the bootstrap PF are very similar. For θ^m the downward bias is a bit more pronounced for the auxiliary PF, whereas for θ^l the distribution of $\hat{\Delta}_1$ is less skewed to the left. The last column of Table 7.2 reports sample moments for $\hat{\Delta}_1$ and $\hat{\Delta}_2$. While the auxiliary PF is able to reduce the variability of the log likelihood discrepancies, the small-sample bias for $\hat{\Delta}_1$ increases by a factor of 2 for θ^m compared to the bootstrap PF.

In Chapters 8 and 9 we will embed a particle filter into a posterior sampler. This is necessary to implement posterior inference for a nonlinear DSGE model. The key requirement for such algorithms to generate draws that can be used to consistently approximate moments and quantiles of the posterior distribution of θ based on a finite number of particles M is that the particle filter generates an unbiased approximation of the likelihood function $p(Y_{1:T}|\theta)$ and its increments $p(y_t|Y_{1:t-1}, \theta)$. While particle filter likelihood approximations are unbiased in theory, in practice the sampling distribution of the approximation may be highly

Figure 7.3: Distribution of Log-Likelihood Approximation Errors



Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y|\theta) - \ln p(Y|\theta)$ based on 100 runs of the PF.

skewed and fat-tailed, such that finite sample averages across a modest number of repetitions may appear biased. This may translate into slow convergence (or failure of convergence) of posterior samplers that rely on particle filter approximations.

In Figure 7.2 we focused on the distribution of the log likelihood approximation $\ln \hat{p}(Y_{1:T}|\theta)$ and it is quite apparent that it provides a downward-biased estimate of $\ln p(Y_{1:T}|\theta)$. The negative bias is expected from Jensen's inequality if the approximation of the likelihood function is unbiased, because the logarithmic transformation is concave. Assessing the bias of $\hat{p}(Y_{1:T}|\theta)$ is numerically delicate because exponentiating a log-likelihood value of around -300 leads to a missing value. Instead, we will consider the following statistic:

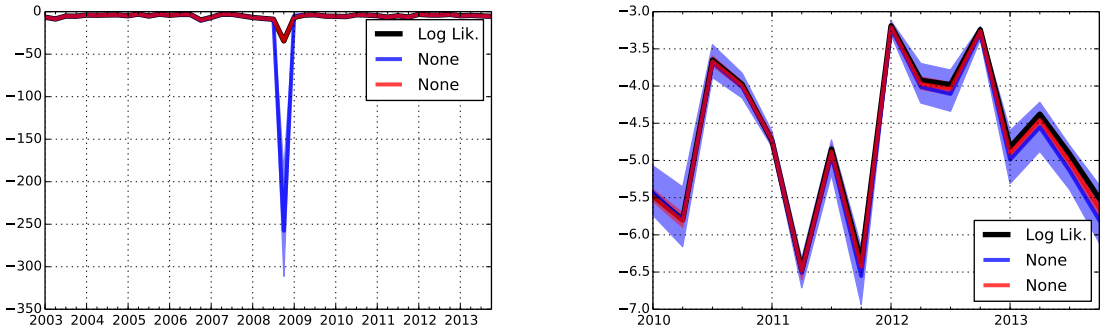
$$\hat{\Delta}_2 = \frac{\hat{p}(Y_{1:T}|\theta)}{p(Y_{1:T}|\theta)} - 1 = \exp[\ln \hat{p}(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)] - 1. \tag{7.73}$$

The computation of $\hat{\Delta}_2$ requires us to exponentiate the difference in log-likelihood values, which is feasible of the particle filter approximation is reasonably accurate. If the particle filter approximation is unbiased, then the sampling mean of $\hat{\Delta}_2$ is equal to zero.

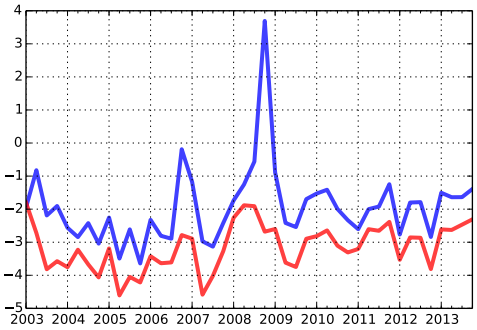
By construction, $\hat{\Delta}_2$ is bounded below by -1. The right panel of Figure 7.2 suggests that for the bootstrap PF, we expect the distribution of $\hat{\Delta}_2$ to have significant mass near -1 (note that $\exp[-5] \approx 0.007$) and a long right tail ($\exp[3] \approx 20$). Table 7.2 reports the means of $\hat{\Delta}_2$ across 100 repetitions: for the conditionally-optimal PF the means given θ^m and θ^l are essentially zero. For the bootstrap PF the mean is close zero conditional on θ^m, but substantially below zero for θ^l. The auxiliary PF is not able to reduce the small-sample bias of $\hat{\Delta}_2$ compared to the bootstrap PF. In fact, at θ^m the bias of the auxiliary PF is more than twice as large (in absolute terms) as the bias of the bootstrap filter.

Figure 7.4: Particle Filtering During the Great Recession and Beyond

Log-likelihood Increments $\hat{p}(y_t|Y_{1:t-1}, \theta^m)$



Log Standard Dev of Log-Likelihood Increments



Notes: Blue is bootstrap PF, red is conditionally-optimal PF.

By construction, the accuracy of the bootstrap PF is very sensitive to outliers in the observations. To the extent that outliers are unlikely under the entertained DSGE model, the forward simulation of the state vector is unlikely to yield many proposed states $\tilde{s}_t^j$ that can rationalize the observation y_t. This leads to an uneven distribution of particle weights and inaccurate Monte Carlo approximations. The recent Great Recession in 2008-09 was a large outlier from the perspective of DSGE models (as well as other popular time series models). Holding the parameter values θ^m and θ^l fixed, we now run the filters on the sample 2003:I to 2013:IV. Results are depicted in Figure 7.4.

The top left panel of Figure 7.4 depicts the sequence of log-likelihood increments. In 2008:IV, which is when output growth collapsed, the log-likelihood increment is substantially lower than in any other period. The conditionally-optimal PF still does well in tracking the actual likelihood, whereas the bootstrap PF approximation becomes highly inaccurate. The

bootstrap PF underestimates the likelihood increment by about 250 units on a log scale. Interestingly, the bootstrap PF recovers fairly quickly in subsequent periods. The top right panel depicts 90% bands for the approximations of the likelihood increments across the 100 repetitions. The width of the band for the bootstrap PF is generally less on 1 unit on the log scale. The bottom panel shows the log standard deviation of the log-likelihood increments. For the conditionally-optimal PF the log standard deviation stays fairly stable over time, though there appears to be a slight increase after 2008. For the bootstrap PF, the standard deviations are generally larger than for the conditionally-optimal PF and there is a large spike in 2008:Q4.

7.6 Application to the SW Model

Our second application of the particle filter considers the SW model. From a computational perspective, the SW model differs from the small-scale DSGE model in terms of the number of observables used in the estimation and with respect to the number of latent state variables. For the estimation of the small-scale New Keynesian model we used three observables and the model has one endogenous state variable and three exogenous shocks. The SW model is estimated based on seven variables and it has more than a dozen state variables. We will examine the extent to which the increased model size leads to a deterioration of the accuracy of the particle filter approximation. The large state space makes it more difficult to accurately integrate out the hidden state-variables with the filter, and the relatively large number of observables creates a potential for model misspecification, which in turn may lead to a deterioration of the bootstrap PF. Recall that the bootstrap PF is sensitive to the accuracy of forecasts of y_t based on the distribution $s_{t-1}|Y_{1:t-1}$.

As in the previous section, we compute the particle filter approximations conditional on two sets of parameter values, θ^m and θ^l, which are summarized in Table 7.3. θ^m is the parameter vector associated with the highest likelihood value among the draws that we previously generated with our posterior sampler. θ^l is a parameter vector that attains a lower likelihood value. The log likelihood difference between the two parameter vectors is approximately 13. The standard deviations of the measurement errors are chosen to be approximately 20% of the sample standard deviation of the time series.[3] We run the filter

[3]The standard deviations for the measurement errors are: 0.1731 (output growth), 0.1394 (consumption growth), 0.4515 (investment growth), 0.1128 (wage growth), 0.5838 (log hours), 0.1230 (inflation), 0.1653 (interest rates).

Table 7.3: Parameter Values For Likelihood Evaluation

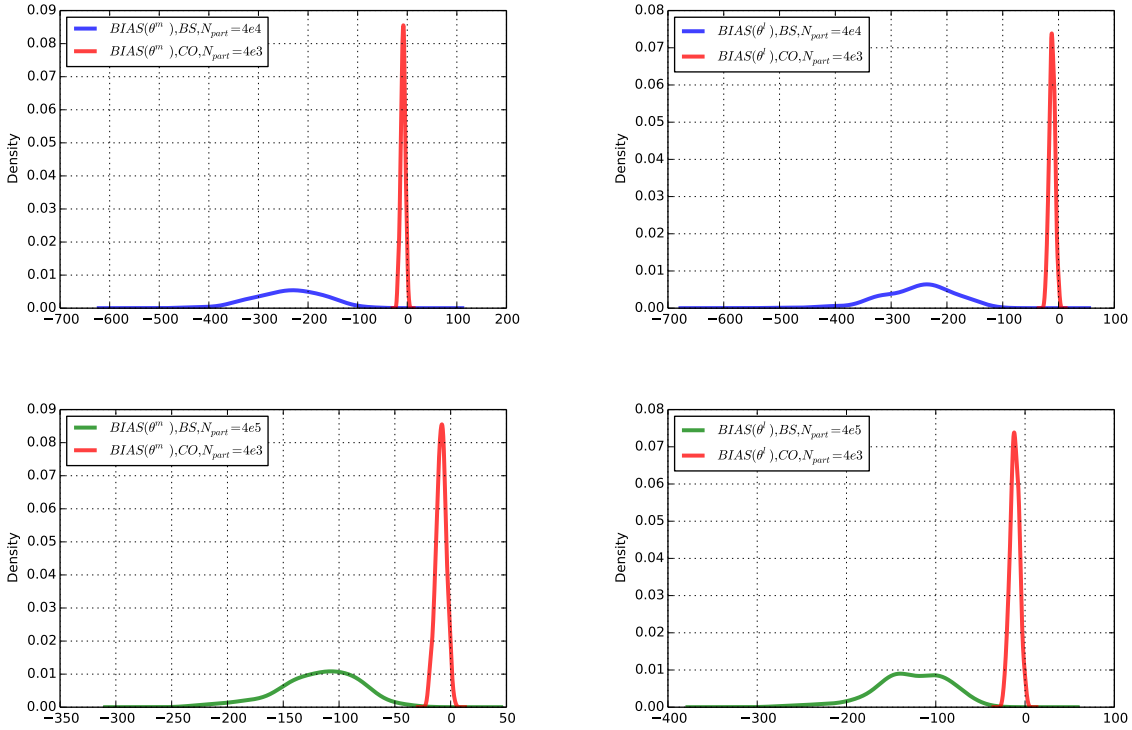
Parameter	θ^m	θ^l	Parameter	θ^m	θ^l
$100(\beta^{-1}-1)$	0.159	0.182	$\bar{\pi}$	0.774	0.571
$\bar{l}$	-1.078	0.019	α	0.181	0.230
σ	1.016	1.166	Φ	1.342	1.455
φ	6.625	4.065	h	0.597	0.511
ξ_w	0.752	0.647	σ_l	2.736	1.217
ξ_p	0.861	0.807	ι_w	0.259	0.452
ι_p	0.463	0.494	ψ	0.837	0.828
r_π	1.769	1.827	ρ	0.855	0.836
r_y	0.090	0.069	$r_{\Delta y}$	0.168	0.156
ρ_a	0.982	0.962	ρ_b	0.868	0.849
ρ_g	0.962	0.947	ρ_i	0.702	0.723
ρ_r	0.414	0.497	ρ_p	0.782	0.831
ρ_w	0.971	0.968	ρ_{ga}	0.450	0.565
μ_p	0.673	0.741	μ_w	0.892	0.871
σ_a	0.375	0.418	σ_b	0.073	0.075
σ_g	0.428	0.444	σ_i	0.350	0.358
σ_r	0.144	0.131	σ_p	0.101	0.117
σ_w	0.311	0.382	$\ln p(Y\|\theta)$	-943.00	-956.06

over the period 1966:Q1 to 2004:Q4.

Figure 7.5 depicts density plots of the log likelihood discrepancy $\hat{\Delta}_1$ for the bootstrap PF and the conditionally-optimal PF. A comparison to Figure 7.2 highlights that the accuracy of the PF deteriorates substantially by moving from a small-scale DSGE model to a medium-scale DSGE model. The results depicted in the top row of Figure 7.5 are based on 40,000 particles for the bootstrap particle filter, which is the same number of particles used for the small-scale DSGE model. According to Table 7.4, the bias of $\hat{\Delta}_1$ at θ^m is -238.49 and the standard deviation is 68.28. The corresponding sample moments obtained for the small-scale model are -1.39 and 2.03.

Increasing the number of particles from 40,000 to 400,000 improves the accuracy of the filter somewhat as shown in the bottom row of Figure 7.5, but also increases the computational

Figure 7.5: Bias of Likelihood Approximation



Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y|\theta) - \ln p(Y|\theta)$. (1,1) θ^m: BS-PF (40,000) versus CO-PF (4,000); (1,2) θ^l: BS-PF (40,000) versus CO-PF (4,000); (2,1) θ^m: BS-PF (400,000) versus CO-PF (4,000); (2,2) θ^l: BS-PF (400,000) versus CO-PF (4,000).

time. For the conditionally-optimal PF we used 4,000 particles which is 10 times more than for the small-scale DSGE model. Compared to the bootstrap PF, there is a substantial gain from using the refined proposal distribution. According to Table 7.4 the small-sample bias of $\hat{\Delta}_1$ drops by more than a factor of 20 and the standard deviation is reduced by more than a factor of 15 relative to the bootstrap PF with 40,000 particles. Unlike for the small-scale DSGE model, the likelihood approximation of the conditionally-optimal PF appears to be biased in the small sample: the means of $\hat{\Delta}_2$ are -0.87 and -0.97 for θ^m and θ^l, respectively.

The left panel of Figure 7.6 plots the filtered exogenous shock process $\hat{g}_t$ from a single run of the Kalman filter, the bootstrap PF, and the conditionally-optimal PF. In the first half of the sample, the conditionally-optimal PF tracks $\mathbb{E}[\hat{g}_t|Y_{1:t}]$ very closely. In the early 1980s a gap between the conditionally-optimal PF approximation and the true mean of $\hat{g}_t$ opens up and for a period of about 40 quarters, the bootstrap PF approximation follows the

Table 7.4: Summary Statistics for Particle Filters, SW Model

	Bootstrap		Cond. Opt.	
Number of Particles M	40,000	400,000	4,000	40,000
Number of Repetitions	100	100	100	100
High Posterior Density: $\theta = \theta^m$				
Bias $\hat{\Delta}_1$	-238.49	-118.20	-8.55	-2.88
StdD $\hat{\Delta}_1$	68.28	35.69	4.43	2.49
Bias $\hat{\Delta}_2$	-1.00	-1.00	-0.87	-0.41
Low Posterior Density: $\theta = \theta^l$				
Bias $\hat{\Delta}_1$	-253.89	-128.13	-11.48	-4.91
StdD $\hat{\Delta}_1$	65.57	41.25	4.98	2.75
Bias $\hat{\Delta}_2$	-1.00	-1.00	-0.97	-0.64

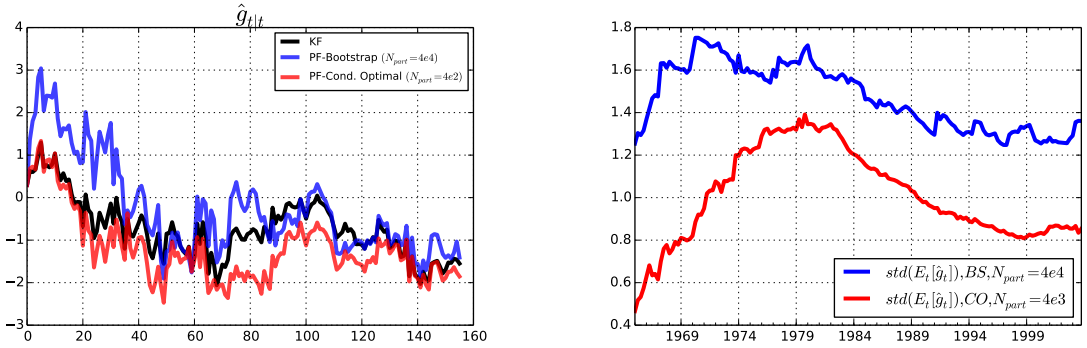
Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (7.72) and (7.73).

path of $\mathbb{E}[\hat{g}_t|Y_{1:t}]$ more closely. The right panel of the figure shows the standard deviation of the two particle filter approximations across 100 repetitions. The conditionally-optimal PF produces a more accurate approximation than the bootstrap PF, but both approximations are associated with considerable variability. For the conditionally-optimal PF, the smallest value of the standard deviation of $\hat{\mathbb{E}}[\hat{g}_t|Y_{1:t}]$ is 0.4 and the largest value is 1.4

7.7 Computational Considerations

The illustrations in Sections 7.5 and 7.6 highlighted that a careful specification of the proposal distribution in the SISR Algorithm 12 is very important. Because of the ease of implementation, the results for the bootstrap PF provide a lower bound on the accuracy of particle filter approximations for DSGE model likelihood functions, whereas the results from the conditionally-optimal PF provide an upper bound, that in applications with nonlinear DSGE models will be unattainable. As discussed in Section 7.4.2, an approximate conditionally-optimal filter could be obtained by using an extended Kalman filter or an unscented Kalman filter to construct an efficient proposal distribution. If the nonlinearities in the DSGE model are mild, then a Kalman filter updating step applied to a linearized

Figure 7.6: Filtered $\hat{g}_t$



Notes: Left panel: shows time path of the mean of $\hat{\mathbb{E}}[\hat{g}_t|Y_{1:t}]$ across 100 repetitions for KF, BS-PF (40,000), and CO-PF (4,000). Right panel shows standard deviation of $\hat{\mathbb{E}}[\hat{g}_t|Y_{1:t}]$ across 100 runs.

version of the DSGE model could be used to obtain a good proposal distribution. While the computation of efficient proposal distribution requires additional time, it makes it possible to reduce the number of particles, which can speed up the particle filter considerably.

While it is possible to parallelize the forecasting steps of the particle filter algorithms, a massive parallelization is difficult because of the high communication costs in the subsequent updating and selection steps. In fact the speed of the resampling routine may become the biggest bottleneck and it is important to use a fast routine, e.g., stratified resampling. DSGE model solutions often generate redundant state variables. In high-dimensional systems it is useful to reduce the dimension of the state vector to its minimum. This reduces the memory requirements to store the particles and it avoids numerical difficulties that may arise from singularities in the distribution of the states.

Chapter 8

Combining Particle Filters with MH Samplers

We previously focused on the particle-filter approximation of the likelihood function of a potentially nonlinear DSGE model. In order to conduct Bayesian inference, the approximate likelihood function has to be embedded into a posterior sampler. We begin by combining the particle filtering methods of Chapter 7 with the MCMC methods of Chapter 4. In a nutshell, we replace the actual likelihood functions that appear in the formula for the acceptance probability $\alpha(\vartheta|\theta^{i-1})$ in Algorithm 4 by particle filter approximations $\hat{p}(Y|\theta)$. This idea was first proposed for the estimation of nonlinear DSGE models by Fernández-Villaverde and Rubio-Ramírez (2007). We refer to the resulting algorithm as PFMH algorithm. It is a special case of a larger class of algorithms called particle Markov chain Monte Carlo (PMCMC). The theoretical properties of PMCMC methods were established in Andrieu, Doucet, and Holenstein (2010). Applications of PF-MH algorithms in other areas of econometrics are discussed in Flury and Shephard (2011).

8.1 The PFMH Algorithm

The statistical theory underlying the PFMH algorithm is very complex and beyond the scope of this book. We refer the interested reader to Andrieu, Doucet, and Holenstein (2010) for a careful exposition. Below we will sketch the main idea behind the algorithm. The exposition is based on Flury and Shephard (2011). We will distinguish between $\{p(Y|\theta), p(\theta|Y), p(Y)\}$

and $\{\hat{p}(Y|\theta), p(\theta|Y), p(Y)\}$. The first triplet consists of the exact likelihood function $p(Y|\theta)$ and the resulting posterior distribution and marginal data density defined as

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}, \quad p(Y) = \int p(Y|\theta)p(\theta)d\theta. \tag{8.1}$$

The second triplet consists of the particle filter approximation of the likelihood function denoted by $\hat{p}(Y|\theta)$ and the resulting posterior and marginal data density:

$$\hat{p}(\theta|Y) = \frac{\hat{p}(Y|\theta)p(\theta)}{\hat{p}(Y)}, \quad \hat{p}(Y) = \int \hat{p}(Y|\theta)p(\theta)d\theta. \tag{8.2}$$

By replacing the exact likelihood function $p(\theta|Y)$ with the particle filter approximation $\hat{p}(Y|\theta)$ in Algorithm 4 one might expect to obtain draws from the approximate posterior $\hat{p}(\theta|Y)$ instead of the exact posterior $p(\theta|Y)$. The surprising implication of the theory developed in Andrieu, Doucet, and Holenstein (2010) is that the distribution of draws from the PFMH algorithm that replaces $p(Y|\theta)$ by $\hat{p}(Y|\theta)$ in fact does converge to the exact posterior. The algorithm takes the following form:

Algorithm 18 (PFMH Algorithm) *For $i = 1$ to N:*

1. *Draw ϑ from a density $q(\vartheta|\theta^{i-1})$.*

2. *Set $\theta^i = \vartheta$ with probability*

$$\alpha(\vartheta|\theta^{i-1}) = \min\left\{1, \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{i-1})}{\hat{p}(Y|\theta^{j-1})p(\theta^{i-1})/q(\theta^{i-1}|\vartheta)}\right\}$$

   and $\theta^i = \theta^{i-1}$ otherwise. The likelihood approximation $\hat{p}(Y|\vartheta)$ is computed using Algorithm 12.

Any of the particle filters described in Chapter 7 could be used in the PFMH algorithm. For concreteness, we used the SISR filter described in Algorithm 12. At each iteration the filter generates draws $\tilde{s}_t^j$ from the proposal distribution $g_t(\cdot)$. Let $\tilde{S}_t = \left(\tilde{s}_t^1, \ldots, \tilde{s}_t^M\right)'$ and denote the entire sequence of draws by $\tilde{S}_{1:T}^{1:M}$. In the selection step we are using multinomial resampling to determine the ancestor for each particle in the next iteration. Thus, we can define a random variable A_t^j that contains this ancestry information. For instance, suppose that during the resampling particle $j = 1$ was assigned the value $\tilde{s}_t^{10}$ then $A_t^1 = 10$. Note that the A_t^j's are random variables whose values are determined during the selection step. Let $A_t = \left(A_t^1, \ldots, A_t^N\right)$ and use $A_{1:T}$ to denote the sequence of A_t's.

The PFMH algorithm operates on a probability space that includes the parameter vector θ as well as $\tilde{S}_{1:T}$ and $A_{1:T}$. We use $U_{1:T}$ to denote the sequence of random vectors that are used to generate $\tilde{S}_{1:T}$ and $A_{1:T}$. $U_{1:T}$ can be thought of as an array of *iid* uniform random numbers. The transformation of $U_{1:T}$ into $(\tilde{S}_{1:T}, A_{1:T})$ typically depends of θ and $Y_{1:T}$ because the proposal distribution $g_t(\tilde{s}_t|s_{t-1}^j)$ in Algorithm 12 depends both on the current observation y_t as well as the parameter vector θ which enters measurement and state-transitions equations, see (7.1).

For concreteness, consider the conditionally-optimal particle filter for a linear state-space model described in Chapter 7.4.1. Implementation of this filter requires sampling from a $N(\bar{s}_{t|t}^j, P_{t|t})$ distribution for each particle j. The mean of this distribution depends on y_t and both mean and covariance matrix depend on θ through the system matrices of the state-space representation 7.2. Draws from this distribution can in principle be obtained, by sampling *iid* uniform random variates, using a probability integral transform to convert them into *iid* draws from a standard normal distribution, and then converting them into draws from a $N(\bar{s}_{t|t}^f, P_{t|t})$. Likewise, in the selection step, the multinomial resampling could be implemented based on draws from *iid* uniform random variables. Therefore, we can express the particle filter approximation of the likelihood function as

$$\hat{p}(Y_{1:T}|\theta) = g(Y_{1:T}|\theta, U_{1:T}). \tag{8.3}$$

where

$$U_{1:T} \sim p(U_{1:T}) = \prod_{t=1}^{T} p(U_t). \tag{8.4}$$

The PFMH algorithm can be interpreted as operating on an enlarged probability space for the $(Y_{1:T}, \theta, U_{1:T})$. Define the joint distribution

$$p_g\big(Y_{1:T}, \theta, U_{1:T}\big) = g(Y_{1:T}|\theta, U_{1:T}) p\big(U_{1:T}\big) p(\theta). \tag{8.5}$$

The PFMH algorithm samples from the joint posterior

$$p_g\big(\theta, U_{1:T}|Y_{1:T}\big) \propto g(Y|\theta, U_{1:T}) p\big(U_{1:T}\big) p(\theta) \tag{8.6}$$

and discards the draws of $\big(U_{1:T}\big)$. For this procedure to be valid, it has to be the case that marginalizing the joint posterior $p_g\big(\theta, U_{1:T}|Y_{1:T}\big)$ with respect to $\big(U_{1:T}\big)$ yields the exact posterior $p(\theta|Y_{1:T})$. In other words, we require that the particle filter produces an unbiased simulation approximation of the likelihood function for all values of θ:

$$\mathbb{E}[\hat{p}(Y_{1:T}|\theta)] = \int g(Y_{1:T}|\theta, U_{1:T}) p\big(U_{1:T}\big) d\theta = p(Y_{1:T}|\theta). \tag{8.7}$$

Del Moral (2004) has shown that particle filters indeed deliver unbiased estimates of the likelihood function.

It turns out that the acceptance probability for the MH algorithm that operates on the enlarged probability space can be directly expressed in terms of the particle filter approximation $\hat{p}(Y_{1:T}|\theta)$. Omitting the $1:T$ subscript from Y and U, the proposal distribution for $(\vartheta, U^*_{1:T})$ in the MH algorithm is given by $q(\vartheta|\theta^{(i-1)})p(U^*_{1:T})$ but there is no need to keep track of the draws $(U^*_{1:T})$. The acceptance ratio for Algorithm 18 can be written as follows.

$$\begin{aligned}\alpha(\vartheta|\theta^{i-1}) &= \min\left\{1, \frac{\frac{g(Y|\vartheta,U^*)p(U^*)p(\vartheta)}{q(\vartheta|\theta^{(i-1)})p(U^*)}}{\frac{g(Y|\theta^{(i-1)},U^{(i-1)})p(U^{(i-1)})p(\theta^{(i-1)})}{q(\theta^{(i-1)}|\theta^*)p(U^{(i-1)})}}\right\} \\ &= \min\left\{1, \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{(i-1)})}{\hat{p}(Y|\theta^{(i-1)})p(\theta^{(i-1)})/q(\theta^{(i-1)}|\vartheta)}\right\}.\end{aligned} \tag{8.8}$$

The terms $p(U^*)$ and $p(U^{(i-1)})$ cancel from the expression in the first line of (8.8) and it suffices to keep track of the particle filter likelihood approximations $\hat{p}(Y|\vartheta)$ and $\hat{p}(Y|\theta^{(i-1)})$.

8.2 Application to the Small-Scale New Keynesian Model

We now apply the PFMH algorithm to the small-scale New Keynesian model, which is estimated over the period 1983:I to 2002:IV. We use the 1-block RWMH-V algorithm and combine it with the Kalman filter, the bootstrap PF, and the conditionally optimal PF. According to the theory sketched in the previous section the PFMH algorithm should accurately approximate the posterior distribution of the DSGE model parameters. Our results are based on 20 runs of each algorithm. In each run we generate 100,000 posterior draws and discard the first 50,000. As in Chapter 7.5, we use 40,000 particles for the bootstrap filter and 400 particles for the conditionally-optimal filter. A single run of the RWMH-V algorithm takes 1:30 minutes with the Kalman filter, approximately 40 minutes with the conditionally-optimal PF, and approximately 1 day with the bootstrap PF.

The results are summarized in Table 8.2. Most notably, despite the inaccurate likelihood approximation of the bootstrap PF documented in Chapter 7.5, the PFMH works remarkably well. Columns 2 to 4 of the table report posterior means which are computed by pooling the draws generated by the 20 runs of the algorithms. Except for some minor discrepancies in the posterior mean for τ and $r^{(A)}$, which are parameters with a high posterior variance, the posterior mean approximations are essentially identical for all three likelihood evaluation

Table 8.1: Accuracy of MH Approximations

	Posterior Mean (Pooled)			5th and 95th Percentiles (Pooled)			Std Dev of Means		
	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF
τ	2.64	2.63	2.68	[1.80, 3.60]	[1.79, 3.60]	[1.85, 3.71]	0.015	0.017	0.157
κ	0.82	0.82	0.83	[0.56, 0.99]	[0.56, 0.99]	[0.59, 0.99]	0.006	0.004	0.037
ψ_1	1.87	1.87	1.87	[1.50, 2.28]	[1.50, 2.28]	[1.49, 2.28]	0.009	0.01	0.085
ψ_2	0.64	0.64	0.64	[0.23, 1.22]	[0.23, 1.20]	[0.23, 1.16]	0.01	0.006	0.068
ρ_r	0.75	0.75	0.75	[0.68, 0.81]	[0.68, 0.81]	[0.68, 0.80]	0.001	0.001	0.012
ρ_g	0.98	0.98	0.98	[0.95, 1.00]	[0.96, 1.00]	[0.96, 1.00]	0.001	0	0.005
ρ_z	0.88	0.88	0.88	[0.83, 0.92]	[0.83, 0.92]	[0.83, 0.92]	0.001	0.001	0.007
$r^{(A)}$	0.44	0.44	0.46	[0.05, 0.99]	[0.05, 0.98]	[0.06, 0.98]	0.012	0.01	0.092
$\pi^{(A)}$	3.32	3.33	3.31	[2.81, 3.82]	[2.82, 3.82]	[2.81, 3.80]	0.013	0.011	0.09
$\gamma^{(Q)}$	0.59	0.59	0.59	[0.36, 0.81]	[0.37, 0.81]	[0.37, 0.82]	0.005	0.005	0.035
σ_r	0.24	0.24	0.24	[0.20, 0.29]	[0.20, 0.29]	[0.20, 0.29]	0.001	0.001	0.008
σ_g	0.68	0.68	0.68	[0.58, 0.79]	[0.58, 0.79]	[0.58, 0.79]	0.002	0.001	0.017
σ_z	0.32	0.32	0.32	[0.27, 0.38]	[0.27, 0.38]	[0.27, 0.38]	0.001	0.001	0.01
$\ln \hat{p}(Y)$	-357.10	-357.11	-361.58				0.033	0.026	1.776

Notes: Results are based on 20 runs of the PF-RWMH-V algorithm. Each run of the algorithm generates 100,000 draws and the first 50,000 are discarded. The likelihood function is computed with the Kalman filter (KF), bootstrap particle filter (BS-PF) or conditionally-optimal particle filter (CO-PF). "Pooled" means that we are pooling the draws from the 20 runs to compute posterior statistics. The BS-PF uses 40,000 particles, whereas the CO-PF uses 400 particles.

methods. There is slightly more variation in the estimated quantiles of the posterior distribution, but overall the three algorithms are in agreement. The last three columns of Table 8.2 contain the standard deviations of the posterior mean estimates across the 20 runs. Not surprisingly, the posterior sampler that is based on the bootstrap PF is the least accurate. The standard deviations are 2 to 4 times as large as for the samplers that utilize either the Kalman filter or the conditionally-optimal PF. As stressed in Section 8.1 the most important requirement for PFMH algorithms is that the particle filter approximation is unbiased – it does not have to be exact.

8.3 Application to the SW Model

We now use the PF-RWMH-V algorithm to estimate the SW model. Unlike in Chapter 6.2, where we used a more diffuse prior distribution to estimate the SW model, we now revert back to the prior originally specified by Smets and Wouters (2007). This prior is summarize in Table A-2 in the Appendix. As shown in Herbst and Schorfheide (2014), under the original prior distribution the RWMH algorithm is much better behaved than under our diffuse prior, because it leads to a posterior distribution that does not exhibit multiple modes. The estimation sample is 1966:I to 2004:IV. Using the RWMH-V algorithm, we estimate the model posteriors using the Kalman Filter and the conditionally-optimal PF.

The results are summarized in Table 8.2. Due to the computational complexity of the PF-RWMH-V algorithm, the results reported in the table are based on 10,000 instead of 100,000 draws from the posterior distribution. We used the conditionally-optimal PF with 40,000 particles and a single-block RWMH-V algorithm in which we scaled the posterior covariance matrix that served as covariance matrix of the proposal distribution by $c^2 = 0.25^2$ for the KF and $c^2 = 0.05^2$ for the conditionally-optimal PF. This leads to acceptance rates of 33% for the KF and 24% for the PF. In our experience, the noisy approximation of the likelihood function through the PF makes it necessary to reduce the variance of the proposal distribution to maintain a targeted acceptance rate. In the SW application the proposed moves using the PF approximation are about five times smaller than under the exact KF likelihood function. This increases the persistence of the Markov chain and leads to a reduction in accuracy. Because of the difference in precision of PF approximations at different points in the parameter space, the RWMH-V acceptance rates will vary much more across chains. For example, the standard deviation of the acceptance rate for the CO-PF PMCMC is 0.09, about ten times larger than for the KF runs.

While the pooled posterior means using the KF and the conditionally-optimal PF reported in Table 8.2 are very similar, the standard deviation of the means across runs is three to five times larger if the PF approximation of the likelihood function is used. Because the PF approximation of the log likelihood function is downward-biased the log marginal date density approximation obtained with the PF is much smaller than the one obtained with the KF.

Reducing the number of particles for the conditionally-optimal PF to 4,000 or switching to the bootstrap OF with 40,000 or 400,000 particles was not successful in the sense that the acceptance rate quickly dropped to zero. Reducing the variance of the proposal distribution

did not solve the problem because to obtain a nontrivial acceptance rate the step-size had to be so small that the sampler would not be able to traverse the high posterior density region of the parameter space in a reasonable amount of time. In view of the accuracy of the likelihood approximation reported in Table 7.4 this is not surprising. The PF approximations are highly volatile and even though the PF approximation is unbiased in theory, finite sample averages appear to be severely biased. In a nutshell, if the variation in the likelihood conditional on a particular value of θ is much larger than the variation that we observe along a Markov chain (evaluating the likelihood for the sequence θ^i, $i = 1, \ldots, N$) that is generated by using the exact likelihood function, the sampler easily gets stuck. Once the PF has generated a positive outlier estimate of $p(Y|\theta)$ is obtained, it becomes extremely difficult to move to a nearby θ because most of the PF evaluations underestimate $p(Y|\theta)$ and θ close to one another are unlikely to differ in likelihood by much.

8.4 Computational Considerations

We implement the PFMH algorithm on a single machine, utilizing up to twelve cores. Efficient parallelization of the algorithm is difficult, because it is challenging to parallelize MCMC algorithms and it is not profitable to use distributed memory parallelization for the filter. For the small-scale DSGE model it takes (HH:MM:SS) 30:20:33 hours to generate 100,000 parameter draws using the bootstrap PF with 40,000 particles. Under the conditionally-optimal filter we only use 400 particles, which reduces the run time to 00:39:20 minutes. Thus, with the conditionally-optimal filter, the PFMH algorithm runs about 50 times faster and delivers highly accurate approximations of the posterior means. For the SW model the computational time is substantially larger. It took 05:14:20:00 (DD:HH:MM:SS) days to generate 10,000 draws using the conditionally-optimal PF with 40,000 particles.

In practical applications with nonlinear DSGE models the conditionally-optimal PF that we used in our numerical illustrations is typically not available and has to be replaced by one of the other filters, possibly an approximately conditionally-optimal PF. Having a good understanding of the accuracy of the PF approximation is crucial. Thus, we recommend to assess the variance of the likelihood approximation at various points in the parameter space as we did in Chapters 7.5 and 7.6 and to tailor the filter until it is reasonably accurate. To put the accuracy of the filter approximation into perspective, one could compare it to the variation in the likelihood function of a linearized DSGE model fitted to the same data, along a sequence of posterior draws θ^i. If the variation in the likelihood function due to the

PF approximation is larger than the variation generated by moving through the parameter space, the PF-MH algorithm is unlikely to produce reliable results.

In general, likelihood evaluations for nonlinear DSGE models are computationally very costly. Rather than spending computational resources on tailoring the proposal density for the PF to reduce the number of particles, one can also try to lower the number of likelihood evaluations in the MH algorithm. Smith (2012) developed a PFMH algorithm based on surrogate transitions. In a nutshell the algorithm proceeds as follows. Instead of evaluating the posterior density (and thereby the DSGE model likelihood function) for every proposal draw ϑ, one first evaluates the likelihood function for an approximate model, e.g., a linearized DSGE model, or an uses a fast approximate filter, e.g., an extended Kalman filter, to obtain a likelihood value for the nonlinear model. Using the surrogate likelihood, one can compute the acceptance probability α. For ϑ's rejected in this step, one never has to execute the time-consuming PF computations. If the proposed draw ϑ is accepted in the first stage, then a second randomization that requires the evaluation of the actual likelihood is necessary to determine whether $\theta^i = \vartheta$ or $\theta^i = \theta^{i-1}$. If the surrogate transition is well tailored, then the acceptance probability in the second step is high and the overall algorithm accelerates the posterior sampler by reducing the number of likelihood evaluations for poor proposals ϑ.

Table 8.2: Accuracy of MH Approximations

	Posterior Mean (Pooled)		5th and 95th Percentiles (Pooled)		Std Dev of Means	
	KF	CO-PF	KF	CO-PF	KF	CO-PF
$(100\beta^{-1}-1)$	0.14	0.15	[0.06, 0.23]	[0.06, 0.24]	0.014	0.047
$\bar{\pi}$	0.73	0.74	[0.56, 0.93]	[0.60, 0.92]	0.026	0.082
$\bar{l}$	0.44	0.38	[-0.97, 1.89]	[-0.81, 1.62]	0.177	0.607
α	0.19	0.20	[0.17, 0.23]	[0.17, 0.22]	0.004	0.013
σ_c	1.49	1.43	[1.29, 1.70]	[1.25, 1.65]	0.019	0.099
Φ	1.47	1.45	[1.35, 1.60]	[1.32, 1.57]	0.014	0.064
φ	5.31	5.24	[3.79, 7.12]	[3.95, 6.77]	0.164	0.729
h	0.70	0.72	[0.63, 0.77]	[0.66, 0.77]	0.008	0.030
ξ_w	0.74	0.75	[0.64, 0.83]	[0.68, 0.83]	0.013	0.032
σ_l	2.27	2.24	[1.25, 3.39]	[1.39, 3.33]	0.128	0.488
ξ_p	0.71	0.71	[0.61, 0.80]	[0.61, 0.80]	0.014	0.051
ι_w	0.54	0.54	[0.38, 0.69]	[0.38, 0.72]	0.015	0.095
ι_p	0.48	0.51	[0.32, 0.64]	[0.35, 0.65]	0.016	0.092
ψ	0.46	0.44	[0.27, 0.66]	[0.26, 0.59]	0.024	0.085
r_π	2.10	2.05	[1.80, 2.39]	[1.80, 2.29]	0.030	0.125
ρ	0.80	0.80	[0.75, 0.85]	[0.76, 0.85]	0.007	0.021
r_y	0.13	0.12	[0.08, 0.18]	[0.08, 0.18]	0.008	0.024
$r_{\Delta y}$	0.21	0.22	[0.17, 0.26]	[0.17, 0.26]	0.006	0.024
ρ_a	0.96	0.96	[0.94, 0.98]	[0.94, 0.97]	0.003	0.009
ρ_b	0.21	0.21	[0.07, 0.37]	[0.07, 0.37]	0.023	0.082
ρ_g	0.97	0.97	[0.93, 0.99]	[0.92, 0.99]	0.003	0.021
ρ_i	0.71	0.71	[0.62, 0.81]	[0.60, 0.82]	0.013	0.062
ρ_r	0.53	0.52	[0.28, 0.74]	[0.28, 0.71]	0.038	0.116
ρ_p	0.81	0.81	[0.60, 0.95]	[0.65, 0.95]	0.033	0.080
ρ_w	0.94	0.94	[0.90, 0.98]	[0.90, 0.97]	0.004	0.021
ρ_{ga}	0.40	0.39	[0.16, 0.67]	[0.07, 0.61]	0.031	0.133
μ_p	0.67	0.66	[0.43, 0.85]	[0.40, 0.85]	0.038	0.120
μ_w	0.81	0.81	[0.69, 0.90]	[0.68, 0.90]	0.011	0.063
σ_a	0.33	0.35	[0.27, 0.40]	[0.26, 0.41]	0.006	0.041
σ_b	0.24	0.24	[0.20, 0.28]	[0.19, 0.28]	0.005	0.027
σ_g	0.50	0.49	[0.45, 0.56]	[0.43, 0.55]	0.008	0.036
σ_i	0.43	0.44	[0.36, 0.52]	[0.34, 0.55]	0.008	0.063
σ_r	0.14	0.14	[0.10, 0.18]	[0.10, 0.19]	0.005	0.022
σ_p	0.13	0.12	[0.10, 0.16]	[0.10, 0.15]	0.004	0.016
σ_w	0.21	0.21	[0.18, 0.25]	[0.18, 0.25]	0.003	0.017
$\ln \hat{p}(Y)$	-964.44	-1036.74			0.298	16.804

Notes: Results are based on 20 runs of the PF-RWMH-V algorithm. Each run of the algorithm generates 10,000 draws. The likelihood function is computed with the Kalman filter (KF) or conditionally-optimal particle filter (CO-PF). "Pooled" means that we are pooling the draws from the 20 runs to compute posterior statistics. The CO-PF uses 40,000 particles to compute the likelihood.

Chapter 9

Combining Particle Filters with SMC Samplers

We now combine the SMC algorithm of Chapter 5 with the particle filter approximation of the likelihood function developed in Chapter 7 to develop an SMC^2 algorithm. Reference: Chopin, Jacob, and Papaspiliopoulos (2012)

9.1 An SMC^2 Algorithm

As with the PFMH algorithm, our goal is to obtain a posterior sampler for the DSGE model parameters for settings in which the likelihood function of the DSGE model cannot be evaluated with the Kalman filter. Starting point is the SMC Algorithm 8. However, we make a number of modifications to our previous algorithm. Some of these modifications are important, others are merely made to simplify the exposition. First and foremost, we add data sequentially to the likelihood function rather than tempering the entire likelihood function: we consider the sequence of posteriors $\pi_n^D(\theta) = p(\theta|Y_{1:t_n})$, defined in (5.2, where $t_n = \lfloor \phi_n T \rfloor$. The advantage of using data tempering are that the particle filter can deliver an unbiased estimate of the incremental weight $p(Y_{t_{n-1}+1:t_n}|\theta)$ in the correction step, whereas the estimate of a concave transformation $p(Y_{1:T}|\theta)^{\phi_n - \phi_{n-1}}$ tends to be biased. Moreover, in general one has to evaluate the likelihood only for t_n observations instead of all T observations, which can speed up computations considerably.

Second, the evaluation of the incremental and the full likelihood function in the correction and mutation steps of Algorithm 8 are replaced by the evaluation of the respective

Table 9.1: Particle System for SMC^2 Sampler After Stage n

Parameter	State			
(θ_n^1, W_n^1)	$(s_{t_n}^{1,1}, \mathcal{W}_{t_n}^{1,1})$	$(s_{t_n}^{1,2}, \mathcal{W}_{t_n}^{1,2})$	$\cdots$	$(s_{t_n}^{1,M}, \mathcal{W}_{t_n}^{1,M})$
(θ_n^2, W_n^2)	$(s_{t_n}^{2,1}, \mathcal{W}_{t_n}^{2,1})$	$(s_{t_n}^{2,2}, \mathcal{W}_{t_n}^{2,2})$	$\cdots$	$(s_{t_n}^{2,M}, \mathcal{W}_{t_n}^{2,M})$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
(θ_n^N, W_n^N)	$(s_{t_n}^{N,1}, \mathcal{W}_{t_n}^{N,1})$	$(s_{t_n}^{N,2}, \mathcal{W}_{t_n}^{N,2})$	$\cdots$	$(s_{t_n}^{N,M}, \mathcal{W}_{t_n}^{N,M})$

particle filter approximations. Using the same notation as in (8.3), we write the particle approximations as

$$\hat{p}(y_{t_{n-1}+1:t_n}|Y_{1:t_{n-1}}|\theta) = g(y_{t_{n-1}+1:t_n}|Y_{1:t_{n-1}}, \theta, U_{1:t_n}), \quad \hat{p}(Y_{1:t_n}|\theta_n) = g(Y_{1:t_n}|\theta_n, U_{1:t_n}). \quad (9.1)$$

As before, $U_{1:t_n}$ is an array of *iid* uniform random variables generated by the particle filter with density $p(U_{1:t_n})$, see (8.4). The approximation of the likelihood increment also depends on the entire sequence $p(U_{1:t_n})$, because of the recursive structure of the filter: the particle approximation of $p(s_{t_{n-1}+1}|Y_{1:t_{n-1}}, \theta)$ is dependent on the particle approximation of $p(s_{t_{n-1}}|Y_{1:t_{n-1}}, \theta)$. The distribution of $U_{1:t_n}$ does neither depend on θ nor on $Y_{1:t_n}$ and can be factorized as

$$p(U_{1:t_n}) = p(U_{1:t_1})p(U_{t_1+1:t_2}) \cdots p(U_{t_{n-1}+1:t_n}). \quad (9.2)$$

To describe the particle system we follow the convention of Chapter 5 and index the parameter vector θ by the stage n of the SMC algorithm and write θ_n. The particles generated by the SMC sampler are indexed $i = 1, \ldots, N$ and the particles generated by the particle filter are indexed $j = 1, \ldots, M$. At stage n we have a particle system $\{\theta_n^i, W_n^i\}_{i=1}^N$ that represents the posterior distribution $p(\theta_n|Y_{1:t_n})$. Moreover, for each θ_n^i we have a particle system that represents the distribution $p(s_t|Y_{1:t_n}, \theta_n^i)$. To distinguish the weights used for the particle values that represent the conditional distribution of θ_t from the weights used to characterize the conditional distribution of s_t, we denote the latter by $\mathcal{W}$ instead of W. Moreover, because the distribution of the states is conditional on the value of θ, we use i, j superscripts: $\{s_t^{i,j}, \mathcal{W}_t^{i,j}\}_{j=1}^M$. The particle system can be arranged in the matrix form given in Table 9.1.

Finally, to streamline the notation used in the description of the algorithm, we assume that during each stage n exactly one observation is added to the likelihood function. Thus,

we can write θ_t instead of θ_n and $Y_{1:t}$ instead of $Y_{1:t_n}$ and the number of stages is $N_\phi = T$. Moreover, we resample the θ particles at every iteration of the algorithm (which means we do not have to keep track of the resampling indicator ρ_t) and we only use one MH step in the mutation phase.

Algorithm 19 (SMC^2)

1. **Initialization.** *Draw the initial particles from the prior:* $\theta_0^i \overset{iid}{\sim} p(\theta)$ *and* $W_0^i = 1$, $i = 1, \ldots, N$.

2. **Recursion.** *For* $t = 1, \ldots, T$,

   (a) **Correction.** *Reweight the particles from stage* $t-1$ *by defining the incremental weights*

   $$\tilde{w}_t^i = \hat{p}(y_t | Y_{1:t-1}, \theta_{t-1}^i) = g(y_t | Y_{1:t-1}, \theta_{t-1}^i, U_{1:t}^i) \tag{9.3}$$

   and the normalized weights

   $$\tilde{W}_t^i = \frac{\tilde{w}_n^i W_{t-1}^i}{\frac{1}{N}\sum_{i=1}^N \tilde{w}_t^i W_{t-1}^i}, \quad i = 1, \ldots, N. \tag{9.4}$$

   An approximation of $\mathbb{E}_{\pi_t}[h(\theta)]$ *is given by*

   $$\tilde{h}_{t,N} = \frac{1}{N}\sum_{i=1}^N \tilde{W}_t^i h(\theta_{t-1}^i). \tag{9.5}$$

   (b) **Selection.** *Resample the particles via multinomial resampling. Let* $\{\hat{\theta}_t^i\}_{i=1}^M$ *denote* M *iid draws from a multinomial distribution characterized by support points and weights* $\{\theta_{t-1}^i, \tilde{W}_t^i\}_{j=1}^M$ *and set* $W_t^i = 1$. *Define the vector of ancestors* $\mathcal{A}_t$ *with elements* $\mathcal{A}_t^i$ *by setting* $\mathcal{A}_t^i = k$ *if the ancestor of resampled particle* i *is particle* k, *that is,* $\hat{\theta}_t^i = \theta_{t-1}^k$.
   An approximation of $\mathbb{E}_{\pi_t}[h(\theta)]$ *is given by*

   $$\hat{h}_{t,N} = \frac{1}{N}\sum_{j=1}^N W_t^i h(\hat{\theta}_t^i). \tag{9.6}$$

   (c) **Mutation.** *Propagate the particles* $\{\hat{\theta}_t^i, W_t^i\}$ *via* 1 *step of an MH algorithm. The proposal distribution is given by*

   $$q(\vartheta_t^i | \hat{\theta}_t^i) p(U_{1:t}^{*i}) \tag{9.7}$$

and the acceptance ratio can be expressed as

$$\alpha(\vartheta_t^i|\hat{\theta}_t^i) = \min\left\{1, \frac{\hat{p}(Y_{1_t}|\vartheta_t^i)p(\vartheta_t^i)/q(\vartheta_t^i|\hat{\theta}_t^i)}{\hat{p}(Y_{1_t}|\hat{\theta}_t^i)p(\hat{\theta}_t^i)/q(\hat{\theta}_t^i|\vartheta_t^i)}\right\}. \tag{9.8}$$

An approximation of $\mathbb{E}_{\pi_t}[h(\theta)]$ *is given by*

$$\bar{h}_{t,N} = \frac{1}{N}\sum_{i=1}^{N} h(\theta_t^i)W_t^i. \tag{9.9}$$

3. For $t = T$ *the final importance sampling approximation of* $\mathbb{E}_\pi[h(\theta)]$ *is given by:*

$$\bar{h}_{T,N} = \sum_{i=1}^{N} h(\theta_T^i)W_T^i. \tag{9.10}$$

A formal analysis of SMC^2 algorithms is provided in Chopin, Jacob, and Papaspiliopoulos (2012). We will provide a heuristic explanation of why the algorithm correctly approximates the target posterior distribution and comment on some aspects of the implementation. At the end of iteration $t-1$ the algorithm has generated particles $\{\theta_{t-1}^i, W_{t-1}^i\}_{i=1}^N$. For each parameter value θ_{t-1}^i there is also a particle filter approximation of the likelihood function $\hat{p}(Y_{1:t-1}|\theta_{t-1}^i)$, a swarm of particles $\{s_{t-1}^{i,j}, \mathcal{W}_{t-1}^{i,j}\}_{j=1}^M$ that represents the distribution $p(s_{t-1}|Y_{1:t-1}, \theta_{t-1}^i)$ and the sequence of random vectors $U_{1:t-1}^i$ that underlies the simulation approximation of the particle filter. To gain an understanding of the algorithm it is useful to focus on the triplets $\{\theta_{t-1}^i, U_{1:t-1}^i, W_{t-1}^i\}_{i=1}^N$. Suppose that

$$\int\int h(\theta, U_{1:t-1})p(U_{1:t-1})p(\theta|Y_{1:t-1})dU_{1:t-1}d\theta \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta_{t-1}^i, U_{1:t-1}^i)W_{t-1}^i.. \tag{9.11}$$

This implies that we obtain the familiar approximation for functions $h(\cdot)$ that do not depend on $U_{1:t-1}$

$$\int h(\theta)p(\theta|Y_{1:t-1})d\theta \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta_{t-1}^i)W_{t-1}^i. \tag{9.12}$$

Correction Step. The incremental likelihood $\hat{p}(y_t|Y_{1:t-1}, \theta_{t-1}^i)$ can be evaluated by iterating the particle filter forward for one period, starting from $\{s_{t-1}^{i,j}, \mathcal{W}_{t-1}^{i,j}\}_{j=1}^M$. Using the notation in (9.1), the particle filter approximation of the likelihood increment can be written as

$$\hat{p}(y_t|Y_{1:t-1}, \theta_{t-1}^i) = g(y_t|Y_{1:t-1}, U_{1:t}^i, \theta_{t-1}^i). \tag{9.13}$$

The value of the likelihood function for $Y_{1:t}$ can be tracked recursively as follows:

$$\begin{aligned}\hat{p}(Y_{1:t}|\theta^i_{t-1}) &= \hat{p}(y_t|Y_{1:t-1},\theta^i_{t-1})\hat{p}(Y_{1:t-1}|\theta^i_{t-1}) \\ &= g(y_t|Y_{1:t},U^i_{1:t},\theta^i_{t-1})g(Y_{1:t-1}|U^i_{1:t-1},\theta^i_{t-1}) \\ &= g(Y_{1:t}|U^i_{1:t},\theta^i_{t-1}).\end{aligned} \tag{9.14}$$

The last equality follows because conditioning $g(Y_{1:t-1}|U^i_{1:t-1},\theta^i_{t-1})$ also on U_t does not change the particle filter approximation of the likelihood function for $Y_{1:t-1}$.

By induction, we can deduce from (9.11) that the Monte Carlo average $\frac{1}{N}\sum_{i=1}^N h(\theta^i_{t-1})\tilde{w}^i_t W^i_{t-1}$ approximates the following integral

$$\begin{aligned}&\int\int h(\theta)g(y_t|Y_{1:t-1},U_{1:t},\theta)p(U_{1:t})p(\theta|Y_{1:t-1})dU_{1:t}d\theta \\ &= \int h(\theta)\left[\int g(y_t|Y_{1:t-1},U_{1:t},\theta)p(U_{1:t})dU_{1:t}\right]p(\theta|Y_{1:t-1})d\theta.\end{aligned} \tag{9.15}$$

Provided that the particle filter approximation of the likelihood increment is unbiased, that is,

$$\int g(y_t|Y_{1:t-1},U_{1:t},\theta)p(U_{1:t})dU_{1:t} = p(y_t|Y_{1:t-1},\theta) \tag{9.16}$$

for each θ, we deduce that $\tilde{h}_{t,N}$ is a consistent estimator of $\mathbb{E}_{\pi_t}[h(\theta)]$.

Selection Step. The selection step Algorithm 19 is very similar to Algorithm 8. To simplify the description of the SMC^2 algorithm, we are resampling in every iteration. Moreover, we are keeping track of the ancestry information in the vector $\mathcal{A}_t$. This is important, because for each resampled particle i we not only need to know its value $\hat{\theta}^i_t$ but we also want to track the corresponding value of the likelihood function $\hat{p}(Y_{1:t}|\hat{\theta}^i_t)$ as well as the particle approximation of the state, given by $\{s^{i,j}_t, W^{i,j}_t\}$, and the set of random numbers $U^i_{1:t}$. In the implementation of the algorithm, the likelihood values are needed for the mutation step and the state particles are useful for a quick evaluation of the incremental likelihood in the correction step of iteration $t+1$ (see above). The $U^i_{1:t}$'s are not required for the actual implementation of the algorithm but are useful to provide a heuristic explanation for the validity of the algorithm.

Mutation Step. The mutation step essentially consists of one iteration of the PFMH algorithm described in Chapter 8.1. For each particle i there is a proposed value ϑ^i_t and an associated particle filter approximation $\hat{p}(Y_{1:t}|\vartheta^i_t)$ of the likelihood and sequence of random vectors $U^*_{1:t}$ drawn from the distribution $p(U_{1:t})$ in (9.2). As in (8.8), the densities $p(U^i_{1:t})$

and $p(U_{1:t}^*)$ cancel from the formula for the acceptance probability $\alpha(\vartheta_t^i|\hat{\theta}_t^i)$. For the implementation it is important to record the likelihood value as well as the particle system for the state s_t for each particle θ_t^i.

9.2 Application to the Small-Scale New Keynesian Model

We now present an application of the SMC^2 algorithm to the small-scale DSGE model. The results in this section can be compared to the results obtained in Chapter 8.2. Because the SMC^2 algorithm requires an unbiased approximation of the likelihood function, we will use data tempering instead of likelihood tempering as in Chapter 5.3. Overall, we compare the output of four algorithms: SMC^2 based on the conditionally-optimal PF; SMC^2 based on the bootstrap PF; SMC based on the Kalman filter likelihood function using data tempering; SMC based on the Kalman filter likelihood function using likelihood tempering. In order to approximate the likelihood function with the particle filter, we are using $M = 40,000$ particles for the bootstrap PF and $M = 400$ particles for the conditionally-optimal PF. The approximation of the posterior distribution is based on $N = 4,000$ particles for θ, $N_\phi = T + 1 = 81$ stages under data tempering, and $N_{blocks} = 3$ blocks for the mutation step.

Table 9.2 summarizes the results from running each algorithm 20 times. We report pooled posterior means from the output of the 20 runs as well as the standard deviation of the posterior mean approximations across the 20 runs. The results in the column labeled KF(L) are based on the Kalman filter likelihood evaluation and obtained from the same algorithm that was used in Chapter 5.3. The results in column KF(D) are also based on the Kalman filter, but the SMC algorithm uses data tempering instead of likelihood tempering. The columns CO-PF and BS-BF contain SMC^2 results based on the conditionally-optimal and the bootstrap PF, respectively. The pooled means of the DSGE model parameters computed from output of the KF(L), KF(D), and CO-PF algorithms are essentially identical. The log marginal data density approximations are less accurate than the posterior mean approximations and vary for the first three algorithms from -358.75 to -356.33.

A comparison of the standard deviations indicates that moving from likelihood tempering to data tempering leads to a deterioration of accuracy. For instance, the standard deviation of the log marginal data density increases from 0.12 to 1.19. As discussed in Chapter 5.3 in DSGE model applications it is important to use a convex tempering schedule that adds very little likelihood information in the initial stages. The implied tempering schedule of our

Table 9.2: Accuracy of SMC^2 Approximations

	Posterior Mean (Pooled)				Std Dev of Means			
	KF(L)	KF(D)	CO-PF	BS-PF	KF(L)	KF(D)	CO-PF	BS-PF
τ	2.65	2.68	2.67	2.60	0.009	0.031	0.06	0.514
κ	0.81	0.81	0.81	0.82	0.003	0.005	0.017	0.069
ψ_1	1.87	1.89	1.88	1.80	0.005	0.015	0.025	0.225
ψ_2	0.66	0.66	0.68	0.59	0.005	0.018	0.023	0.193
ρ_r	0.75	0.75	0.75	0.73	0.001	0.002	0.006	0.026
ρ_g	0.98	0.98	0.98	0.97	0	0.001	0.005	0.008
ρ_z	0.88	0.88	0.88	0.86	0.001	0.002	0.003	0.012
$r^{(A)}$	0.45	0.46	0.46	0.30	0.004	0.025	0.048	0.324
$\pi^{(A)}$	3.32	3.30	3.28	3.44	0.006	0.031	0.074	0.275
$\gamma^{(Q)}$	0.59	0.59	0.58	0.66	0.003	0.013	0.031	0.123
σ_r	0.24	0.24	0.24	0.23	0.001	0.001	0.003	0.023
σ_g	0.68	0.68	0.68	0.72	0.001	0.001	0.005	0.069
σ_z	0.32	0.32	0.32	0.36	0.001	0.001	0.004	0.04
σ_z	0.32	0.32	0.32	0.36	0.001	0.001	0.004	0.04
$\ln p(Y)$	-358.75	-357.34	-356.33	-340.47	0.120	1.191	4.374	14.49

Notes: Preliminary results. D is data tempering and L is likelihood tempering. KF is Kalman filter, CO-PF is conditionally-optimal PF, BS-PF is bootstrap PF. CO-PF and BS-PF use data tempering.

sequential estimation procedure is linear and adds a full observation in stage $n = 2$ (recall that $n = 1$ corresponds to sampling from the prior distribution). Replacing the Kalman filter evaluation of the likelihood function by the conditionally-optimal particle filter, increases the standard deviations further. Compared to KF(D) the standard deviations of the posterior mean approximations increase by factors ranging from 1.5 to 5. A comparison with Table 8.2 indicates that the SMC algorithm is more sensitive to the switch from the Kalman filter likelihood to the particle filter approximation. Using the conditionally-optimal particle filter, there seems to be no deterioration in accuracy of the RWMH algorithm. Finally, replacing the conditionally-optimal PF by the bootstrap PF leads to an additional deterioration in accuracy. Compared to KF(D) the standard deviations for the BS-PF approach are an order of magnitude larger. Nonetheless, the pooled posterior means are fairly close to those obtained from the other three algorithms.

9.3 Computational Considerations

The SMC^2 results reported in Table 9.2 are obtained by utilizing 40 processors. We parallelized the likelihood evaluations $\hat{p}(Y_{1:t}|\theta_t^i)$ for the θ_t^i particles rather than the particle filter computations for the swarms $\{s_t^{i,j}, \mathcal{W}_t^{i,j}\}_{j=1}^M$. The likelihood evaluations are computationally costly and do not require communication across processors. The run time for the SMC^2 with conditionally-optimal PF ($N = 4,000$, $M = 400$) is 23:24 minutes, where as the algorithm with bootstrap PF ($N = 4,000$ and $M = 40,000$) runs for 08:05:35 hours. The bootstrap PF performs poorly in terms of accuracy and runtime.

After running the particle filter for the sample $Y_{1:t-1}$ once could in principle save the particle swarm for the final state s_{t-1} for each θ_t^i. In the period t forecasting step, this information can then be used to quickly evaluate the likelihood increment. In our experience with the small-scale DSGE model, the sheer memory size of the objects (in the range of 10-20 gigabytes) precluded us from saving the $t-1$ state particle swarms in a distributed parallel environment in which memory transfers are costly. Instead, we re-computed the entire likelihood for $Y_{1:t}$ in each iteration.

Our sequential (data-tempering) implementation of the SMC^2 algorithm suffers from particle degeneracy in the intial stages, i.e., for small sample sizes. Instead of initially sampling from the prior distribution, one could initialize the algorithm by using an importance sampler with a student-t proposal distribution that approximates the posterior distribution obtained conditional on a small set of observations, e.g., $Y_{1:2}$ or $Y_{1:5}$, as suggested in Creal (2007).

Bibliography

Altug, S. (1989): "Time-to-Build and Aggregate Fluctuations: Some New Evidence," *International Economic Review*, 30(4), 889–920.

An, S., and F. Schorfheide (2007a): "Bayesian Analysis of DSGE Models," *Econometric Reviews*, 26(2-4), 113–172.

——— (2007b): "Bayesian Analysis of DSGE Models," *Econometric Reviews*, 26(2-4), 113–172.

Anderson, G. (2000): "A Reliable and Computationally Efficient Algorithm for Imposing the Saddle Point Property in Dynamic Models," *Manuscript*, Federal Reserve Board of Governors.

Andreasen, M. M. (2013): "Non-Linear DSGE Models and the Central Difference Kalman Filter," *Journal of Applied Econometrics*, 28(6), 929–955.

Andrieu, C., A. Doucet, and R. Holenstein (2010): "Particle Markov Chain Monte Carlo Methods," *Journal of the Royal Statistical Society Series B*, 72(3), 269–342.

Ardia, D., N. Bastürk, L. Hoogerheide, and H. K. van Dijk (2012): "A Comparative Study of Monte Carlo Methods for Efficient Evaluation of Marginal Likelihood," *Computational Statistics and Data Analysis*, 56(11), 3398–3414.

Arulampalam, S., S. Maskell, N. Gordon, and T. Clapp (2002): "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, 50(2), 174–188.

Aruoba, S. B., J. Fernández-Villaverde, and J. F. Rubio-Ramírez (2006): "Comparing Solution Methods for Dynamic Equilibrium Economies," *Journal of Economic Dynamics and Control*, 30(12), 2477–2508.

Berzuini, C., and W. Gilks (2001): "RESAMPLE-MOVE Filtering with Cross-Model Jumps," in *Sequential Monte Carlo Methods in Practice*, ed. by A. Doucet, N. de Freitas, and N. Gordon, pp. 117–138. Springer Verlag.

Bianchi, F. (2013): "Regime Switches, Agents' Beliefs, and Post-World War II U.S. Macroeconomic Dynamics," *Review of Economic Studies*, 80(2), 463–490.

Binder, M., and H. Pesaran (1997): "Multivariate Linear Rational Expectations Models: Characterization of the Nature of the Solutions and Their Fully Recursive Computation," *Econometric Theory*, 13(6), 877–888.

Blanchard, O. J., and C. M. Kahn (1980): "The Solution of Linear Difference Models under Rational Expectations," *Econometrica*, 48(5), 1305–1312.

Cappé, O., S. J. Godsill, and E. Moulines (2007): "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," *Proceedings of the IEEE*, 95(5), 899–924.

Cappé, O., E. Moulines, and T. Ryden (2005): *Inference in Hidden Markov Models*. Springer Verlag.

Chen, R., and J. Liu (2000): "Mixture Kalman Filters," *Journal of the Royal Statistical Society Series B*, 62, 493–508.

Chib, S., and I. Jeliazkov (2001): "Marginal Likelihoods from the Metropolis Hastings Output," *Journal of the American Statistical Association*, 96(453), 270–281.

Chib, S., and S. Ramamurthy (2010): "Tailored Randomized Block MCMC Methods with Application to DSGE Models," *Journal of Econometrics*, 155(1), 19–38.

Chopin, N. (2002): "A Sequential Particle Filter for Static Models," *Biometrika*, 89(3), 539–551.

——— (2004): "Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Inference," *Annals of Statistics*, 32(6), 2385–2411.

Chopin, N., P. E. Jacob, and O. Papaspiliopoulos (2012): "SMC^2: An Efficient Algorithm for Sequential Analysis of State-Space Models," *arXiv:1101.1528*.

Christiano, L. J. (2002): "Solving Dynamic Equilibrium Models by a Methods of Undetermined Coefficients," *Computational Economics*, 20(1-2), 21–55.

Christiano, L. J., M. Eichenbaum, and C. L. Evans (2005): "Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy," *Journal of Political Economy*, 113(1), 1–45.

Creal, D. (2007): "Sequential Monte Carlo Samplers for Bayesian DSGE Models," *Unpublished Manuscript, Vrije Universitiet*.

——— (2012): "A Survey of Sequential Monte Carlo Methods for Economics and Finance," *Econometric Reviews*, 31(3), 245–296.

Curdia, V., and R. Reis (2009): "Correlated Disturbances and U.S. Business Cycles," *Working Paper*.

——— (2010): "Correlated Disturbances and U.S. Business Cycles," *Manuscript, Columbia University and FRB New York*.

Davig, T., and E. M. Leeper (2007): "Generalizing the Taylor Principle," *American Economic Review*, 97(3), 607–635.

DeJong, D. N., B. F. Ingram, and C. H. Whiteman (2000): "A Bayesian Approach to Dynamic Macroeconomics," *Journal of Econometrics*, 98(2), 203 – 223.

Del Moral, P. (2004): *Feynman-Kac Formulae*. Springer Verlag.

——— (2013): *Mean Field Simulation for Monte Carlo Integration*. Chapman & Hall/CRC.

Del Negro, M., and F. Schorfheide (2008): "Forming Priors for DSGE Models (and How it Affects the Assessment of Nominal Rigidities)," *Journal of Monetary Economics*, 55(7), 1191–1208.

——— (2013): "DSGE Model-Based Forecasting," in *Handbook of Economic Forecasting*, ed. by G. Elliott, and A. Timmermann, vol. 2, forthcoming. North Holland, Amsterdam.

Doucet, A., N. de Freitas, and N. Gordon (eds.) (2001): *Sequential Monte Carlo Methods in Practice*. Springer Verlag.

Doucet, A., and A. M. Johansen (2011): "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later," in *Handook of Nonlinear Filtering*, ed. by D. Crisan, and B. Rozovsky. Oxford University Press.

Durbin, J., and S. J. Koopman (2001): *Time Series Analysis by State Space Methods*. Oxford University Press.

Durham, G., and J. Geweke (2012): "Adaptive Sequential Posterior Simulators for Massively Parallel Computing Environments," *Unpublished Manuscript*.

Farmer, R., D. Waggoner, and T. Zha (2009): "Understanding Markov Switching Rational Expectations Models," *Journal of Economic Theory*, 144(5), 1849–1867.

Fernández-Villaverde, J., and J. F. Rubio-Ramírez (2007): "Estimating Macroeconomic Models: A Likelihood Approach," *Review of Economic Studies*, 74(4), 1059–1087.

Flury, T., and N. Shephard (2011): "Bayesian Inference Based Only On Simulated Likelihood: Particle Filter Analysis of Dynamic Economic Models," *Econometric Theory*, 27, 933–956.

Gali, J. (2008): *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework*. Princeton University Press.

Geweke, J. (1989): "Bayesian Inference in Econometric Models Using Monte Carlo Integration," *Econometrica*, 57(6), 1317–1399.

——— (1999): "Using Simulation Methods for Bayesian Econometric Models: Inference, Development, and Communication," *Econometric Reviews*, 18(1), 1–126.

——— (2005): *Contemporary Bayesian Econometrics and Statistics*. John Wiley & Sons, Inc.

Gordon, N., D. Salmond, and A. F. Smith (1993): "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *Radar and Signal Processing, IEE Proceedings F*, 140(2), 107–113.

Guo, D., X. Wang, and R. Chen (2005): "New Sequential Monte Carlo Methods for Nonlinear Dynamic Systems," *Statistics and Computing*, 15, 135.147.

Hamilton, J. D. (1994): *Time Series Analysis*. Pri.

Herbst, E. (2011): "Gradient and Hessian-based MCMC for DSGE Models," *Unpublished Manuscript, University of Pennsylvania*.

Herbst, E., and F. Schorfheide (2014): "Sequential Monte Carlo Sampling for DSGE Models," *Journal of Applied Econometrics*, forthcoming.

Ireland, P. N. (2004): "A Method for Taking Models to the Data," *Journal of Economic Dynamics and Control*, 28(6), 1205–1226.

Johnson, R. (1970): "Asymptotic Expansions Associated with Posterior Distributions," *Annals of Mathematical Statistics*, 41, 851–864.

Judd, K. (1998): *Numerical Methods in Economics*. MIT Press, Cambridge.

Justiniano, A., and G. E. Primiceri (2008): "The Time-Varying Volatility of Macroeconomic Fluctuations," *American Economic Review*, 98(3), 604–641.

King, R. G., C. I. Plosser, and S. Rebelo (1988): "Production, Growth, and Business Cycles: I The Basic Neoclassical Model," *Journal of Monetary Economics*, 21(2-3), 195–232.

King, R. G., and M. W. Watson (1998): "The Solution of Singluar Linear Difference Systems under Rational Expectations," *International Economic Review*, 39(4), 1015–1026.

Klein, P. (2000): "Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model," *Journal of Economic Dynamics and Control*, 24(10), 1405–1423.

Koenker, R. (2005): *Quantile Regression*. Cambridge University Press.

Koenker, R., and G. Bassett (1978): "Regression Quantiles," *Econometrica*, 46(1), 33–50.

Kohn, R., P. Giordani, and I. Strid (2010): "Adaptive Hybrid Metropolis-Hastings Samplers for DSGE Models," *Working Paper*.

Kollmann, R. (2014): "Tractable Latent State Filtering for Non-Linear DSGE Models Using a Second-Order Approximation and Pruning," *Computational Economics*, forthcoming.

Künsch, H. R. (2005): "Recursive Monte Carlo Filters: Algorithms and Theoretical Analysis," *Annals of Statistics*, 33(5), 1983–2021.

Kydland, F. E., and E. C. Prescott (1982): "Time to Build and Aggregate Fluctuations," *Econometrica*, 50(6), 1345–70.

Leeper, E. M., M. Plante, and N. Traum (2010): "Dynamics of Fiscaling Financing in the United States," *Journal of Econometrics*, 156, 304–321.

Liu, J. S. (2001): *Monte Carlo Strategies in Scientific Computing*. Springer Verlag.

Liu, J. S., and R. Chen (1998): "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, 93(443), 1032–1044.

Liu, J. S., R. Chen, and T. Logvinenko (2001): "A Theoretical Franework for Sequential Importance Sampling with Resampling," in *Sequential Monte Carlo Methods in Practice*, ed. by A. Doucet, N. de Freitas, and N. Gordon, pp. 225–246. Springer Verlag.

Liu, Z., D. F. Waggoner, and T. Zha (2011): "Sources of Macroeconomic Fluctuations: A Regime-switching DSGE Approach," *Quantitative Economics*, 2, 251–301.

Lubik, T., and F. Schorfheide (2003): "Computing Sunspot Equilibria in Linear Rational Expectations Models," *Journal of Economic Dynamics and Control*, 28(2), 273–285.

Müller, U. (2011): "Measuring Prior Sensitivity and Prior Informativeness in Large Bayesian Models," *Manuscript, Princeton University*.

Murray, L. M., A. Lee, and P. E. Jacob (2014): "Parallel Resampling in the Particle Filter," *arXiv Working Paper*, 1301.4019v2.

Otrok, C. (2001): "On Measuring the Welfare Costs of Business Cycles," *Journal of Monetary Economics*, 47(1), 61–92.

Phillips, D., and A. Smith (1994): "Bayesian Model Comparison via Jump Diffusions," Technical report 94-20, Imperial College of Science, Technology, and Medicine, London.

Pitt, M. K., and N. Shephard (1999): "Filtering via Simulation: Auxiliary Particle Filters," *Journal of the American Statistical Association*, 94(446), 590–599.

——— (2001): "Auxiliary Variable Based Particle Filters," in *Sequential Monte Carlo Methods in Practice*, ed. by A. Doucet, N. de Freitas, and N. Gordon, pp. 273–294. Springer Verlag.

Qi, Y., and T. P. Minka (2002): "Hessian-based Markov Chain Monte-Carlo Algorithms," *Unpublished Manuscript*.

Ríos-Rull, J.-V., F. Schorfheide, C. Fuentes-Albero, M. Kryshko, and R. Santaeulalia-Llopis (2012): "Methods versus Substance: Measuring the Effects of Technology Shocks," *Journal of Monetary Economics*, 59(8), 826–846.

Robert, C. P., and G. Casella (2004): *Monte Carlo Statistical Methods*. Springer.

Roberts, G., A. Gelman, and G. W.R. (1997): "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms," *The Annals of Applied Probability*, 7(1), 110–120.

Roberts, G., and J. S. Rosenthal (1998): "Markov-Chain Monte Carlo: Some Practical Implications of Theoretical Results," *The Canadian Journal of Statistics*, 25(1), 5–20.

Roberts, G., and O. Stramer (2002): "Langevin Diffusions and Metropolis-Hastings Algorithms," *Methodology and Computing in Applied Probability*, 4, 337–357.

Roberts, G. O., and S. Sahu (1997): "Updating Schemes, Correlation Structure, Blocking and Parameterization for the Gibbs Sampler," *Journal of the Royal Statistical Society. Series B (Methodological)*, 59(2), 291–317.

Roberts, G. O., and R. Tweedie (1992): "Exponential Convergence of Langevin Diffusions and Their Discrete Approximations," *Bernoulli*, 2, 341 – 363.

Rosenthal, J. S. (2000): "Parallel Computing and Monte Carlo Algorithms," *Far East Journal of Theoretical Statistics*, 4, 207–236.

Rotemberg, J. J., and M. Woodford (1997): "An Optimization-Based Econometric Framework for the Evaluation of Monetary Policy," in *NBER Macroeconomics Annual 1997*, ed. by B. S. Bernanke, and J. J. Rotemberg. MIT Press, Cambridge.

Sargent, T. J. (1989): "Two Models of Measurements and the Investment Accelerator," *Journal of Political Economy*, 97(2), 251–287.

Schorfheide, F. (2000): "Loss Function-based Evaluation of DSGE Models," *Journal of Applied Econometrics*, 15, 645–670.

——— (2005): "Learning and Monetary Policy Shifts," *Review of Economic Dynamics*, 8(2), 392–419.

——— (2010): "Estimation and Evaluation of DSGE Models: Progress and Challenges," *NBER Working Paper*.

Schorfheide, F., D. Song, and A. Yaron (2014): "Identifiying Long-Run Risks: A Bayesian Mixed-Frequency Approach," *NBER Working Paper*, 20303.

Sims, C. A. (2002): "Solving Linear Rational Expectations Models," *Computational Economics*, 20, 1–20.

Sims, C. A., D. Waggoner, and T. Zha (2008): "Methods for Inference in Large Multiple-Equation Markov-Switching Models," *Journal of Econometrics*, 146(2), 255–274.

Smets, F., and R. Wouters (2003): "An Estimated Dynamic Stochastic General Equilibrium Model of the Euro Area," *Journal of the European Economic Association*, 1(5), 1123–1175.

Smets, F., and R. Wouters (2007): "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 97, 586–608.

Smith, M. (2012): "Estimating Nonlinear Economic Models Using Surrogate Transitions," *Manuscript, RePEc*.

Strid, I. (2009): "Efficient Parallelisation of Metropolis-Hastings Algorithms Using a Prefetching Approach," *Computational Statistics and Data Analysis*, in press.

van der Vaart, A. (1998): *Asymptotic Statistics*. Cambridge University Press.

Woodford, M. (2003): *Interest and Prices*. Princeton University Press.

Appendix A

Model Descriptions

A.1 The Smets-Wouters Model

The equilibrium conditions of the Smets and Wouters (2007) model take the following form:

$$\hat{y}_t = c_y\hat{c}_t + i_y\hat{i}_t + z_y\hat{z}_t + \varepsilon_t^g \tag{A.1}$$

$$\hat{c}_t = \frac{h/\gamma}{1+h/\gamma}\hat{c}_{t-1} + \frac{1}{1+h/\gamma}E_t\hat{c}_{t+1} + \frac{wl_c(\sigma_c-1)}{\sigma_c(1+h/\gamma)}(\hat{l}_t - E_t\hat{l}_{t+1}) - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}(\hat{r}_t - E_t\hat{\pi}_{t+1}) - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}\varepsilon_t^b \tag{A.2}$$

$$\hat{i}_t = \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}\hat{i}_{t-1} + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}E_t\hat{i}_{t+1} + \frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})}\hat{q}_t + \varepsilon_t^i \tag{A.3}$$

$$\hat{q}_t = \beta(1-\delta)\gamma^{-\sigma_c}E_t\hat{q}_{t+1} - \hat{r}_t + E_t\hat{\pi}_{t+1} + (1-\beta(1-\delta)\gamma^{-\sigma_c})E_t\hat{r}_{t+1}^k - \varepsilon_t^b \tag{A.4}$$

$$\hat{y}_t = \Phi(\alpha\hat{k}_t^s + (1-\alpha)\hat{l}_t + \varepsilon_t^a) \tag{A.5}$$

$$\hat{k}_t^s = \hat{k}_{t-1} + \hat{z}_t \tag{A.6}$$

$$\hat{z}_t = \frac{1-\psi}{\psi}\hat{r}_t^k \tag{A.7}$$

$$\hat{k}_t = \frac{(1-\delta)}{\gamma}\hat{k}_{t-1} + (1-(1-\delta)/\gamma)\hat{i}_t + (1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i \tag{A.8}$$

$$\hat{\mu}_t^p = \alpha(\hat{k}_t^s - \hat{l}_t) - \hat{w}_t + \varepsilon_t^a \tag{A.9}$$

$$\hat{\pi}_t = \frac{\beta\gamma^{(1-\sigma_c)}}{1+\iota_p\beta\gamma^{(1-\sigma_c)}}E_t\hat{\pi}_{t+1} + \frac{\iota_p}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_{t-1} - \frac{(1-\beta\gamma^{(1-\sigma_c)}\xi_p)(1-\xi_p)}{(1+\iota_p\beta\gamma^{(1-\sigma_c)})(1+(\Phi-1)\varepsilon_p)\xi_p}\hat{\mu}_t^p + \varepsilon_t^p \tag{A.10}$$

$$\hat{r}_t^k = \hat{l}_t + \hat{w}_t - \hat{k}_t^s \tag{A.11}$$

$$\hat{\mu}_t^w = \hat{w}_t - \sigma_l\hat{l}_t - \frac{1}{1-h/\gamma}(\hat{c}_t - h/\gamma\hat{c}_{t-1}) \tag{A.12}$$

$$\hat{w}_t = \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}(E_t\hat{w}_{t+1} + E_t\hat{\pi}_{t+1}) + \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}(\hat{w}_{t-1} - \iota_w\hat{\pi}_{t-1}) - \frac{1+\beta\gamma^{(1-\sigma_c)}\iota_w}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_t - \frac{(1-\beta\gamma^{(1-\sigma_c)}\xi_w)(1-\xi_w)}{(1+\beta\gamma^{(1-\sigma_c)})(1+(\lambda_w-1)\epsilon_w)\xi_w}\hat{\mu}_t^w + \varepsilon_t^w \tag{A.13}$$

$$\hat{r}_t = \rho\hat{r}_{t-1} + (1-\rho)(r_\pi\hat{\pi}_t + r_y(\hat{y}_t - \hat{y}_t^*)) + r_{\Delta y}((\hat{y}_t - \hat{y}_t^*) - (\hat{y}_{t-1} - \hat{y}_{t-1}^*)) + \varepsilon_t^r. \tag{A.14}$$

The exogenous shocks evolve according to

$$\begin{aligned}
\varepsilon_t^a &= \rho_a \varepsilon_{t-1}^a + \eta_t^a && \text{(A.15)}\\
\varepsilon_t^b &= \rho_b \varepsilon_{t-1}^b + \eta_t^b && \text{(A.16)}\\
\varepsilon_t^g &= \rho_g \varepsilon_{t-1}^a + \rho_{ga}\eta_t^a + \eta_t^g && \text{(A.17)}\\
\varepsilon_t^i &= \rho_i \varepsilon_{t-1}^i + \eta_t^i && \text{(A.18)}\\
\varepsilon_t^r &= \rho_r \varepsilon_{t-1}^r + \eta_t^r && \text{(A.19)}\\
\varepsilon_t^p &= \rho_r \varepsilon_{t-1}^p + \eta_t^p - \mu_p \eta_{t-1}^p && \text{(A.20)}\\
\varepsilon_t^w &= \rho_w \varepsilon_{t-1}^w + \eta_t^w - \mu_w \eta_{t-1}^w. && \text{(A.21)}
\end{aligned}$$

The counterfactual no-rigidity prices and quantities evolve according to

$$\begin{aligned}
\hat{y}_t^* &= c_y \hat{c}_t^* + i_y \hat{i}_t^* + z_y \hat{z}_t^* + \varepsilon_t^g && \text{(A.22)}\\
\hat{c}_t^* &= \frac{h/\gamma}{1+h/\gamma}\hat{c}_{t-1}^* + \frac{1}{1+h/\gamma}E_t\hat{c}_{t+1}^* + \frac{wl_c(\sigma_c-1)}{\sigma_c(1+h/\gamma)}(\hat{l}_t^* - E_t\hat{l}_{t+1}^*) && \text{(A.23)}\\
&\quad -\frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}r_t^* - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}\varepsilon_t^b\\
\hat{i}_t^* &= \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}\hat{i}_{t-1}^* + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}E_t\hat{i}_{t+1}^* + \frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})}\hat{q}_t^* + \varepsilon_t^i && \text{(A.24)}\\
\hat{q}_t^* &= \beta(1-\delta)\gamma^{-\sigma_c}E_t\hat{q}_{t+1}^* - r_t^* + (1-\beta(1-\delta)\gamma^{-\sigma_c})E_t r_{t+1}^{k*} - \varepsilon_t^b && \text{(A.25)}\\
\hat{y}_t^* &= \Phi(\alpha k_t^{s*} + (1-\alpha)\hat{l}_t^* + \varepsilon_t^a) && \text{(A.26)}\\
\hat{k}_t^{s*} &= k_{t-1}^* + z_t^* && \text{(A.27)}\\
\hat{z}_t^* &= \frac{1-\psi}{\psi}\hat{r}_t^{k*} && \text{(A.28)}\\
\hat{k}_t^* &= \frac{(1-\delta)}{\gamma}\hat{k}_{t-1}^* + (1-(1-\delta)/\gamma)\hat{i}_t + (1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i && \text{(A.29)}\\
\hat{w}_t^* &= \alpha(\hat{k}_t^{s*} - \hat{l}_t^*) + \varepsilon_t^a && \text{(A.30)}\\
\hat{r}_t^{k*} &= \hat{l}_t^* + \hat{w}_t^* - \hat{k}_t^* && \text{(A.31)}\\
\hat{w}_t^* &= \sigma_l \hat{l}_t^* + \frac{1}{1-h/\gamma}(\hat{c}_t^* + h/\gamma\hat{c}_{t-1}^*). && \text{(A.32)}
\end{aligned}$$

The steady state (ratios) that appear in the measurement equation or the log-linearized equilibrium conditions are given by

$$\begin{aligned}
\gamma &= \bar{\gamma}/100 + 1 && \text{(A.33)}\\
\pi^* &= \bar{\pi}/100 + 1 && \text{(A.34)}\\
\bar{r} &= 100(\beta^{-1}\gamma^{\sigma_c}\pi^* - 1) && \text{(A.35)}\\
r^k_{ss} &= \gamma^{\sigma_c}/\beta - (1-\delta) && \text{(A.36)}\\
w_{ss} &= \left(\frac{\alpha^\alpha(1-\alpha)^{(1-\alpha)}}{\Phi r^{k\ \alpha}_{ss}}\right)^{\frac{1}{1-\alpha}} && \text{(A.37)}\\
i_k &= (1-(1-\delta)/\gamma)\gamma && \text{(A.38)}\\
l_k &= \frac{1-\alpha}{\alpha}\frac{r^k_{ss}}{w_{ss}} && \text{(A.39)}\\
k_y &= \Phi l_k^{(\alpha-1)} && \text{(A.40)}\\
i_y &= (\gamma - 1 + \delta)k_y && \text{(A.41)}\\
c_y &= 1 - g_y - i_y && \text{(A.42)}\\
z_y &= r^k_{ss}k_y && \text{(A.43)}\\
wl_c &= \frac{1}{\lambda_w}\frac{1-\alpha}{\alpha}\frac{r^k_{ss}k_y}{c_y}. && \text{(A.44)}
\end{aligned}$$

The measurement equations take the form:

$$\begin{aligned}
YGR_t &= \bar{\gamma} + \hat{y}_t - \hat{y}_{t-1} && \text{(A.45)}\\
INF_t &= \bar{\pi} + \hat{\pi}_t\\
FFR_t &= \bar{r} + \hat{R}_t\\
CGR_t &= \bar{\gamma} + \hat{c}_t - \hat{c}_{t-1}\\
IGR_t &= \bar{\gamma} + \hat{i}_t - \hat{i}_{t-1}\\
WGR_t &= \bar{\gamma} + \hat{w}_t - \hat{w}_{t-1}\\
HOURS_t &= \bar{l} + \hat{l}_t.
\end{aligned}$$

The diffuse prior distribution for the parameters of the SW model is summarized in Table A-1.

Table A-1: Diffuse Prior for SW Model

Parameter	Type	Para (1)	Para (2)	Parameter	Type	Para (1)	Para (2)
φ	Normal	4.00	4.50	α	Normal	0.30	0.15
σ_c	Normal	1.50	1.11	ρ_a	Uniform	0.00	1.00
h	Uniform	0.00	1.00	ρ_b	Uniform	0.00	1.00
ξ_w	Uniform	0.00	1.00	ρ_g	Uniform	0.00	1.00
σ_l	Normal	2.00	2.25	ρ_i	Uniform	0.00	1.00
ξ_p	Uniform	0.00	1.00	ρ_r	Uniform	0.00	1.00
ι_w	Uniform	0.00	1.00	ρ_p	Uniform	0.00	1.00
ι_p	Uniform	0.00	1.00	ρ_w	Uniform	0.00	1.00
ψ	Uniform	0.00	1.00	μ_p	Uniform	0.00	1.00
Φ	Normal	1.25	0.36	μ_w	Uniform	0.00	1.00
r_π	Normal	1.50	0.75	ρ_{ga}	Uniform	0.00	1.00
ρ	Uniform	0.00	1.00	σ_a	Inv. Gamma	0.10	2.00
r_y	Normal	0.12	0.15	σ_b	Inv. Gamma	0.10	2.00
$r_{\Delta y}$	Normal	0.12	0.15	σ_g	Inv. Gamma	0.10	2.00
π	Gamma	0.62	0.30	σ_i	Inv. Gamma	0.10	2.00
$100(\beta^{-1}-1)$	Gamma	0.25	0.30	σ_r	Inv. Gamma	0.10	2.00
l	Normal	0.00	6.00	σ_p	Inv. Gamma	0.10	2.00
γ	Normal	0.40	0.30	σ_w	Inv. Gamma	0.10	2.00

Notes: Para (1) and Para (2) correspond to the mean and standard deviation of the Beta, Gamma, and Normal distributions and to the upper and lower bounds of the support for the Uniform distribution. For the Inv. Gamma distribution, Para (1) and Para (2) refer to s and ν, where $p(\sigma|\nu,s) \propto \sigma^{-\nu-1}e^{-\nu s^2/2\sigma^2}$. The following parameters are fixed during the estimation: $\delta = 0.025$, $g_y = 0.18$, $\lambda_w = 1.50$, $\varepsilon_w = 10.0$, and $\varepsilon_p = 10$.

Table A-2: SW Model: Standard Prior

Parameter	Type	Para (1)	Para (2)	Parameter	Type	Para (1)	Para (2)
φ	Normal	4.00	1.50	α	Normal	0.30	0.05
σ_c	Normal	1.50	0.37	ρ_a	Beta	0.50	0.20
h	Beta	0.70	0.10	ρ_b	Beta	0.50	0.20
ξ_w	Beta	0.50	0.10	ρ_g	Beta	0.50	0.20
σ_l	Normal	2.00	0.75	ρ_i	Beta	0.50	0.20
ξ_p	Beta	0.50	0.10	ρ_r	Beta	0.50	0.20
ι_w	Beta	0.50	0.15	ρ_p	Beta	0.50	0.20
ι_p	Beta	0.50	0.15	ρ_w	Beta	0.50	0.20
ψ	Beta	0.50	0.15	μ_p	Beta	0.50	0.20
Φ	Normal	1.25	0.12	μ_w	Beta	0.50	0.20
r_π	Normal	1.50	0.25	ρ_{ga}	Beta	0.50	0.20
ρ	Beta	0.75	0.10	σ_a	Inv. Gamma	0.10	2.00
r_y	Normal	0.12	0.05	σ_b	Inv. Gamma	0.10	2.00
$r_{\Delta y}$	Normal	0.12	0.05	σ_g	Inv. Gamma	0.10	2.00
π	Gamma	0.62	0.10	σ_i	Inv. Gamma	0.10	2.00
$100(\beta^{-1}-1)$	Gamma	0.25	0.10	σ_r	Inv. Gamma	0.10	2.00
l	Normal	0.00	2.00	σ_p	Inv. Gamma	0.10	2.00
γ	Normal	0.40	0.10	σ_w	Inv. Gamma	0.10	2.00

Notes: Para (1) and Para (2) correspond to the mean and standard deviation of the Beta, Gamma, and Normal distributions and to the upper and lower bounds of the support for the Uniform distribution. For the Inv. Gamma distribution, Para (1) and Para (2) refer to s and ν, where $p(\sigma|\nu,s) \propto \sigma^{-\nu-1}e^{-\nu s^2/2\sigma^2}$.

The standard prior distribution for the parameters of the SW model is summarized in Table **??**

A.2 The Fiscal Policy Model

Log linear equilibrium conditions:

$$\hat{u}_t^b - \frac{\gamma(1+h)}{1-h}\hat{C}_t + \frac{\gamma h}{1-h}\hat{C}_{t-1} - \frac{\tau^c}{1+\tau^c}\hat{\tau}_t^c = \hat{R}_t - \frac{\tau^c}{1+\tau^c}E_t\hat{\tau}_{t+1}^c + E_t u_{t+1}^b - \frac{\gamma}{1-h}E_t\hat{C}_{t+1} \tag{A.46}$$

$$\hat{u}_t^l + (1+\kappa)\hat{l}_t + \frac{\tau^c}{1+\tau^c}\hat{\tau}_t^c = \hat{Y}_t - \frac{\tau^l}{1+\tau^l}\hat{\tau}_t^l - \frac{\gamma}{1-h}\hat{C}_t + \frac{\gamma h}{1-h}\hat{C}_{t-1} \tag{A.47}$$

$$\hat{q}_t = E_t\hat{u}_{t+1}^b - \frac{\gamma}{1-h}E_t\hat{C}_{t+1} + \frac{\gamma(1+h)}{1-h}\hat{C}_t - \frac{\tau^c}{1+\tau^c}E_t\tau_{t+1}^c - \hat{u}_t^b - \frac{\gamma h}{1-h}\hat{C}_{t-1} + \tag{A.48}$$
$$\frac{\tau^c}{1+\tau^c}\hat{\tau}_t^c + \beta(1-\tau^k)\alpha\frac{Y}{K}E_t\hat{Y}_{t+1} - \beta(1-\tau^k)\alpha\frac{Y}{K}\hat{K}_t - \beta\tau^k\alpha\frac{Y}{K}E_t\hat{\tau}_{t+1}^k - \beta\delta_1 E_t\hat{\nu}_{t+1}] + \beta(1-\delta_0)E_t\hat{q}_{t+1}$$

$$Y_t - \frac{\tau^k}{1-\tau^k}\hat{\tau}_t^k - \hat{K}_{t+1} = \hat{q}_t + \left(1 + \frac{\delta_2}{\delta_0}\right)\hat{\nu}_t \tag{A.49}$$

$$\frac{1}{s''(1)}\hat{q}_t + (1-\beta)\hat{I}_t + \hat{I}_{t-1} + \beta E_t\hat{u}_t^i + \beta E_t\hat{u}_{t+1}^i = 0 \tag{A.50}$$

$$Y\hat{Y}_t = C\hat{C}_t + G\hat{G}_t + I\hat{I}_t \tag{A.51}$$

$$\hat{K}_t = (1-\delta_0)K_{t-1} + \delta_1\hat{\nu}_t + \delta_0 I_t \tag{A.52}$$

$$B\hat{B}_t + \tau^k\alpha Y(\hat{\tau}_t^k + \hat{Y}_t) + \tau^l(1-\alpha)Y(\hat{\tau}_t^l + \hat{Y}_t) + \tau^c C(\hat{\tau}_t^c + \hat{C}_t) = \frac{B}{\beta}\hat{R}_{t-1} + \frac{B}{\beta}\hat{B}_{t-1} + G\hat{G}_t + Z\hat{Z}_t \tag{A.53}$$

$$\hat{Y}_t = \hat{u}_t^a + \alpha\nu_t + \alpha\hat{K}_{t-1} + (1-\alpha)\hat{L}_t. \tag{A.54}$$

Tax Processes:

$$\hat{\tau}_t^k = \varphi_k\hat{Y}_t + \gamma_k\hat{B}_{t-1} + \phi_{kl}\hat{u}_t^l + \phi_{kc}\hat{u}_t^c + \hat{u}_t^k, \tag{A.55}$$

$$\hat{\tau}_t^l = \varphi_l\hat{Y}_t + \gamma_l\hat{B}_{t-1} + \phi_{kl}\hat{u}_t^l + \phi_{lc}\hat{u}_t^c + \hat{u}_t^l, \tag{A.56}$$

$$\hat{\tau}_t^c = \phi_{kc}\hat{u}_t^c + \phi_{cl}\hat{u}_t^c + \hat{u}_t^c. \tag{A.57}$$

Table A-3: Fiscal Model: Posterior Moments - Part 2

	Based on LPT Prior		Based on Diff. Prior	
	Mean	[5%, 95%] Int.	Mean	[5%, 95%] Int.
Endogenous Propagation Parameters				
γ	2.5	[1.82, 3.35]	2.5	[1.81, 3.31]
κ	2.4	[1.70, 3.31]	2.5	[1.74, 3.37]
h	0.57	[0.46, 0.68]	0.57	[0.46, 0.67]
s''	7.0	[6.08, 7.98]	6.9	[6.06, 7.89]
δ_2	0.25	[0.16, 0.39]	0.24	[0.16, 0.37]
Endogenous Propagation Parameters				
ρ_a	0.96	[0.93, 0.98]	0.96	[0.93, 0.98]
ρ_b	0.65	[0.60, 0.69]	0.65	[0.60, 0.69]
ρ_l	0.98	[0.96, 1.00]	0.98	[0.96, 1.00]
ρ_i	0.48	[0.38, 0.57]	0.47	[0.37, 0.57]
ρ_g	0.96	[0.94, 0.98]	0.96	[0.94, 0.98]
ρ_{tk}	0.93	[0.89, 0.97]	0.94	[0.88, 0.98]
ρ_{tl}	0.98	[0.95, 1.00]	0.93	[0.86, 0.98]
ρ_{tc}	0.93	[0.89, 0.97]	0.97	[0.94, 0.99]
ρ_z	0.95	[0.91, 0.98]	0.95	[0.91, 0.98]
σ_b	7.2	[6.48, 8.02]	7.2	[6.47, 8.00]
σ_l	3.2	[2.55, 4.10]	3.2	[2.55, 4.08]
σ_i	5.7	[4.98, 6.47]	5.6	[4.98, 6.40]
σ_a	0.64	[0.59, 0.70]	0.64	[0.59, 0.70]

Appendix B

Data Sources

B.1 Small-Scale New Keynesian DSGE Model

The data from the estimation comes from **??**. Here we detail the construction of the extended sample (2003:I to 2013:IV) for 7.5.

1. **Per Capita Real Output Growth** Take the level of real gross domestic product, (`FRED` mnemonic "GDPC1"), call it GDP_t. Take the quarterly average of the Civilian Non-institutional Population (`FRED` mnemonic "CNP16OV" / BLS series "LNS10000000"), call it POP_t. Then,

$$\text{Per Capita Real Output Growth} = 100 \left[\log \left(\frac{GDP_t}{POP_t} \right) - \log \left(\frac{GDP_{t-1}}{POP_{t-1}} \right) \right].$$

2. **Annualized Inflation.** Take the CPI price level, (`FRED` mnemonic "CPIAUCSL"), call it CPI_t. Then,

$$\text{Annualized Inflation} = 400 \log \left(\frac{CPI_t}{CPI_{t-1}} \right).$$

3. **Federal Funds Rate.** Take the effective federal funds rate (`FRED` mnemonic "FEDFUNDS"), call it FFR_t. Then,

$$\text{Federal Funds Rate} = FFR_t/4.$$

B.2 Smets-Wouters Model

[FROM CHUNG, HERBST, KILEY] The data covers 1966:Q1 to 2004:Q4. The construction follows that of Smets and Wouters (2007). Output data come from the NIPA; other sources are noted in the exposition.

1. **Per Capita Real Output Growth.** Take the level of real gross domestic product, (`FRED` mnemonic "GDPC1"), call it GDP_t. Take the quarterly average of the Civilian Non-institutional Population (`FRED` mnemonic "CNP16OV" / BLS series "LNS10000000"), normalized so that it's 1992Q3 value is one, call it POP_t. Then,

$$\text{Per Capita Real Output Growth} = 100\left[\log\left(\frac{GDP_t}{POP_t}\right) - \log\left(\frac{GDP_{t-1}}{POP_{t-1}}\right)\right].$$

2. **Per Capita Real Consumption Growth.** Take the level of personal consumption expenditures (`FRED` mnemonic "PCEC"), call it $CONS_t$. Take the level of the GDP price deflator (`FRED` mnemonic "GDPDEF"), call it $GDPP_t$. Then

$$\text{Per Capita Real Consumption Growth} = 100\left[\log\left(\frac{CONS_t}{GDPP_tPOP_t}\right) - \log\left(\frac{CONS_{t-1}}{GDPP_{t-1}POP_{t-1}}\right)\right].$$

3. **Per Capita Real Investment Growth.** Take the level of fixed private investment (`FRED` mnemonic "FPI"), call it INV_t. Then,

$$\text{Per Capita Real Investment Growth} = 100\left[\log\left(\frac{INV_t}{GDPP_tPOP_t}\right) - \log\left(\frac{INV_{t-1}}{GDPP_{t-1}POP_{t-1}}\right)\right].$$

4. **Per Capita Real Wage Growth.** Take the BLS measure of compensation per hour for the nonfarm business sector (`FRED` mnemonic "COMPNFB" / BLS series "PRS85006103"), call it W_t. Then

$$\text{Per Capita Real Wage Growth} = 100\left[\log\left(\frac{W_t}{GDPP_t}\right) - \log\left(\frac{W_{t-1}}{GDPP_{t-1}}\right)\right].$$

5. **Per Capita Hours Index.** Take the index of average weekly nonfarm business hours (`FRED` mnemonic / BLS series "PRS85006023"), call it $HOURS_t$. Take the number of employed civilians (`FRED` mnemonic "CE16OV"), normalized so that its 1992Q3 value is 1, call it EMP_t. Then,

$$\text{Per Capita Hours} = 100\log\left(\frac{HOURS_tEMP_t}{POP_t}\right).$$

The series is then demeaned.

6. **Inflation.** Take the GDP price deflator, then

$$\text{Inflation} = 100 \log \left(\frac{GDPP_t}{GDPP_{t-1}} \right).$$

7. **Federal Funds Rate.** Take the effective federal funds rate (`FRED` mnemonic "FEDFUNDS"), call it FFR_t. Then,

$$\text{Federal Funds Rate} = FFR_t/4.$$

B.3 Fiscal Policy Model