

Particle MCMC and Sequential Monte Carlo Squared for DSGE Models

Edward Herbst
Federal Reserve Board

Frank Schorfheide*
*University of Pennsylvania
CEPR, and NBER*

February 13, 2015

Abstract

This paper explores the efficacy and accuracy of Bayesian estimation of nonlinear dynamic stochastic equilibrium (DSGE) models using two different approaches. First, we consider particle Markov chain Monte Carlo (PMCMC) techniques [Andrieu, Doucet, and Holenstein (2010)], wherein a particle filter-based estimate of the likelihood function is used in lieu of an exact likelihood within an MCMC algorithm. Second, we consider an alternative approach in which the parameters are estimated jointly with the states of the nonlinear model, as in Chopin, Jacob, and Papaspiliopoulos (2012). We assess the effectiveness of various configurations of these two algorithms using two DSGE models where the true likelihood is known. Specifically, we focus on how the number of particles, type of particle filter, and size of model affects the numerical efficiency of both posterior samplers. We also discuss computational considerations important for implementing both approaches.

JEL CLASSIFICATION: C11, C15, E10

KEY WORDS: Bayesian Analysis, DSGE Models, Monte Carlo Methods

*Correspondence: E. Herbst: Board of Governors of the Federal Reserve System, 20th Street and Constitution Avenue N.W., Washington, D.C. 20551. Email: edward.p.herbst@frb.gov. F. Schorfheide: Department of Economics, 3718 Locust Walk, University of Pennsylvania, Philadelphia, PA 19104-6297. Email: schorf@ssc.upenn.edu.

1 Introduction

We explore the efficacy and accuracy of Bayesian estimation of nonlinear dynamic stochastic equilibrium (DSGE) models. Bayesian estimation of such models is challenging and relatively few papers have attempted full posterior elicitation: see, for example, Fernandez-Villaverde and Rubio-Ramirez (2004) and Gust, Lopez-Salido, and Smith (2012). Therefore, there is relatively little understanding of the econometric performance previously-used methods, such as particle MCMC, and newer Sequential Monte Carlo techniques.

We assesses two posterior simulators for nonlinear state space models. First, we consider particle Markov chain Monte Carlo (PMCMC) techniques [Andrieu, Doucet, and Holenstein (2010)], wherein a particle filter-based estimate of the likelihood function is used in lieu of an exact likelihood within an MCMC algorithm. Second, we consider alternative approach in which the parameters are estimated jointly with the states of the nonlinear model, as in Chopin, Jacob, and Papaspiliopoulos (2012), using so-called the sequential Monte Carlo squared (SMC^2) algorithm. Since we are only interested in the econometrics of both simulators, we use *linear* models as our benchmarks. This allows us to compare both approaches to their Kalman filter-based equivalents, and thus isolate the loss in efficiency attributable solely to the particle approximation to the likelihood. We use a small-scale New Keynesian model and the Smets and Wouters (2007)–henceforth SW model—to get a sense of how model complexity affects the simulators.

For both approaches, the particle filter forms the basis of the estimation. In our examples, we span (loosely speaking) “best” and “worst” case scenarios, using both the naive bootstrap particle filter and a particle filter tailored with a so-called “conditionally-optimal” proposal distribution. We also vary the number of particles as well as several other tuning parameters in our examination.

Our results can be summarized as follows. For both $PMCMC$ and SMC^2 There can be substantial gain to adapting the particle filter to the particular economic model being estimated. The bootstrap particle filter performs very poorly in most cases, requiring substantially more particles to obtain reliable likelihood estimates. This inaccuracy affects the posterior simulators: for the larger, SW model, the particle MCMC algorithm broke down completely under the bootstrap particle filter. Finally, SMC^2 seems more sensitive to the accuracy of the particle approximation than $PMCMC$. While $PMCMC$ using the conditionally-optimal particle filter perform nearly as well as their Kalman Filter-based counterparts, the same is not true for SMC^2 , which is clearly less efficient than a Kalman Filter-based SMC algorithm.

The material that follows is abstracted from the yet-to-be-published book monograph, “Bayesian Estimation of DSGE Models.” The discussion of the particle MCMC and Sequential Monte Carlo methods build off some of the earlier material in the book, so we include relevant sections here for completeness. We provide a rough table of contents for navigating the document below.

- **Chapter 1.** Describes the small-scale New Keynesian model and the SW model used in the simulations.
- **Chapter 4.** Describes the basics of the random walk Metropolis-Hastings algorithm that is the backbone of the particle MCMC method.
- **Chapter 5.** Describes the basic Sequential Monte Carlo (SMC) method for estimating model parameters that forms the basis for the SMC^2 method.
- **Chapter 8.** Introduces the particle filter for evaluating the likelihood of nonlinear state space models. We define and describe the bootstrap version of the particle filter as well as one based on “conditionally optimal” proposals.
- **Chapter 9.** Describes the PMCMC algorithm and shows posterior simulation results for small-scale new Keynesian model and the SW model.
- **Chapter 10.** Describes the SMC^2 algorithm and shows posterior simulation results for the small-scale New Keynesian model.

Bayesian Estimation of DSGE Models

Edward Herbst
Frank Schorfheide

February 5, 2015

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 aggregate variables such as total factor productivity or nominal interest rates set by a central bank. This uncertainty is generated by exogenous stochastic processes that may shift technology 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 likelihood function, that is, a joint probability distribution for the endogenous model variables such as output, consumption, investment, and inflation that depends on the structural parameters of the model. These structural parameters characterize agents' preferences, production technologies, and the law of motion of the exogenous shocks. 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 analytically. 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 (2007). The model economy consists of final good producing firms, intermediate goods producing firms, households, a central bank, and a fiscal authority. We will first describe the decision problems of these agents, then describe the law of motion of the exogenous processes, and finally summarize the equilibrium conditions. 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 in this book.

1.1.1 Firms

Production takes place in two stages. There are monopolistically-competitive intermediate goods producing firms and perfectly competitive final goods producing firms that aggregate the intermediate goods into a single good that is used for household and government consumption. This two-stage production process makes it fairly straightforward to introduce price stickiness, which in turn creates a real effect of monetary policy.

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

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

The final good producers take input prices $P_t(j)$ and output prices P_t as given. The revenue from the sale of the final good is $P_t Y_t$ and the input costs incurred to produce Y_t are $\int_0^1 P_t(j) Y_t(j) dj$. Maximization of profits

$$\Pi_t = P_t \left(\int_0^1 Y_t(j)^{1-\nu} dj \right)^{\frac{1}{1-\nu}} - \int_0^1 P_t(j) Y_t(j) dj, \quad (1.2)$$

with respect to the inputs $Y_t(j)$ implies that the demand for intermediate good j is given by

$$Y_t(j) = \left(\frac{P_t(j)}{P_t} \right)^{-1/\nu} Y_t. \quad (1.3)$$

Thus, the parameter $1/\nu$ represents the elasticity of demand for each intermediate good. In the absence of an entry cost final good producers will enter the market until profits are equal to zero. From the zero-profit condition, it is possible to derive the following relationship between the intermediate goods prices and the price of the final good:

$$P_t = \left(\int_0^1 P_t(j)^{\frac{\nu-1}{\nu}} dj \right)^{\frac{\nu}{\nu-1}}. \quad (1.4)$$

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

$$Y_t(j) = A_t N_t(j), \quad (1.5)$$

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 . To keep the model simple, we abstract from capital as a factor or production for now. Labor is hired in a perfectly competitive factor market at the real wage W_t .

In order to introduce nominal price stickiness, we assume that firms face quadratic price adjustment costs

$$AC_t(j) = \frac{\phi}{2} \left(\frac{P_t(j)}{P_{t-1}(j)} - \pi \right)^2 Y_t(j), \quad (1.6)$$

where ϕ governs the price rigidity in the economy and π is the steady state inflation rate associated with the final good. Under this adjustment cost specification it is costless to change prices at the rate π . If the price change deviates from π the firm incurs a cost in terms of lost output that is a quadratic function of the discrepancy between the price change and π . The larger the adjustment cost parameter ϕ , the more reluctant the intermediate goods producers are to change their prices and the more rigid the prices are at the aggregate

level. Firm j chooses its labor input $N_t(j)$ and the price $P_t(j)$ to maximize the present value of future profits

$$\mathbb{E}_t \left[\sum_{s=0}^{\infty} \beta^s Q_{t+s|t} \left(\frac{P_{t+s}(j)}{P_{t+s}} Y_{t+s}(j) - W_{t+s} N_{t+s}(j) - AC_{t+s}(j) \right) \right]. \quad (1.7)$$

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 consumption C_t relative to a habit stock (which is approximated by the level of technology A_t)¹ and real money balances M_t/P_t . The household derives disutility from hours worked H_t and maximizes

$$\mathbb{E}_t \left[\sum_{s=0}^{\infty} \beta^s \left(\frac{(C_{t+s}/A_{t+s})^{1-\tau} - 1}{1-\tau} + \chi_M \ln \left(\frac{M_{t+s}}{P_{t+s}} \right) - \chi_H H_{t+s} \right) \right], \quad (1.8)$$

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

$$P_t C_t + B_t + M_t + T_t = P_t W_t H_t + R_{t-1} B_{t-1} + M_{t-1} + P_t D_t + P_t SC_t, \quad (1.9)$$

where SC_t is the net cash inflow from trading a full set of state-contingent securities.

1.1.3 Monetary and Fiscal Policy

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

$$R_t = R_t^*{}^{1-\rho_R} R_{t-1}^{\rho_R} e^{\epsilon_{R,t}}, \quad (1.10)$$

¹This 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.

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

$$R_t^* = r\pi^* \left(\frac{\pi_t}{\pi^*}\right)^{\psi_1} \left(\frac{Y_t}{Y_t^*}\right)^{\psi_2}. \quad (1.11)$$

Here r is the steady state real interest rate (defined below), π_t is the gross inflation rate defined as $\pi_t = P_t/P_{t-1}$, and π^* is the target inflation rate. Y_t^* in (1.11) is the level of output that would prevail in the absence of nominal rigidities.

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

$$P_t G_t + R_{t-1} B_{t-1} + M_{t-1} = T_t + B_t + M_t. \quad (1.12)$$

1.1.4 Exogenous Processes

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

$$\ln A_t = \ln \gamma + \ln A_{t-1} + \ln z_t, \quad \text{where} \quad \ln z_t = \rho_z \ln z_{t-1} + \epsilon_{z,t}. \quad (1.13)$$

Thus, on average technology grows at the rate γ and z_t captures exogenous fluctuations of the technology growth rate. Define $g_t = 1/(1 - \zeta_t)$, where ζ_t was previously defined as the fraction of aggregate output purchased by the government. We assume that

$$\ln g_t = (1 - \rho_g) \ln g + \rho_g \ln g_{t-1} + \epsilon_{g,t}. \quad (1.14)$$

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

$$Y_t = C_t + G_t + AC_t \quad \text{and} \quad H_t = N_t. \quad (1.15)$$

Because the households have access to a full set of state-contingent claims, it turns out that $Q_{t+s|t}$ in (1.7) is

$$Q_{t+s|t} = (C_{t+s}/C_t)^{-\tau} (A_t/A_{t+s})^{1-\tau}. \quad (1.16)$$

Thus, in equilibrium households and firms are using the same stochastic discount factor. Moreover, it can be shown that output, consumption, interest rates, and inflation have to satisfy the following optimality conditions

$$1 = \beta \mathbb{E}_t \left[\left(\frac{C_{t+1}/A_{t+1}}{C_t/A_t} \right)^{-\tau} \frac{A_t}{A_{t+1}} \frac{R_t}{\pi_{t+1}} \right] \quad (1.17)$$

$$1 = \frac{1}{\nu} \left[1 - \left(\frac{C_t}{A_t} \right)^\tau \right] + \phi(\pi_t - \pi) \left[\left(1 - \frac{1}{2\nu} \right) \pi_t + \frac{\pi}{2\nu} \right] - \phi \beta \mathbb{E}_t \left[\left(\frac{C_{t+1}/A_{t+1}}{C_t/A_t} \right)^{-\tau} \frac{Y_{t+1}/A_{t+1}}{Y_t/A_t} (\pi_{t+1} - \pi) \pi_{t+1} \right]. \quad (1.18)$$

(1.17) is the consumption Euler equation which reflects the first-order-condition with respect to the government bonds B_t . In equilibrium, the household equates the marginal utility of consuming a dollar today with the discounted marginal utility from investing the dollar, earning interest R_t and consuming it in the next period. (1.18) characterizes the profit maximizing choice of the intermediate goods producing firms. The first-order condition for the firms' problem depends on the wage W_t . We used the households' labor supply condition to replace W_t by a function of the marginal utility of consumption. In the absence of nominal rigidities ($\phi = 0$) aggregate output is given by

$$Y_t^* = (1 - \nu)^{1/\tau} A_t g_t, \quad (1.19)$$

which is the target level of output that appears in the monetary policy rule (1.11).

In Chapter 2.1 we will use a solution technique for the DSGE model that is based on a Taylor series approximation of the equilibrium conditions. A natural point around which to construct this approximation is the steady state of the DSGE model. The steady state is attained by setting the innovations $\epsilon_{R,t}$, $\epsilon_{g,t}$, and $\epsilon_{z,t}$ to zero at all times. Because technology $\ln A_t$ evolves according to a random walk with drift $\ln \gamma$, consumption and output need to be detrended for a steady state to exist. Let $c_t = C_t/A_t$ and $y_t = Y_t/A_t$, and $y_t^* = Y_t^*/A_t$. Then the steady state is given by

$$\pi = \pi^*, \quad r = \frac{\gamma}{\beta}, \quad R = r\pi^*, \quad c = (1 - \nu)^{1/\tau}, \quad \text{and} \quad y = gc = y^*. \quad (1.20)$$

Steady state inflation equals the targeted inflation rate π_* ; the real rate depends on the growth rate of the economy γ and the reciprocal of the households' discount factor β ; and

finally steady state output can be determined from the aggregate resource constraint. the nominal interest rate is determined by the Fisher equation; the dependence of the steady state consumption on ν reflects the distortion generated by the monopolistic competition among intermediate goods producers. We are now in a position to rewrite the equilibrium conditions by expressing each variable in terms of percentage deviations from its steady state value. Let $\hat{x}_t = \ln(x_t/x)$ and write

$$1 = \beta \mathbb{E}_t \left[e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{R}_t - \hat{z}_{t+1} - \hat{\pi}_{t+1}} \right] \quad (1.21)$$

$$\begin{aligned} \frac{1-\nu}{\nu \phi \pi^2} (e^{\tau \hat{c}_t} - 1) &= (e^{\hat{\pi}_t} - 1) \left[\left(1 - \frac{1}{2\nu}\right) e^{\hat{\pi}_t} + \frac{1}{2\nu} \right] \\ &\quad - \beta \mathbb{E}_t \left[(e^{\hat{\pi}_{t+1}} - 1) e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{y}_{t+1} - \hat{y}_t + \hat{\pi}_{t+1}} \right] \end{aligned} \quad (1.22)$$

$$e^{\hat{c}_t - \hat{y}_t} = e^{-\hat{y}_t} - \frac{\phi \pi^2 g}{2} (e^{\hat{\pi}_t} - 1)^2 \quad (1.23)$$

$$\hat{R}_t = \rho_R \hat{R}_{t-1} + (1 - \rho_R) \psi_1 \hat{\pi}_t + (1 - \rho_R) \psi_2 (\hat{y}_t - \hat{g}_t) + \epsilon_{R,t} \quad (1.24)$$

$$\hat{g}_t = \rho_g \hat{g}_{t-1} + \epsilon_{g,t} \quad (1.25)$$

$$\hat{z}_t = \rho_z \hat{z}_{t-1} + \epsilon_{z,t}. \quad (1.26)$$

The equilibrium law of motion of consumption, output, interest rates, and inflation has to satisfy the expectational difference equations (1.21) to (1.26).

1.2 Other DSGE Models Considered in this Book

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. We will present a brief overview of these models below and provide further details as needed in Chapter 6.

1.2.1 The Smets-Wouters Model

The Smets and Wouters (2007) model is a more elaborate version of the small-scale DSGE model presented in the previous section. In the SW model capital is a factor of intermediate goods production, and in addition to price stickiness the model features nominal 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 (the publication dates are misleading), Smets and Wouters (2003) augmented the Christiano-Eichenbaum-Evans model by additional exogenous structural shocks (among them price markup shocks, wage markup shocks, preference shocks, and others) 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 has been highly influential, not just in academic circles but also in central banks because it demonstrated that a modern DSGE model that is usable 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. By now, the SW model has become one of the workhorse models in the DSGE model literature and in central banks around the world. It forms the core of most large-scale DSGE model that augment the SW model with additional features such as a multi-sector production structure or financial frictions. Because of its widespread use, we will consider its estimation in this book.

1.2.2 A DSGE Model For the Analysis of Fiscal Policy

In the small-scale New Keynesian DSGE model and in the SW model fiscal policy is passive and non-distortionary. The level of government spending as a fraction of GDP is assumed to evolve exogenously and an implicit money demand equation determines the amount of seignorage generated by the interest rate feedback rule. Fiscal policy is passive in the sense that the government raises lump-sum taxes (or distributes lump-sum transfers) to ensure that its budget constraint is satisfied in every period. These lump-sum taxes are non-distortionary, because they do not affect the decisions of households and firms. The exact magnitude of the lump-sum taxes and the level of government debt are not uniquely determined, but they also do not matter for macroeconomic outcomes. Both the small-scale DSGE model and the

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 5 introduced in Chapter 3.5 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 introduce a benchmark RWMH algorithm in Section 4.1 and apply it to the small-scale New Keynesian DSGE model in Section 4.2. The DSGE model likelihood function in combination with the prior distribution presented in Chapter 2.3 leads to posterior distribution that has a fairly regular elliptical shape. In turn, the draws from a simple RWMH algorithm can be used to obtain an accurate numerical approximation of posterior moments.

Unfortunately, in many other 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 or misspecification. The DSGE model may suffer from a local identification problem that generates a posterior that is very flat in certain directions of the parameter space, similar to the posterior encountered in the simple set-identified model of Chapter 3.3. 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 simulators 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

$$q(\cdot|\theta^{i-1}) = p_t(\cdot|\mu(\theta^{i-1}), \Sigma(\theta^{i-1}), \nu), \quad (4.1)$$

where $p_t(\cdot)$ refers to the density of a student- t distribution. Thus, our exploration of proposal densities concentrates 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 Chapter 3.5, 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 the parameters can drastically improve the accuracy of Monte Carlo approximations. Finally, Section 4.6 takes a brief look at the numerical approximation of marginal data densities that are used to compute posterior model probabilities.

In Chapter 3.5, 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.

4.1 A Benchmark Algorithm

The most widely-used MH algorithm for DSGE model applications is the *random walk MH* (RWMH) algorithm. The mean of the proposal distribution in (4.1) is simply the current

location in the chain and its variance is prespecified:

$$\mu(\theta^{i-1}) = \theta^{i-1} \text{ and } \Sigma(\theta^{i-1}) = c^2 \hat{\Sigma} \quad (4.2)$$

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 η 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 $\nu = \infty$ in (4.1), that is, we use a multivariate normal proposal distribution in keeping with most of the literature. Typically, the choice of 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 Chapter 3.5, 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 θ , comprises 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, e.g., the matrix $\hat{\Sigma}$ is diagonal, then the draw ϑ is unlikely to reflect the fact that if β is large then δ should also be large, and vice versa. Therefore, $p(\vartheta|Y)$ is likely to be smaller than $p(\theta^{i-1}|Y)$, and so the proposed draw will be rejected with high probability. As a consequence, the chain will have a high rejection rate, exhibit a high autocorrelation, and the Monte Carlo estimates derived from it will have a 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 be difficult. 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 a 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 $T \rightarrow \infty$. 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 8 below). In both 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 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 eventually to guarantee the convergence of the posterior simulator. Samplers which constantly (or automatically) adjust $\hat{\Sigma}$ are known as adaptive samplers and require substantially more elaborate theoretical justifications.

Instead of strictly following one of the two approaches of tuning $\hat{\Sigma}$ that we just described, we use an estimate of the posterior covariance, $\mathbb{V}_\pi[\theta]$, obtained from an earlier estimation in the subsequent numerical illustrations. While this approach is impractical in empirical work, it is useful for the purpose of comparing the performance of different posterior samplers, because it avoids a distortion due to a mismatch between the Hessian-based estimate and the posterior covariance. Thus, it is a best-case scenario for the algorithm. To summarize, we examine the following variant of the RWMH algorithm:

$$\text{RWMH-V} \quad : \quad \hat{\Sigma} = \mathbb{V}_\pi[\theta].$$

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 large serial correlation and a high variance of the resulting Monte Carlo estimates. Similarly, if the chain rejects too frequently, it may get stuck in one region of the parameter space, again resulting in accurate estimates. However, for the special case of a target distribution which is multivariate normal, Roberts, Gelman, and Gilks (1997) have 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 factor c can be tuned during the burn-in period or via pre-estimation

Chapter 5

Sequential Monte Carlo Methods

Importance sampling has been rarely used in DSGE model application. A key difficulty with Algorithm 4, in particular in high-dimensional parameter spaces, is to find a good proposal density. In this chapter, we will explore methods in which proposal densities are constructed sequentially. Suppose ϕ_n , $n = 1, \dots, N_\phi$, is a sequence that slowly increases from zero to one. We can define a sequence of tempered posteriors as

$$\pi_n(\theta) = \frac{[p(Y|\theta)]^{\phi_n} p(\theta)}{\int [p(Y|\theta)]^{\phi_n} p(\theta) d\theta} \quad n = 0, \dots, N_\phi, \quad \phi_n \uparrow 1. \quad (5.1)$$

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 8. Chopin (2002) showed how to adapt the particle filtering techniques to conduct posterior inference for a static parameter vector. Textbook treatments of SMC algorithms can be found, for instance, in Liu (2001) and Cappé, Moulines, and Ryden (2005). The volume by Doucet, de Freitas, and Gordon (2001) discusses many applications and practical aspects of SMC. Creal (2012) provides a recent survey focusing on SMC applications in econometrics.

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 exposition in this chapter 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 (2014), is that SMC is much more amenable to parallelization. Durham and Geweke (2014) show how to implement a 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.

We present a generic SMC algorithm in Section 5.1. Further details on the implementation of the algorithm, the adaptive choice of tuning constants, and the convergence of the Monte Carlo approximations constructed from the output of the algorithm are provided in Section 5.2. Finally, we apply the algorithm to the small-scale New Keynesian DSGE model in Section 5.3. 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 A Generic SMC Algorithm

Just as the basic importance sampling algorithm, SMC algorithms generate weighted draws from the sequence of posteriors $\{\pi_n\}_{n=1}^{N_\phi}$ in (5.1). 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

$$\bar{h}_{n,N} = \frac{1}{N} \sum_{i=1}^N W_n^i h(\theta_n^i) \xrightarrow{a.s.} \mathbb{E}_\pi[h(\theta_n)]. \quad (5.2)$$

Starting from stage $n - 1$ particles $\{\theta_{n-1}^i, W_{n-1}^i\}_{i=1}^N$ the algorithm proceeds in three steps, using Chopin (2004)'s terminology: *correction*, that is, reweighting the stage $n - 1$ particles to reflect the density in iteration n ; *selection*, that is, eliminating a highly uneven distribution of particle weights (degeneracy) by resampling the particles; and *mutation*, that is, propagating the particles forward using a Markov transition kernel to adapt the particle values to the stage n bridge density.

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 $p(Y_{1:\lfloor\phi_n T\rfloor}|\theta)$:

$$\pi_n^{(D)}(\theta) = \frac{p(Y_{1:\lfloor\phi_n T\rfloor})p(\theta)}{\int p(Y_{1:\lfloor\phi_n T\rfloor})p(\theta)d\theta}, \quad (5.3)$$

where $\lfloor x \rfloor$ is the largest integer that is less or equal to x . 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 also discuss the necessary adjustments for data tempering.

The algorithm provided below relies on sequences of tuning parameters. To make the exposition more transparent, we begin by assuming that these sequences are provided *ex ante*. Let $\{\rho_n\}_{n=1}^{N_\phi}$ be a sequence of zeros and ones that determine whether the particles are resampled in the selection step and let $\{\zeta_n\}_{n=1}^{\phi N}$ of tuning parameters for the Markov transition density in the mutation step. The adaptive choice of these tuning parameters will be discussed in Section 5.2.2 below.

Algorithm 8 (Generic SMC Algorithm with Likelihood Tempering)

1. **Initialization.** ($\phi_0 = 0$). Draw the initial particles from the prior: $\theta_1^i \stackrel{iid}{\sim} p(\theta)$ and $W_1^i = 1$, $i = 1, \dots, N$.

2. **Recursion.** For $n = 1, \dots, N_\phi$,

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

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

and the normalized weights

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

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

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

(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 $W_n^i = \tilde{W}_n^i$, $i = 1, \dots, N$. An approximation of $\mathbb{E}_{\pi_n}[h(\theta)]$ is given by

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

(c) Mutation. Propagate the particles $\{\hat{\theta}_i, W_n^i\}$ via N_{MH} 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

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

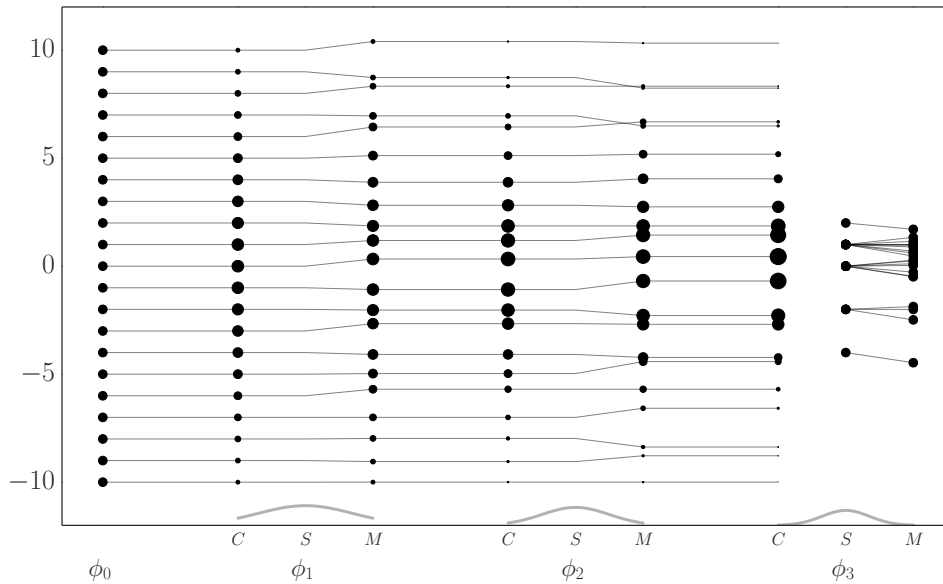
3. For $n = N_\phi$ ($\phi_{N_\phi} = 1$) the final importance sampling approximation of $\mathbb{E}_\pi[h(\theta)]$ is given by:

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

5.1.1 Three-Step Particle Propagation

A stylized representation of the propagation of the particles is depicted in Figure 5.1. Each dot corresponds to a particle and its size indicates the weight. At stage $n = 0$, $N = 21$ draws are generated from a $U[-10, 10]$ distribution and each particle receives the weight $W_0^i = 1$. At stage $n = 1$ the particles are reweighted during the correction step (the size of the dots is no longer uniform) and the particle values are modified during the mutation step (the location of the dots shifted). The bottom of the figure depicts the target posterior density. As n and ϕ_n increase, the target distribution becomes more concentrated. The concentration is mainly reflected in the increased weight of the particles with values between -3 and 3. The figure is generated under the assumption that ρ_n equals one for $n = 3$ and zero otherwise. Thus, in iteration $n = 3$ the resampling step is executed and the uneven particle weights are equalized. While the selection step generates 21 particles with equal weights, there are only six distinct particle values, four of which have multiple copies. The subsequent mutation step restores the diversity in particle values.

Figure 5.1: SMC: Evolution of Particles



Notes: The vertical location of each dot represents the particle value and the diameter of the dot represents its weight. The densities at the bottom represent the tempered target posterior $\pi_n(\cdot)$. C is Correction; S is Selection; and M is Mutation.

5.1.2 A Closer Look at the Algorithm

Algorithm 8 is initialized for $n = 0$ by generating *iid* draws from the prior distribution. This initialization works 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 $\phi_0 > 0$ and to use an initial distribution that is also informed by the tempered likelihood function $[p(Y|\theta)]^{\phi_0}$. If the particles are initialized based on a more general distribution with density $g(\theta)$, then for $n = 1$ the incremental weights have to be corrected by the ratio $p(\theta)/g(\theta)$.

The correction step reweights 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 Chapter 3.4,

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 accurate 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

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

The correction steps deliver a numerical approximation of the marginal data density as a by-product. Using arguments that we will present in more detail in Section 5.2.4 below, it can be verified that the unnormalized particle weights converge under suitable regularity conditions as follows:

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

Thus, the data density approximation is given by

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

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 resampled whenever the indicator ρ_n is equal to one. On the one hand, resampling introduces noise in the Monte Carlo approximation, which makes it undesirable. On the other hand, resampling equalizes 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 in Section 5.2.3 below.

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 reweight 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.

5.1.3 A Numerical Illustration

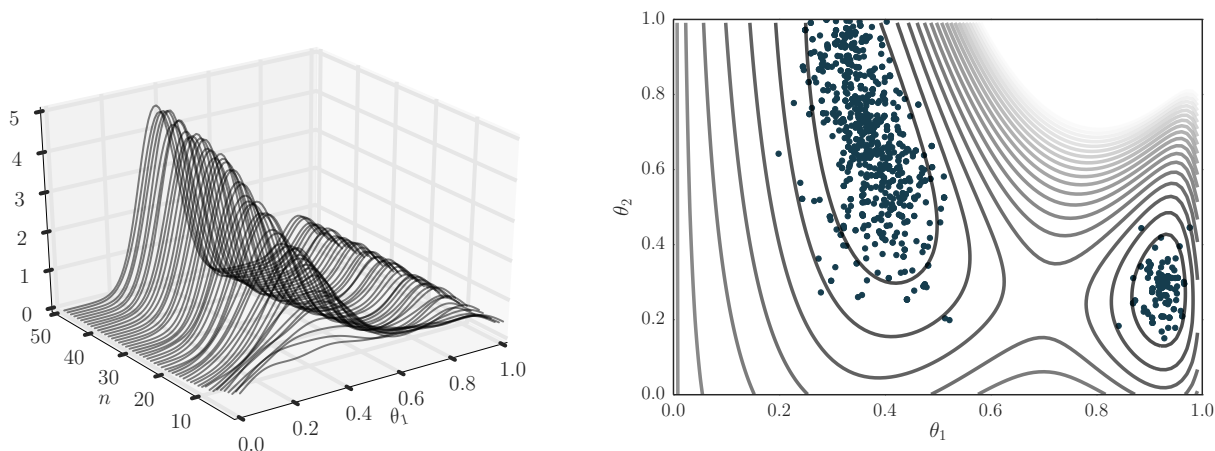
We provide a first numerical illustration of the SMC algorithm in the context of the stylized state-space model introduced in Chapter 4.3. Recall that the model is given by

$$y_t = [1 \ 1]s_t, \quad s_t = \begin{bmatrix} \theta_1^2 & 0 \\ (1 - \theta_1^2) - \theta_1\theta_2 & (1 - \theta_1^2) \end{bmatrix} s_{t-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \epsilon_t, \quad \epsilon_t \sim iidN(0, 1).$$

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 $0 \leq \theta_2 \leq 1$. Because 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 $N = 1,024$ particles, $N_\phi = 50$ stages, and a linear tempering schedule that sets $\phi_n = n/50$. The transition kernel for the mutation step is generated by a single step of the RWMH-V algorithm.

Some of the output of the SMC algorithm is depicted in Figure 5.2. The left 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 posterior is still unimodal during

Figure 5.2: SMC Posterior Approximations for Stylized State-Space Model



Notes: The left panel depicts kernel density estimates of the sequence $\pi_n(\theta)$ for $n = 0, \dots, 50$. The right panel depicts the contours of the posterior $\pi(\theta)$ as well as draws from $\pi(\theta)$.

the first few stages of the algorithm, a clear bimodal shape as emerged for $n = 10$. As ϕ_n approaches one, the bimodality of the posterior becomes more pronounced. The left 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. The right panel shows a contour plot of the joint posterior density of θ_1 and θ_2 as well as the draws from the posterior $\pi(\theta) = \pi_{N_\phi}(\theta)$ obtained in the last stage of Algorithm 8. The algorithm successfully generates draws from the two high-posterior-density regions.

5.2 Further Details of the SMC Algorithm

Our initial description of the SMC algorithm left out some details that are important for the successful implementation of the algorithm. Section 5.2.1 discusses the choice of the transition kernel in the mutation step. Section 5.2.2 considers the adaptive choice of various tuning parameters of the algorithm. 5.2.4 outlines some convergence results for Monte Carlo approximations constructed from the output of the SMC sampler. Finally, the resampling step is discussed in more detail in Section 5.2.3.

Chapter 8

Particle Filters

The key difficulty that arises when the Bayesian estimation of DSGE models is extended from linear to nonlinear models is the evaluation of the likelihood function. Throughout this book, we will focus on the use of particle filters to accomplish this task. Our starting point is a state-space representation for the nonlinear DSGE model of the form

$$\begin{aligned}y_t &= \Psi(s_t, t; \theta) + u_t, & u_t &\sim F_u(\cdot; \theta) \\s_t &= \Phi(s_{t-1}, \epsilon_t; \theta), & \epsilon_t &\sim F_\epsilon(\cdot; \theta).\end{aligned}\tag{8.1}$$

As discussed in the previous chapter, the functions $\Psi(s_t, t; \theta)$ and $\Phi(s_{t-1}, \epsilon_t; \theta)$ are generated numerically by the solution method. For reasons that will become apparent below, 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 nonlinear state-space model (8.1), the numerical illustrations and empirical applications are based on the linear Gaussian model

$$\begin{aligned}y_t &= \Psi_0(\theta) + \Psi_1(\theta)t + \Psi_2(\theta)s_t + u_t, & u_t &\sim N(0, \Sigma_u), \\s_t &= \Phi_1(\theta)s_{t-1} + \Phi_\epsilon(\theta)\epsilon_t, & \epsilon_t &\sim N(0, \Sigma_\epsilon)\end{aligned}\tag{8.2}$$

obtained by solving a log-linearized DSGE model. For model (8.2) 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). Kantas, Doucet, Singh, Maciejowski, and Chopin (2014) discuss using particle filters in the context of estimating the parameters of a state space models. 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).

The remainder of this chapter is organized as follows. We introduce the bootstrap particle filter in Section 8.1. This filter is easy to implement and has been widely used in DSGE model applications. The bootstrap filter is a special case of a sequential importance sampling with resampling (SISR) algorithm. This more general algorithm is presented in Section 8.2. An important step in the generic particle filter algorithm is to generate draws that reflect the distribution of the states in period t conditional on the information $Y_{1:t}$. These draws are generated through an importance sampling step in which states are drawn from a proposal distribution and reweighted. For the bootstrap particle filter, this proposal distribution is based on the state transition equation. Unfortunately, the forward iteration of the state transition equation might produce draws that are associated with highly variable weights, which in turn leads to imprecise Monte Carlo approximations.

The accuracy of the particle filter can be improved by choosing other proposal distributions. We discuss in Section 8.3 how to construct more efficient proposal distributions. While the tailoring (or adaption) of the proposal distributions tends to require additional computations, the number of particles can often be reduced drastically, which leads to an improvement in efficiency. DSGE model-specific implementation issues of the particle filter are examined in Section 8.4. Finally, we present the auxiliary particle filter and a filter recently proposed by DeJong, Liesenfeld, Moura, Richard, and Dharmarajan (2013) in Section 8.5. Various versions of the particle filter are applied to the small-scale New Keynesian DSGE model and the SW model in Sections 8.6 and 8.7. We close this chapter with some computational considerations in Section 8.8. Throughout this chapter we condition on a fixed vector of parameter values θ and defer the topic of parameter estimation to Chapters 9 and 10.

8.1 The Bootstrap Particle Filter

We 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 (8.1) forward. This version of the particle filter is due to Gordon, Salmond, and Smith (1993) and called 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 . A resampling step is necessary to prevent the distribution of the particle weights from degenerating. We discuss the adaptive choice of ρ_t below. The function $h(\cdot)$ is used to denote transformations of interest of the state vector s_t . The particle filter algorithm closely follows the steps of the generic filter in Algorithm 1.

Algorithm 11 (Bootstrap Particle Filter)

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

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

(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:

$$\tilde{s}_t^j = \Phi(s_{t-1}^j, \epsilon_t^j; \theta), \quad \epsilon_t^j \sim F_\epsilon(\cdot; \theta). \quad (8.3)$$

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

$$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^j) W_{t-1}^j. \quad (8.4)$$

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

$$\tilde{w}_t^j = p(y_t | \tilde{s}_t^j, \theta). \quad (8.5)$$

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

$$\hat{p}(y_t | Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j. \quad (8.6)$$

If the measurement errors are $N(0, \Sigma_u)$ then the incremental weights take the form

$$\tilde{w}_t^j = (2\pi)^{-n/2} |\Sigma_u|^{-1/2} \exp \left\{ -\frac{1}{2} (y_t - \Psi(\tilde{s}_t^j, t; \theta))' \Sigma_u^{-1} (y_t - \Psi(\tilde{s}_t^j, t; \theta)) \right\}, \quad (8.7)$$

where n here denotes the dimension of y_t .

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

$$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j}. \quad (8.8)$$

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

$$\tilde{h}_{t,M} = \frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^j) \tilde{W}_t^j. \quad (8.9)$$

(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 $W_t^j = 1$ for $j = 1, \dots, M$.

Case (ii): If $\rho_t = 0$, let $s_t^j = \tilde{s}_t^j$ and $W_t^j = \tilde{W}_t^j$ for $j = 1, \dots, M$.

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

$$\bar{h}_{t,M} = \frac{1}{M} \sum_{j=1}^M h(s_t^j) W_t^j. \quad (8.10)$$

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

$$\ln \hat{p}(Y_{1:T}|\theta) = \sum_{t=1}^T \ln \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j \right). \quad (8.11)$$

As for the SMC Algorithm 8, we can define an effective sample size (in terms of number of particles) as

$$\widehat{ESS}_t = M / \left(\frac{1}{M} \sum_{j=1}^M (\tilde{W}_t^j)^2 \right). \quad (8.12)$$

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, say $M/2$. In the remainder of this section we discuss the asymptotic properties of the particle filters, letting the number of particles $M \rightarrow \infty$, and the role of the measurement errors u_t in the bootstrap particle filter.

8.1.1 Asymptotic Properties

The convergence theory underlying the particle filter is similar to the theory sketched in Chapter 5.2.4 for the SMC sampler. As in our presentation of the SMC sampler, in the

subsequent exposition we will abstract many of the technical details that underly the convergence theory for the SMC sampler. Our exposition will be mainly heuristic, meaning that we will present some basic convergence results and the key steps for their derivation. Rigorous derivations are provided in Chopin (2004). As in Chapter 5.2.4 the asymptotic variance formulas are represented in recursive form. While this renders them unusable for the computation of numerical standard errors, they do provide some qualitative insights into the accuracy of the Monte Carlo approximations generated by the particle filter.

To simplify the notation, we drop the parameter vector θ from the conditioning set. Starting point is the assumption that Monte Carlo averages constructed from the $t - 1$ particle swarm $\{s_{t-1}^j, W_{t-1}^j\}_{j=1}^M$ satisfy an SLLN and a CLT of the form

$$\begin{aligned} \bar{h}_{t-1,M} &\xrightarrow{a.s.} \mathbb{E}[h(s_{t-1})|Y_{1:t-1}], \\ \sqrt{M}(\bar{h}_{t-1,M} - \mathbb{E}[h(s_{t-1})|Y_{1:t-1}]) &\implies N(0, \Omega_{t-1}(h)). \end{aligned} \quad (8.13)$$

Based on this assumption, we will show the convergence of $\hat{h}_{t,M}$, $\hat{p}(y_t|Y_{1:t-1})$, $\tilde{h}_{t,M}$, and $\bar{h}_{t,M}$. We write $\Omega_{t-1}(h)$ to indicate that the asymptotic covariance matrix depends on the function $h(s_t)$ for which the expected value is computed. The filter is typically initialized by directly sampling from the initial distribution $p(s_0)$, which immediately delivers the SLLN and CLT provided the required moments of $h(s_0)$ exist. We now sketch the convergence arguments for the Monte Carlo approximations in Steps 2(a) to 2(d) of Algorithm 11. 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 $\mathbb{E}_{p(\cdot|s_{t-1}^j)}[h]$ and decompose

$$\begin{aligned} \hat{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t-1}] &= \frac{1}{M} \sum_{j=1}^M \left(h(\tilde{s}_t^j) - \mathbb{E}_{p(\cdot|s_{t-1}^j)}[h] \right) W_{t-1}^j \\ &\quad + \frac{1}{M} \sum_{j=1}^M \left(\mathbb{E}_{p(\cdot|s_{t-1}^j)}[h] W_{t-1}^j - \mathbb{E}[h(s_t)|Y_{1:t-1}] \right) \\ &= I + II, \end{aligned} \quad (8.14)$$

say. This decomposition is similar to the decomposition (5.31) used in the analysis of the mutation step of the SMC algorithm.

Conditional on $\{s_{t-1}^j, W_{t-1}^j\}_{j=1}^M$ 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

$$\begin{aligned} \frac{1}{M} \sum_{j=1}^M \mathbb{E}_{p(\cdot|s_{t-1}^j)}[h]W_{t-1}^j &\xrightarrow{a.s.} \mathbb{E}[\mathbb{E}_{p(\cdot|s_{t-1})}[h] | Y_{1:t-1}] \\ &= \int \left[\int h(s_t)p(s_t|s_{t-1})ds_t \right] p(s_{t-1}|Y_{1:t-1})ds_{t-1} \\ &= \mathbb{E}[h(s_t)|Y_{1:t-1}] \end{aligned} \quad (8.15)$$

Thus, under suitable regularity conditions

$$\hat{h}_{t,M} \xrightarrow{a.s.} \mathbb{E}[h(s_t)|Y_{1:t-1}], \quad \sqrt{M}(\hat{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t-1}]) \implies N(0, \hat{\Omega}_t(h)). \quad (8.16)$$

The asymptotic covariance matrix $\hat{\Omega}_t(h)$ is given by the sum of the asymptotic variances of the terms $\sqrt{M} \cdot I$ and $\sqrt{M} \cdot II$ in (8.14). 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 (8.16) by setting $h(s_t) = p(y_t|s_t)$.

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

$$\mathbb{E}[h(s_t)|Y_{1:t}] = \frac{\int h(s_t)p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t}{\int p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t} \approx \frac{\frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^j)\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j} = \tilde{h}_{t,M}. \quad (8.17)$$

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 (5.24) in the correction step of the SMC Algorithm 8 and its convergence can be analyzed in a similar manner. Define the normalized incremental weights as

$$v_t(s_t) = \frac{p(y_t|s_t)}{\int p(y_t|s_t)p(s_t|Y_{1:t-1})ds_t}. \quad (8.18)$$

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

$$\begin{aligned} \sqrt{M}(\tilde{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t}]) \\ \implies N(0, \tilde{\Omega}_t(h)), \quad \tilde{\Omega}_t(h) = \hat{\Omega}_t(v_t(s_t)(h(s_t) - \mathbb{E}[h(s_t)|Y_{1:t}])). \end{aligned} \quad (8.19)$$

The expression for the asymptotic covariance matrix $\tilde{\Omega}_t(h)$ highlights that the accuracy depends on the distribution of the incremental weights. Roughly, the larger the variance of the particle weights, the less accurate the approximation.

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 under multinomial resampling

$$\sqrt{M}(\bar{h}_{t,M} - \mathbb{E}[h(s_t)|Y_{1:t}]) \implies N(0, \Omega_t(h)), \quad \Omega_t(h) = \tilde{\Omega}_t(h) + \mathbb{V}[h(s_t)|Y_{1:t}]. \quad (8.20)$$

As discussed in Chapter 5.2.3, the variance can be reduced by replacing the multinomial resampling with a more efficient resampling scheme.

8.1.2 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 $p(Y_{1:T}|\theta)$ is nondegenerate. It is apparent from the formulas in Table 2.1 that the Kalman filter iterations are well defined even if the measurement error covariance matrix Σ_u in the linear Gaussian state space model (8.2) 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 bootstrap particle filter the case of $\Sigma_u = 0$ presents a serious problem. The incremental weights \tilde{w}_t^j in (8.7) are degenerate if $\Sigma_u = 0$ because the conditional distribution of $y_t|(s_t, \theta)$ is a pointmass. For a particle j , this point mass is located at $y_t^j = \Psi(\tilde{s}_t^j, t; \theta)$. If the innovation ϵ_t^j is drawn from a continuous distribution in the forecasting step 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 $y_t^j = \Psi(\tilde{s}_t^j, t; \theta)$ 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 example (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 $\delta = y_t - \Psi(s_t, t; \theta)$. Suppose that in population the δ is distributed according to a $N(0, 1)$. In this case $v_t(s_t)$ in (8.18) 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 (8.19) which captures the accuracy of $\tilde{h}_{t,M}$ as well as the heuristic effective sample size measure defined in (8.12) depend on the variance of the particle weights, which in population is given by

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

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.

8.2 A Generic Particle Filter

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 leads to the following algorithm:

Algorithm 12 (Generic Particle Filter)

1. **Initialization.** (*Same as Algorithm 11*)
2. **Recursion.** For $t = 1, \dots, T$:

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

$$\omega_t^j = \frac{p(\tilde{s}_t^j|s_{t-1}^j, \theta)}{g_t(\tilde{s}_t^j|s_{t-1}^j, \theta)}. \quad (8.21)$$

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

$$\hat{h}_{t,M} = \frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^j) \omega_t^j W_{t-1}^j. \quad (8.22)$$

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

$$\tilde{w}_t^j = p(y_t|\tilde{s}_t^j, \theta) \omega_t^j. \quad (8.23)$$

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

$$\hat{p}(y_t|Y_{1:t-1}, \theta) = \frac{1}{M} \sum_{j=1}^M \tilde{w}_t^j W_{t-1}^j. \quad (8.24)$$

(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 (8.22) as well as the definition of the incremental weights \tilde{w}_t^j in (8.23). The main goal of replacing the forward iteration of the state-transition equation by an importance sampling step is to improve the accuracy of $\hat{p}(y_t|Y_{1:t-1}, \theta)$ in Step 2(b) and $\tilde{h}_{t,M}$ in Step 2(c).

We subsequently focus on the analysis of $\hat{p}(y_t|Y_{1:t-1}, \theta)$. Emphasizing the dependence of ω_t^j on both \tilde{s}_t^j and s_{t-1}^j , write

$$\begin{aligned} & \hat{p}(y_t|Y_{1:t-1}) - p(y_t|Y_{1:t-1}) \\ &= \frac{1}{M} \sum_{j=1}^M \left(p(y_t|\tilde{s}_t^j) \omega_t^j(\tilde{s}_t^j, s_{t-1}^j) - \mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[p(y_t|\tilde{s}_t^j) \omega_t^j(\tilde{s}_t^j, s_{t-1}^j)] \right) W_{t-1}^j \\ & \quad + \frac{1}{M} \sum_{j=1}^M \left(\mathbb{E}_{g_t(\cdot|s_{t-1}^j)}[p(y_t|\tilde{s}_t^j) \omega_t^j(\tilde{s}_t^j, s_{t-1}^j)] - p(y_t|Y_{1:t-1}) \right) W_{t-1}^j \\ &= I + II, \end{aligned} \quad (8.25)$$

say. Consider term *II*. First, notice that

$$\begin{aligned}
& \frac{1}{M} \sum_{j=1}^M \mathbb{E}_{g_t(\cdot|s_{t-1}^j)} [p(y_t|\tilde{s}_t^j) \omega_t^j(\tilde{s}_t^j, s_{t-1}^j)] W_{t-1}^j \\
& \xrightarrow{a.s.} \mathbb{E} \left[\mathbb{E}_{g_t(\cdot|s_{t-1})} [p(y_t|\tilde{s}_t^j) \omega_t(\tilde{s}_t^j, s_{t-1})] \right] \\
& = \int \left[\int p(y_t|\tilde{s}_t^j) \frac{p(\tilde{s}_t^j|s_{t-1})}{g_t(\tilde{s}_t^j|s_{t-1})} g_t(\tilde{s}_t^j|s_{t-1}) d\tilde{s}_t^j \right] p(s_{t-1}|Y_{1:t-1}) ds_{t-1} \\
& = p(y_t|Y_{1:t-1}),
\end{aligned} \tag{8.26}$$

which implies that term *II* converges to zero almost surely and ensures the consistency of the Monte Carlo approximation. Second, because

$$\mathbb{E}_{g_t(\cdot|s_{t-1})} [p(y_t|\tilde{s}_t^j) \omega_t(\tilde{s}_t^j, s_{t-1})] = \int p(y_t|\tilde{s}_t^j) \frac{p(\tilde{s}_t^j|s_{t-1})}{g_t(\tilde{s}_t^j|s_{t-1})} g_t(\tilde{s}_t^j|s_{t-1}) d\tilde{s}_t^j = p(y_t|s_{t-1}), \tag{8.27}$$

the variance of term *II* is independent of the choice of the importance density $g_t(\tilde{s}_t^j|s_{t-1}^j)$.

Now consider term *I*. Conditional on $\{s_{t-1}^j, W_{t-1}^j\}_{j=1}^M$ the weights W_{t-1}^j are known and the summands form a triangular array of mean-zero random variables that are independently distributed within each row. Recall that

$$p(y_t|\tilde{s}_t^j) \omega_t^j(\tilde{s}_t^j, s_{t-1}^j) = \frac{p(y_t|\tilde{s}_t^j) p(\tilde{s}_t^j|s_{t-1}^j)}{g_t(\tilde{s}_t^j|s_{t-1}^j)}. \tag{8.28}$$

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 term *I* conditional on $\{s_{t-1}^j, W_{t-1}^j\}_{j=1}^M$. Such filters are called adapted particle filters. In turn, this leads to a variance reduction of the Monte Carlo approximation of $p(y_t|Y_{1:t-1})$. A similar argument can be applied to the variance of $\tilde{h}_{t,M}$. The bootstrap particle filter simply sets $g_t(\tilde{s}_t^j|s_{t-1}^j) = p(\tilde{s}_t^j|s_{t-1}^j)$ and ignores the information in y_t . We will subsequently discuss more efficient choices of $g_t(\tilde{s}_t^j|s_{t-1}^j)$.

8.3 Adapting the Generic Filter

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), and Creal (2012). We will focus in this section on the choice of

the proposal density $g_t(\tilde{s}_t^j | s_{t-1}^j)$. The starting point is the conditionally-optimal importance distribution. In nonlinear DSGE models it is typically infeasible to generate draws from this distribution but it might be possible to construct an approximately conditionally-optimal proposal. Finally, we consider conditionally linear Gaussian state-space model for which it is possible to use Kalman filter updating for a subset of state variables conditional on the remaining elements of the state vector.

8.3.1 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_{t-1}^j . Given (y_t, s_{t-1}^j) the weights \tilde{w}_t^j are constant (as a function of \tilde{s}_t) if

$$g_t(\tilde{s}_t | s_{t-1}^j) = p(\tilde{s}_t | y_t, s_{t-1}^j), \quad (8.29)$$

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

$$\tilde{w}_t^j = \int p(y_t | s_t) p(s_t | s_{t-1}^j) ds_t. \quad (8.30)$$

In a typical (nonlinear) DSGE model applications 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) to generate draws from the conditionally-optimal distribution. However, for this approach to work efficiently, the user needs to find a good proposal distribution within the accept-reject algorithm.

As mentioned above, our numerical illustrations below are all based on DSGE models that take the form of the linear Gaussian state-space model (8.2). In this case, one can obtain $p(\tilde{s}_t | y_t, s_{t-1}^j)$ from the Kalman filter updating step described in Table 2.1. Let

$$\begin{aligned} \bar{s}_{t|t-1}^j &= \Phi_1 s_{t-1}^j & P_{t|t-1} &= \Phi_\epsilon \Sigma_\epsilon \Phi_\epsilon' \\ \bar{y}_{t|t-1}^j &= \Psi_0 + \Psi_1 t + \Psi_2 \bar{s}_{t|t-1}^j & F_{t|t-1} &= \Psi_2 P_{t|t-1} \Psi_2' + \Sigma_u \\ \bar{s}_{t|t}^j &= \bar{s}_{t|t-1}^j + P_{t|t-1} \Psi_2' F_{t|t-1}^{-1} (y_t - \bar{y}_{t|t-1}) & P_{t|t} &= P_{t|t-1} - P_{t|t-1} \Psi_2' F_{t|t-1}^{-1} \Psi_2 P_{t|t-1}. \end{aligned}$$

The conditionally optimal proposal distribution is given by

$$\tilde{s}_t | (s_{t-1}^j, y_t) \sim N(\bar{s}_{t|t}^j, P_{t|t}). \quad (8.31)$$

We will use (8.31) as a benchmark to document how accurate a well-adapted particle filter could be. In applications with nonlinear DSGE models, in which it is not possible to sample directly from $p(\tilde{s}_t | y_t, s_{t-1}^j)$, the documented level of accuracy is typically not attainable.

8.3.2 Approximately Conditionally-Optimal Distributions

In a typical DSGE model application, sampling from the conditionally-optimal importance distribution is infeasible or computationally too costly. Alternatively, 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 (8.31) 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).

8.3.3 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 (1.24) with a time-varying process $\pi_t^*(m_t)$ of the form

$$\pi_t^* = m_t \pi_L^* + (1 - m_t) \pi_H^*, \quad \mathbb{P}\{m_t = l | m_{t-1} = l\} = \eta_l, \quad l \in \{0, 1\}. \quad (8.32)$$

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 m_t . The solution

¹A richer DSGE model with a Markov switching target inflation rate was subsequently estimated by Liu, Waggoner, and Zha (2011).

Chapter 9

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 8 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 5 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 PFMH algorithms in other areas of econometrics are discussed in Flury and Shephard (2011).

9.1 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 $\{\hat{p}(Y|\theta), \hat{p}(\theta|Y), \hat{p}(Y)\}$. The first triplet consists of the exact likelihood function $p(Y|\theta)$ and the resulting posterior distribution and marginal data density defined as

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)}, \quad p(Y) = \int p(Y|\theta)p(\theta)d\theta. \quad (9.1)$$

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:

$$\hat{p}(\theta|Y) = \frac{\hat{p}(Y|\theta)p(\theta)}{\hat{p}(Y)}, \quad \hat{p}(Y) = \int \hat{p}(Y|\theta)p(\theta)d\theta. \quad (9.2)$$

By replacing the exact likelihood function $p(\theta|Y)$ with the particle filter approximation $\hat{p}(Y|\theta)$ in Algorithm 5 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 8 could be used in the PFMH algorithm. For concreteness, we use the generic filter described in Algorithm 12. At each iteration the filter generates draws \tilde{s}_t^j from the proposal distribution $g_t(\cdot|s_{t-1}^j)$. Let $\tilde{S}_t = (\tilde{s}_t^1, \dots, \tilde{s}_t^M)'$ 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 $A_t = (A_t^1, \dots, A_t^N)$ 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 on 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 (8.1).

For concreteness, consider the conditionally-optimal particle filter for a linear state-space model described in Chapter 8.3.1. Implementation of this filter requires sampling from a $N(\bar{s}_{t|t}^j, P_{t|t}^j)$ 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 (8.2). 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 $N(\bar{s}_{t|t}^j, P_{t|t}^j)$. 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

$$\hat{p}(Y_{1:T}|\theta) = g(Y_{1:T}|\theta, U_{1:T}). \quad (9.3)$$

where

$$U_{1:T} \sim p(U_{1:T}) = \prod_{t=1}^T p(U_t). \quad (9.4)$$

The PFMH algorithm can be interpreted as operating on an enlarged probability space for the triplet $(Y_{1:T}, \theta, U_{1:T})$. Define the joint distribution

$$p_g(Y_{1:T}, \theta, U_{1:T}) = g(Y_{1:T}|\theta, U_{1:T})p(U_{1:T})p(\theta). \quad (9.5)$$

The PFMH algorithm samples from the joint posterior

$$p_g(\theta, U_{1:T}|Y_{1:T}) \propto g(Y| \theta, U_{1:T})p(U_{1:T})p(\theta) \quad (9.6)$$

and discards the draws of $(U_{1:T})$. For this procedure to be valid, it has to be the case that marginalizing the joint posterior $p_g(\theta, U_{1:T}|Y_{1:T})$ with respect to $(U_{1:T})$ 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 :

$$\mathbb{E}[\hat{p}(Y_{1:T}|\theta)] = \int g(Y_{1:T}|\theta, U_{1:T})p(U_{1:T})d\theta = p(Y_{1:T}|\theta). \quad (9.7)$$

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 $\hat{p}(Y_{1:T}|\theta)$. On the enlarged probability space, one needs to generate a proposed draw for both θ and $U_{1:T}$. We denote these draws by ϑ and $U_{1:T}^*$. The proposal distribution for $(\vartheta, U_{1:T}^*)$ in the MH algorithm is given by $q(\vartheta|\theta^{(i-1)})p(U_{1:T}^*)$. There is no need to keep track of the draws $(U_{1:T}^*)$, because the acceptance ratio for Algorithm 18 can be written as follows (omitting the time subscripts):

$$\begin{aligned} \alpha(\vartheta|\theta^{i-1}) &= \min \left\{ 1, \frac{\frac{g(Y|\vartheta, U^*)p(U^*)p(\vartheta)}{q(\vartheta|\theta^{(i-1)})p(U^*)}}{\frac{g(Y|\theta^{(i-1)}, U^{(i-1)})p(U^{(i-1)})p(\theta^{(i-1)})}{q(\theta^{(i-1)}|\theta^*)p(U^{(i-1)})}} \right\} \\ &= \min \left\{ 1, \frac{\hat{p}(Y|\vartheta)p(\vartheta)/q(\vartheta|\theta^{(i-1)})}{\hat{p}(Y|\theta^{(i-1)})p(\theta^{(i-1)})/q(\theta^{(i-1)}|\vartheta)} \right\}. \end{aligned} \quad (9.8)$$

The terms $p(U^*)$ and $p(U^{(i-1)})$ cancel from the expression in the first line of (9.8) and it suffices to record the particle filter likelihood approximations $\hat{p}(Y|\vartheta)$ and $\hat{p}(Y|\theta^{(i-1)})$.

9.2 Application to the Small-Scale DSGE 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 $N_{run} = 20$ runs of each algorithm. In each run we generate $N = 100,000$ posterior draws and discard the first $N_0 = 50,000$. As in Chapter 8.6, we use $M = 40,000$ particles for the bootstrap filter and $M = 400$ particles for the conditionally-optimal filter. A single run of the RWMH-V algorithm takes 1 minute and 30 seconds 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 9.1. Most notably, despite the inaccurate likelihood approximation of the bootstrap PF documented in Chapter 8.6, the PFMH works remarkably well. Columns 2 to 4 of the table report posterior means which are computed by pooling the

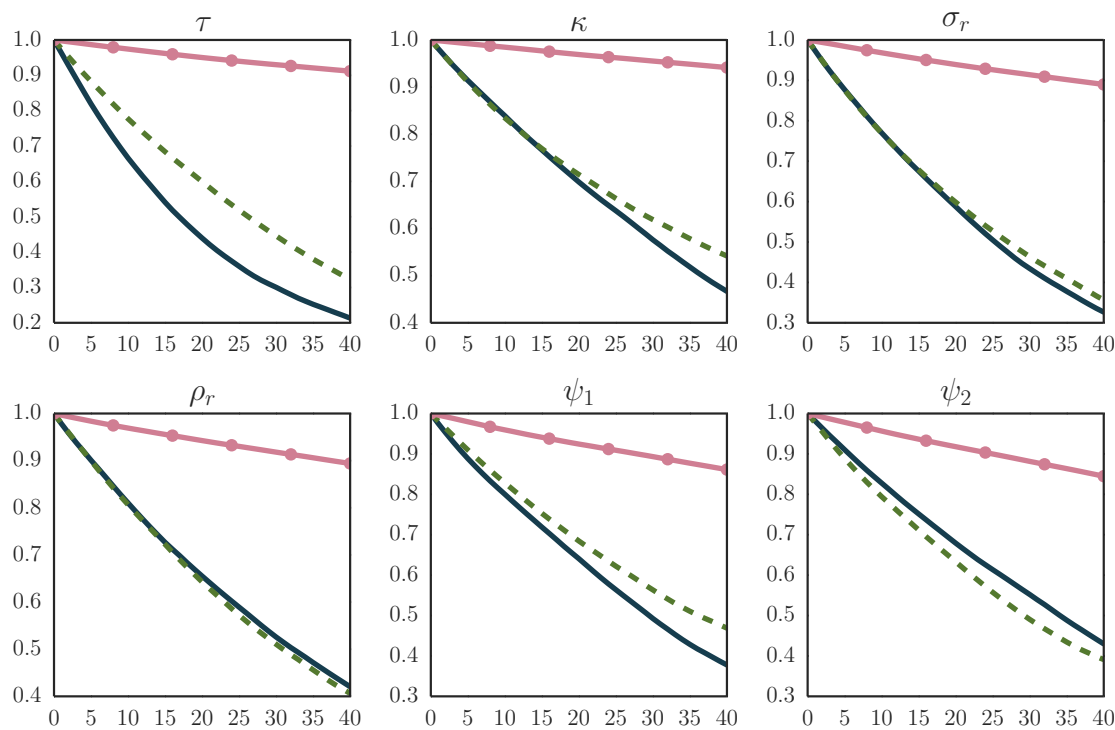
Table 9.1: Accuracy of MH Approximations

	Posterior Mean (Pooled)			Inefficiency Factors			Std Dev of Means		
	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF	KF	CO-PF	BS-PF
τ	2.63	2.62	2.64	66.17	126.76	1360.22	0.020	0.028	0.091
κ	0.82	0.81	0.82	128.00	97.11	1887.37	0.007	0.006	0.026
ψ_1	1.88	1.88	1.87	113.46	159.53	749.22	0.011	0.013	0.029
ψ_2	0.64	0.64	0.63	61.28	56.10	681.85	0.011	0.010	0.036
ρ_r	0.75	0.75	0.75	108.46	134.01	1535.34	0.002	0.002	0.007
ρ_g	0.98	0.98	0.98	94.10	88.48	1613.77	0.001	0.001	0.002
ρ_z	0.88	0.88	0.88	124.24	118.74	1518.66	0.001	0.001	0.005
$r^{(A)}$	0.44	0.44	0.44	148.46	151.81	1115.74	0.016	0.016	0.044
$\pi^{(A)}$	3.32	3.33	3.32	152.08	141.62	1057.90	0.017	0.016	0.045
$\gamma^{(Q)}$	0.59	0.59	0.59	106.68	142.37	899.34	0.006	0.007	0.018
σ_r	0.24	0.24	0.24	35.21	179.15	1105.99	0.001	0.002	0.004
σ_g	0.68	0.68	0.67	98.22	64.18	1490.81	0.003	0.002	0.011
σ_z	0.32	0.32	0.32	84.77	61.55	575.90	0.001	0.001	0.003
$\ln \hat{p}(Y)$	-357.14	-357.17	-358.32				0.040	0.038	0.949

Notes: Results are based on $N_{run} = 20$ runs of the PF-RWMH-V algorithm. Each run of the algorithm generates $N = 100,000$ draws and the first $N_0 = 50,000$ are discarded. The likelihood function is computed with the Kalman filter (KF), bootstrap particle filter (BS-PF, $M = 40,000$) or conditionally-optimal particle filter (CO-PF, $M = 400$). “Pooled” means that we are pooling the draws from the $N_{run} = 20$ runs to compute posterior statistics.

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 methods. Columns 5 to 7 contain the inefficiency factors $\text{InEff}_N(\bar{\theta})$ for each parameters and the last three columns of Table 9.1 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. Under the Kalman filter the inefficiency factors range from 35 to about 150, whereas under the bootstrap particle filter they range from 575 to 1,890. As stressed in Section 9.1 the most important requirement for PFMH algorithms is that the particle filter approximation

Figure 9.1: Autocorrelation of PFMH Draws



Notes: The figure depicts autocorrelation functions computed from the output of the 1 Block RWMH-V algorithm based on the Kalman filter (solid), the conditionally-optimal particle filter (dashed) and the bootstrap particle filter (solid with dots).

is unbiased – it does not have to be exact.

In Figure 9.1 we depict autocorrelation functions for parameter draws computed based on the output of the PFMH algorithms. As a benchmark, the figure also contains autocorrelation functions obtained from the sampler that uses the exact likelihood function computed with the Kalman filter. While under the conditionally-optimal particle filter the persistence of the Markov chain for the DSGE model parameters is comparable to the persistence under the Kalman filter, the use of the bootstrap particle filter raises the serial correlation of the parameter draws drastically, which leads to the less precise Monte Carlo approximations reported in Table 9.1.

9.3 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 summarized 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, running each algorithm $N_{run} = 20$ times. The bootstrap particle filters with $M = 40,000$ and $M = 400,000$ particles turned out to be too inaccurate to deliver reliable posterior estimates.

Table 9.2 show the results for the Kalman filter and the conditionally-optimal PF. While the the MCMC chains generated using the KF and CO-PF generate roughly the same means for the parameter draws on average, the variability across the chains is much higher for the CO-PF. According to inefficiency factors, the KF chains are about 10 more efficient than the CO-PF.

The results are summarized in Table 9.2. Due to the computational complexity of the PF-RWMH-V algorithm, the results reported in the table are based on $N = 10,000$ instead of $N = 100,000$ draws from the posterior distribution. We used the conditionally-optimal PF with $M = 40,000$ 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 conditionally-optimal 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 varies 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 9.1 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 data 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 PF 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 8.4 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 , $i = 1, \dots, N$) that is generated by using the exact likelihood function, the sampler easily gets stuck in the sense that the acceptance rate for proposed draws drops to zero. Once the PF has generated an estimate $\hat{p}(Y|\theta)$ that exceeds $p(Y|\theta)$ by a wide margin, it becomes extremely difficult to move to a nearby θ . The reason is that, on the one hand, most of the PF evaluations underestimate $p(Y|\theta)$ and, on the other hand, two θ 's that are close to each other tend to be associated with similar likelihood values.

9.4 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 30:20:33 [hh:mm:ss] 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 8.6 and 8.7 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 one 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 9.2: Accuracy of MH Approximations

	Post. Mean (Pooled)		Ineff. Factors		Std Dev of Means	
	KF	CO-PF	KF	CO-PF	KF	CO-PF
$(100\beta^{-1} - 1)$	0.14	0.14	172.58	3732.90	0.007	0.034
$\bar{\pi}$	0.73	0.74	185.99	4343.83	0.016	0.079
\bar{l}	0.51	0.37	174.39	3133.89	0.130	0.552
α	0.19	0.20	149.77	5244.47	0.003	0.015
σ_c	1.49	1.45	86.27	3557.81	0.013	0.086
Φ	1.47	1.45	134.34	4930.55	0.009	0.056
φ	5.34	5.35	138.54	3210.16	0.131	0.628
h	0.70	0.72	277.64	3058.26	0.008	0.027
ξ_w	0.75	0.75	343.89	2594.43	0.012	0.034
σ_l	2.28	2.31	162.09	4426.89	0.091	0.477
ξ_p	0.72	0.72	182.47	6777.88	0.008	0.051
ι_w	0.54	0.53	241.80	4984.35	0.016	0.073
ι_p	0.48	0.50	205.27	5487.34	0.015	0.078
ψ	0.45	0.44	248.15	3598.14	0.020	0.078
r_π	2.09	2.09	98.32	3302.07	0.020	0.116
ρ	0.80	0.80	241.63	4896.54	0.006	0.025
r_y	0.13	0.13	243.85	4755.65	0.005	0.023
$r_{\Delta y}$	0.21	0.21	101.94	5324.19	0.003	0.022
ρ_a	0.96	0.96	153.46	1358.87	0.002	0.005
ρ_b	0.22	0.21	325.98	4468.10	0.018	0.068
ρ_g	0.97	0.97	57.08	2687.56	0.002	0.011
ρ_i	0.71	0.70	219.11	4735.33	0.009	0.044
ρ_r	0.54	0.54	194.73	4184.04	0.020	0.094
ρ_p	0.80	0.81	338.69	2527.79	0.022	0.061
ρ_w	0.94	0.94	135.83	4851.01	0.003	0.019
ρ_{ga}	0.41	0.37	196.38	5621.86	0.025	0.133
μ_p	0.66	0.66	300.29	3552.33	0.025	0.087
μ_w	0.82	0.81	218.43	5074.31	0.011	0.052
σ_a	0.34	0.34	128.00	5096.75	0.005	0.034
σ_b	0.24	0.24	186.13	3494.71	0.004	0.016
σ_g	0.51	0.49	208.14	2945.02	0.006	0.021
σ_i	0.43	0.44	115.42	6093.72	0.006	0.043
σ_r	0.14	0.14	193.37	3408.01	0.004	0.016
σ_p	0.13	0.13	194.22	4587.76	0.003	0.013
σ_w	0.22	0.22	211.80	2256.19	0.004	0.012
$\ln \hat{p}(Y)$	-964.44	-1017.94			0.298	9.139

Notes: Results are based on $N_{run} = 20$ runs of the PF-RWMH-V algorithm. Each run of the algorithm generates $N = 10,000$ 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 $N_{run} = 20$ runs to compute posterior statistics. The CO-PF uses $M = 40,000$ particles to compute the likelihood.

Chapter 10

Combining Particle Filters with SMC Samplers

Following recent work by Chopin, Jacob, and Papaspiliopoulos (2012), we now combine the SMC algorithm of Chapter 5 with the particle filter approximation of the likelihood function developed in Chapter 8 to develop an SMC^2 algorithm.

10.1 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. The 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 (5.3), 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 10.1: 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 (9.3), we write the particle approximations as

$$\hat{p}(y_{t_{n-1}+1:t_n} | Y_{1:t_{n-1}}, \theta) = g(y_{t_{n-1}+1:t_n} | Y_{1:t_{n-1}}, \theta, U_{1:t_n}), \quad \hat{p}(Y_{1:t_n} | \theta_n) = g(Y_{1:t_n} | \theta_n, U_{1:t_n}). \quad (10.1)$$

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 (9.4). 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

$$p(U_{1:t_n}) = p(U_{1:t_1})p(U_{t_1+1:t_2}) \cdots p(U_{t_{n-1}+1:t_n}). \quad (10.2)$$

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 $i = 1, \dots, N$ and the particles generated by the particle filter are indexed $j = 1, \dots, M$. 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 10.1.

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: $\theta_0^i \stackrel{iid}{\sim} p(\theta)$ and $W_0^i = 1$, $i = 1, \dots, N$.

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

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

$$\tilde{w}_t^i = \hat{p}(y_t | Y_{1:t-1}, \theta_{t-1}^i) = g(y_t | Y_{1:t-1}, \theta_{t-1}^i, U_{1:t}^i) \quad (10.3)$$

and the normalized weights

$$\tilde{W}_t^i = \frac{\tilde{w}_t^i W_{t-1}^i}{\frac{1}{N} \sum_{i=1}^N \tilde{w}_t^i W_{t-1}^i}, \quad i = 1, \dots, N. \quad (10.4)$$

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

$$\tilde{h}_{t,N} = \frac{1}{N} \sum_{i=1}^N \tilde{W}_t^i h(\theta_{t-1}^i). \quad (10.5)$$

(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

$$\hat{h}_{t,N} = \frac{1}{N} \sum_{j=1}^N W_t^j h(\hat{\theta}_t^j). \quad (10.6)$$

(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

$$q(\vartheta_t^i | \hat{\theta}_t^i) p(U_{1:t}^{*i}) \quad (10.7)$$

and the acceptance ratio can be expressed as

$$\alpha(\vartheta_t^i|\hat{\theta}_t^i) = \min \left\{ 1, \frac{\hat{p}(Y_{1:t}|\vartheta_t^i)p(\vartheta_t^i)/q(\vartheta_t^i|\hat{\theta}_t^i)}{\hat{p}(Y_{1:t}|\hat{\theta}_t^i)p(\hat{\theta}_t^i)/q(\hat{\theta}_t^i|\vartheta_t^i)} \right\}. \quad (10.8)$$

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

$$\bar{h}_{t,N} = \frac{1}{N} \sum_{i=1}^N h(\theta_t^i) W_t^i. \quad (10.9)$$

3. For $t = T$ the final importance sampling approximation of $\mathbb{E}_{\pi}[h(\theta)]$ is given by:

$$\bar{h}_{T,N} = \sum_{i=1}^N h(\theta_T^i) W_T^i. \quad (10.10)$$

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

$$\int \int h(\theta, U_{1:t-1}) p(U_{1:t-1}) p(\theta|Y_{1:t-1}) dU_{1:t-1} d\theta \approx \frac{1}{N} \sum_{i=1}^N h(\theta_{t-1}^i, U_{1:t-1}^i) W_{t-1}^i. \quad (10.11)$$

This implies that we obtain the familiar approximation for functions $h(\cdot)$ that do not depend on $U_{1:t-1}$

$$\int h(\theta) p(\theta|Y_{1:t-1}) d\theta \approx \frac{1}{N} \sum_{i=1}^N h(\theta_{t-1}^i) W_{t-1}^i. \quad (10.12)$$

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 (10.1), the particle filter approximation of the likelihood increment can be written as

$$\hat{p}(y_t|Y_{1:t-1}, \theta_{t-1}^i) = g(y_t|Y_{1:t-1}, U_{1:t}^i, \theta_{t-1}^i). \quad (10.13)$$

The value of the likelihood function for $Y_{1:t}$ can be tracked recursively as follows:

$$\begin{aligned}\hat{p}(Y_{1:t}|\theta_{t-1}^i) &= \hat{p}(y_t|Y_{1:t-1}, \theta_{t-1}^i)\hat{p}(Y_{1:t-1}|\theta_{t-1}^i) \\ &= g(y_t|Y_{1:t}, U_{1:t}^i, \theta_{t-1}^i)g(Y_{1:t-1}|U_{1:t-1}^i, \theta_{t-1}^i) \\ &= g(Y_{1:t}|U_{1:t}^i, \theta_{t-1}^i).\end{aligned}\tag{10.14}$$

The last equality follows because conditioning $g(Y_{1:t-1}|U_{1:t-1}^i, \theta_{t-1}^i)$ 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 (10.11) that the Monte Carlo average $\frac{1}{N} \sum_{i=1}^N h(\theta_{t-1}^i)\tilde{w}_t^i W_{t-1}^i$ approximates the following integral

$$\begin{aligned}&\int \int h(\theta)g(y_t|Y_{1:t-1}, U_{1:t}, \theta)p(U_{1:t})p(\theta|Y_{1:t-1})dU_{1:t}d\theta \\ &= \int h(\theta) \left[\int g(y_t|Y_{1:t-1}, U_{1:t}, \theta)p(U_{1:t})dU_{1:t} \right] p(\theta|Y_{1:t-1})d\theta.\end{aligned}\tag{10.15}$$

Provided that the particle filter approximation of the likelihood increment is unbiased, that is,

$$\int g(y_t|Y_{1:t-1}, U_{1:t}, \theta)p(U_{1:t})dU_{1:t} = p(y_t|Y_{1:t-1}, \theta)\tag{10.16}$$

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 $\hat{\theta}_t^i$ but we also want to track the corresponding value of the likelihood function $\hat{p}(Y_{1:t}|\hat{\theta}_t^i)$ as well as the particle approximation of the state, given by $\{s_t^{i,j}, W_t^{i,j}\}$, and the set of random numbers $U_{1:t}^i$. 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 $U_{1:t}^i$'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 9.1. For each particle i there is a proposed value ϑ_t^i and an associated particle filter approximation $\hat{p}(Y_{1:t}|\vartheta_t^i)$ of the likelihood and sequence of random vectors $U_{1:t}^*$ drawn from the distribution $p(U_{1:t})$ in (10.2). As in (9.8), the densities $p(U_{1:t}^i)$

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 .

10.2 Application to the Small-Scale DSGE 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 9.2. 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; and 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 10.2 summarizes the results from running each algorithm $N_{run} = 20$ times. We report pooled posterior means from the output of the 20 runs as well as inefficiency factors $\text{InEff}_N(\bar{\theta})$ and 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 and the inefficiency factors 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

Table 10.2: Accuracy of SMC^2 Approximations

	Posterior Mean (Pooled)				Inefficiency Factors				Std Dev of Means			
	KF(L)	KF(D)	CO-PF	BS-PF	KF(L)	KF(D)	CO-PF	BS-PF	KF(L)	KF(D)	CO-PF	BS-PF
τ	2.65	2.67	2.68	2.53	1.51	10.41	47.60	6570	0.01	0.03	0.07	0.76
κ	0.81	0.81	0.81	0.70	1.40	8.36	40.60	7223	0.00	0.01	0.01	0.18
ψ_1	1.87	1.88	1.87	1.89	3.29	18.27	22.56	4785	0.01	0.02	0.02	0.27
ψ_2	0.66	0.66	0.67	0.65	2.72	10.02	43.30	4197	0.01	0.02	0.03	0.34
ρ_r	0.75	0.75	0.75	0.72	1.31	11.39	60.18	14979	0.00	0.00	0.01	0.08
ρ_g	0.98	0.98	0.98	0.95	1.32	4.28	250.34	21736	0.00	0.00	0.00	0.04
ρ_z	0.88	0.88	0.88	0.84	3.16	15.06	35.35	10802	0.00	0.00	0.00	0.05
$r^{(A)}$	0.45	0.46	0.44	0.46	1.09	26.58	73.78	7971	0.00	0.02	0.04	0.42
$\pi^{(A)}$	3.32	3.31	3.31	3.56	2.15	40.45	158.64	6529	0.01	0.03	0.06	0.40
$\gamma^{(Q)}$	0.59	0.59	0.59	0.64	2.35	32.35	133.25	5296	0.00	0.01	0.03	0.16
σ_r	0.24	0.24	0.24	0.26	0.75	7.29	43.96	16084	0.00	0.00	0.00	0.06
σ_g	0.68	0.68	0.68	0.73	1.30	1.48	20.20	5098	0.00	0.00	0.00	0.08
σ_z	0.32	0.32	0.32	0.42	2.32	3.63	26.98	41284	0.00	0.00	0.00	0.11
$\ln p(Y)$	-358.75	-357.34	-356.33	-340.47					0.120	1.191	4.374	14.49

Notes: Results are based on $N_{run} = 20$ runs of the SMC^2 algorithm with $N = 4,000$ particles. D is data tempering and L is likelihood tempering. KF is Kalman filter, CO-PF is conditionally-optimal PF with $M = 400$, BS-PF is bootstrap PF with $M = 40,000$. CO-PF and BS-PF use data tempering.

convex tempering schedule that adds very little likelihood information in the initial stages. The implied tempering schedule of our sequential estimation procedure is linear and adds a full observation in stage $n = 1$ (recall that $n = 0$ 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. The inefficiency factor for the KF(D) algorithm range from 1.5 to 40, whereas they range from 20 to 250 for CO-PF. A comparison with Table 9.1 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 and the smallest inefficiency factor is 4,200. Nonetheless, the pooled posterior means are fairly close to those obtained from the other three algorithms.

10.3 Computational Considerations

The SMC^2 results reported in Table 10.2 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 [mm:ss] minutes, where as the algorithm with bootstrap PF ($N = 4,000$ and $M = 40,000$) runs for 08:05:35 [hh:mm:ss] hours. The bootstrap PF performs poorly in terms of accuracy and runtime.

After running the particle filter for the sample $Y_{1:t-1}$ one 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 initial 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 (2007): “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, B., P. CUBA-BORDA, AND F. SCHORFHEIDE (2014): “Macroeconomic Dynamics Near the ZLB: A Tale of Two Countries,” *NBER Working Paper*, 19248.
- ARUOBA, S. BORAGAN, L. B., AND F. SCHORFHEIDE (2013): “Assessing DSGE Model Nonlinearities,” *NBER Working Paper*, 19693, Manuscript, University of Maryland and University of Pennsylvania.

- 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.
- CANOVA, F., AND L. SALA (2009): “Back to Square One: Identification Issues in DSGE Models,” *Journal of Monetary Economics*, 56, 431 – 449.
- 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.
- CHANG, Y., T. DOH, AND F. SCHORFHEIDE (2007): “Non-stationary Hours in a DSGE Model,” *Journal of Money, Credit, and Banking*, 39(6), 1357–1373.
- CHEN, R., AND J. LIU (2000): “Mixture Kalman Filters,” *Journal of the Royal Statistical Society Series B*, 62, 493–508.
- CHIB, S., AND E. GREENBERG (1995): “Understanding the Metropolis-Hastings Algorithm,” *The American Statistician*, 49, 327–335.
- 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): “SMC²: 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.
- COX, W., AND E. HIRSCHHORN (1983): “The Market Value of US Government Debt,” *Journal of Monetary Economics*, 11, 261–272.
- CREAL, D. (2007): “Sequential Monte Carlo Samplers for Bayesian DSGE Models,” *Unpublished Manuscript, Vrije Universiteit*.
- (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.

- DEJONG, D. N., R. LIESENFELD, G. V. MOURA, J.-F. RICHARD, AND H. DHARMARAJAN (2013): “Efficient Likelihood Evaluation of State-Space Representations,” *Review of Economic Studies*, 80(2), 538–567.
- DEL MORAL, P. (2004): *Feynman-Kac Formulae*. Springer Verlag.
- (2013): *Mean Field Simulation for Monte Carlo Integration*. Chapman & Hall/CRC.
- DEL NEGRO, M., R. B. HASEGAWA, AND F. SCHORFHEIDE (2014): “Dynamic Prediction Pools: An Investigation of Financial Frictions and Forecasting Performance,” *NBER Working Paper*, 20575.
- 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 *Handbook 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 (2014): “Adaptive Sequential Posterior Simulators for Massively Parallel Computing Environments,” in *Advances in Econometrics*, ed. by I. Jeliazkov, and D. Poirier, vol. 34, chap. 6, pp. 1–44. Emerald Group Publishing Limited.
- FARMER, R., D. WAGGONER, AND T. ZHA (2009): “Understanding Markov Switching Rational Expectations Models,” *Journal of Economic Theory*, 144(5), 1849–1867.
- FERNÁNDEZ-VILLAYERDE, J., G. GORDON, P. GUERRÓN-QUINTANA, AND J. F. RUBIO-RAMÍREZ (2012): “Nonlinear Adventures at the Zero Lower Bound,” *National Bureau of Economic Research Working Paper*.

- FERNANDEZ-VILLAVERDE, J., AND J. F. RUBIO-RAMIREZ (2004): “Comparing dynamic equilibrium models to data: a Bayesian approach,” *Journal of Econometrics*, 123(1), 153–187.
- FERNÁNDEZ-VILLAVERDE, J., AND J. F. RUBIO-RAMÍREZ (2007): “Estimating Macroeconomic Models: A Likelihood Approach,” *Review of Economic Studies*, 74(4), 1059–1087.
- (2013): “Macroeconomics and Volatility: Data, Models, and Estimation,” in *Advances in Economics and Econometrics: Tenth World Congress*, ed. by D. Acemoglu, M. Arellano, and E. Dekel, vol. 3, pp. 137–183. Cambridge University Press.
- 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.
- FOERSTER, A., J. F. RUBIO-RAMIREZ, D. F. WAGGONER, AND T. ZHA (2014): “Perturbation Methods for Markov-Switching DSGE Models,” *Manuscript, FRB Kansas City and Atlanta*.
- 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.
- GIROLAMI, M., AND B. CALDERHEAD (2011): “Riemann Manifold Langevin and Hamilton Monte Carlo Methods (with discussion),” *Journal of the Royal Statistical Society Series B*, 73, 123–214.
- 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.
- GUST, C., D. LOPEZ-SALIDO, AND M. E. SMITH (2012): “The Empirical Implications of the Interest-Rate Lower Bound,” *Manuscript, Federal Reserve Board*.
- HALPERN, E. F. (1974): “Posterior Consistency for Coefficient Estimation and Model Selection in the General Linear Hypothesis,” *Annals of Statistics*, 2(4), 703–712.
- HAMILTON, J. D. (1989): “A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle,” *Econometrica*, 57(2), 357–384.
- (1994): *Time Series Analysis*. Pri.
- HAMMERSLEY, J., AND D. HANDSCOMB (1964): *Monte Carlo Methods*. Methuen and Company, London.
- HASTINGS, W. (1970): “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” *Biometrika*, 57, 97–109.
- 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.
- HOETING, J. A., D. MADIGAN, A. E. RAFTERY, AND C. T. VOLINSKY (1999): “Bayesian Model Averaging: A Tutorial,” *Statistical Science*, 14(4), 382–417.
- HOLMES, M. (1995): *Introduction to Perturbation Methods*. Cambridge University Press.
- IRELAND, P. N. (2004): “A Method for Taking Models to the Data,” *Journal of Economic Dynamics and Control*, 28(6), 1205–1226.
- ISKREV, N. (2010): “Local Identification of DSGE Models,” *Journal of Monetary Economics*, 2, 189–202.
- 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.
- KANTAS, N., A. DOUCET, S. SINGH, J. MACIEJOWSKI, AND N. CHOPIN (2014): “On Particle Methods for Parameter Estimation in State-Space Models,” *arXiv Working Paper*, 1412.8659v1.
- KASS, R. E., AND A. E. RAFTERY (1995): “Bayes Factors,” *Journal of the American Statistical Association*, 90(430), 773–795.
- KIM, J., S. KIM, E. SCHAUMBURG, AND C. A. SIMS (2008): “Calculating and Using Second-Order Accurate Solutions of Discrete Time Dynamic Equilibrium Models,” *Journal of Economic Dynamics and Control*, 32, 3397–3414.
- KIM, J., AND F. RUGE-MURCIA (2009): “How Much Inflation is Necessary to Grease the Wheels,” *Journal of Monetary Economics*, 56, 365–377.
- 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 Singular 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.
- KLOEK, T., AND H. K. VAN DIJK (1978): “Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo,” *Econometrica*, 46, 1–20.
- 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.

- KOMUNJER, I., AND S. NG (2011): “Dynamic Identification of DSGE Models,” *Econometrica*, 79(6), 1995–2032.
- KOOP, G. (2003): *Bayesian Econometrics*. John Wiley & Sons, Inc.
- 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.
- LANCASTER, T. (2004): *An Introduction to Modern Bayesian Econometrics*. Blackwell Publishing.
- LEAMER, E. E. (1978): *Specification Searches*. Wiley, New York.
- LEEPER, E. M., M. PLANTE, AND N. TRAUM (2010): “Dynamics of Fiscal 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 Framework 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.
- (2006): “A Bayesian Look at the New Open Macroeconomics,” *NBER Macroeconomics Annual 2005*.
- MALIAR, L., AND S. MALIAR (2014): “Merging Simulation and Projection Approaches to Solve High-Dimensional Problems with an Application to a New Keynesian Model,” *Quantitative Economics*, forthcoming.

- METROPOLIS, N., A. ROSENBLUTH, M. ROSENBLUTH, A. TELLER, AND E. TELLER (1953): "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1091.
- MIN, C.-K., AND A. ZELLNER (1993): "Bayesian and Non-Bayesian Methods for Combining Models and Forecasts with Applications to Forecasting International Growth Rates," *Journal of Econometrics*, 56(1-2), 89–118.
- 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.
- NEAL, R. (2010): "MCMC using Hamiltonian Dynamics," in *Handbook of Markov Chain Monte Carlo*, ed. by S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, pp. 113–162. Chapman & Hall, CRC Press.
- 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.
- 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. SANTAELULIA-LLOPIS (2012): "Methods versus Substance: Measuring the Effects of Technology Shocks," *Journal of Monetary Economics*, 59(8), 826–846.
- ROBERT, C. P. (1994): *The Bayesian Choice*. Springer Verlag.
- ROBERT, C. P., AND G. CASELLA (2004): *Monte Carlo Statistical Methods*. Springer.
- 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., A. GELMAN, AND W. R. GILKS (1997): “Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms,” *The Annals of Applied Probability*, 7(1), 110–120.
- 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.
- 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.
- SCHMITT-GROHÉ, S., AND M. URIBE (2004): “Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function,” *Journal of Economic Dynamics and Control*, 28, 755–775.
- (2012): “What’s News in Business Cycles?,” *Econometrica*, forthcoming.
- 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): “Identifying Long-Run Risks: A Bayesian Mixed-Frequency Approach,” *NBER Working Paper*, 20303.
- SCHWARZ, G. (1978): “Estimating the Dimension of a Model,” *Annals of Statistics*, 6(2), 461–464.

- 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.
- TIERNEY, L. (1994): “Markov Chains for Exploring Posterior Distributions,” *The Annals of Statistics*, 22(4), 1701–1728.
- VAN DER VAART, A. (1998): *Asymptotic Statistics*. Cambridge University Press.
- WOODFORD, M. (2003): *Interest and Prices*. Princeton University Press.
- WRIGHT, J. (2008): “Bayesian Model Averaging and Exchange Rate Forecasting,” *Journal of Econometrics*, 146, 329–341.

Appendix A

Model Descriptions

A.1 Smets-Wouters Model

The log-linearized equilibrium conditions of the Smets and Wouters (2007) model take the following form:

$$\hat{y}_t = c_y \hat{c}_t + i_y \hat{i}_t + z_y \hat{z}_t + \varepsilon_t^g \quad (\text{A.1})$$

$$\begin{aligned} \hat{c}_t = & \frac{h/\gamma}{1+h/\gamma} \hat{c}_{t-1} + \frac{1}{1+h/\gamma} E_t \hat{c}_{t+1} + \frac{wl_c(\sigma_c - 1)}{\sigma_c(1+h/\gamma)} (\hat{l}_t - E_t \hat{l}_{t+1}) \\ & - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} (\hat{r}_t - E_t \hat{\pi}_{t+1}) - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} \varepsilon_t^b \end{aligned} \quad (\text{A.2})$$

$$\hat{i}_t = \frac{1}{1+\beta\gamma^{(1-\sigma_c)}} \hat{i}_{t-1} + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}} E_t \hat{i}_{t+1} + \frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})} \hat{q}_t + \varepsilon_t^i \quad (\text{A.3})$$

$$\hat{q}_t = \beta(1-\delta)\gamma^{-\sigma_c} E_t \hat{q}_{t+1} - \hat{r}_t + E_t \hat{\pi}_{t+1} + (1-\beta(1-\delta)\gamma^{-\sigma_c}) E_t \hat{r}_{t+1}^k - \varepsilon_t^b \quad (\text{A.4})$$

$$\hat{y}_t = \Phi(\alpha \hat{k}_t^s + (1-\alpha)\hat{l}_t + \varepsilon_t^a) \quad (\text{A.5})$$

$$\hat{k}_t^s = \hat{k}_{t-1} + \hat{z}_t \quad (\text{A.6})$$

$$\hat{z}_t = \frac{1-\psi}{\psi} \hat{r}_t^k \quad (\text{A.7})$$

$$\hat{k}_t = \frac{(1-\delta)}{\gamma} \hat{k}_{t-1} + (1-(1-\delta)/\gamma)\hat{i}_t + (1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i \quad (\text{A.8})$$

$$\hat{\mu}_t^p = \alpha(\hat{k}_t^s - \hat{l}_t) - \hat{w}_t + \varepsilon_t^a \quad (\text{A.9})$$

$$\begin{aligned} \hat{\pi}_t = & \frac{\beta\gamma^{(1-\sigma_c)}}{1+\iota_p\beta\gamma^{(1-\sigma_c)}} E_t \hat{\pi}_{t+1} + \frac{\iota_p}{1+\beta\gamma^{(1-\sigma_c)}} \hat{\pi}_{t-1} \\ & - \frac{(1-\beta\gamma^{(1-\sigma_c)})\xi_p(1-\xi_p)}{(1+\iota_p\beta\gamma^{(1-\sigma_c)})(1+(\Phi-1)\varepsilon_p)\xi_p} \hat{\mu}_t^p + \varepsilon_t^p \end{aligned} \quad (\text{A.10})$$

$$\hat{r}_t^k = \hat{l}_t + \hat{w}_t - \hat{k}_t^s \quad (\text{A.11})$$

$$\hat{\mu}_t^w = \hat{w}_t - \sigma_l \hat{l}_t - \frac{1}{1 - h/\gamma} (\hat{c}_t - h/\gamma \hat{c}_{t-1}) \quad (\text{A.12})$$

$$\hat{w}_t = \frac{\beta\gamma^{(1-\sigma_c)}}{1 + \beta\gamma^{(1-\sigma_c)}} (E_t \hat{w}_{t+1} + E_t \hat{\pi}_{t+1}) + \frac{1}{1 + \beta\gamma^{(1-\sigma_c)}} (\hat{w}_{t-1} - \iota_w \hat{\pi}_{t-1}) \quad (\text{A.13})$$

$$\begin{aligned} & - \frac{1 + \beta\gamma^{(1-\sigma_c)} \iota_w}{1 + \beta\gamma^{(1-\sigma_c)}} \hat{\pi}_t - \frac{(1 - \beta\gamma^{(1-\sigma_c)} \xi_w)(1 - \xi_w)}{(1 + \beta\gamma^{(1-\sigma_c)})(1 + (\lambda_w - 1)\epsilon_w) \xi_w} \hat{\mu}_t^w + \varepsilon_t^w \\ \hat{r}_t &= \rho \hat{r}_{t-1} + (1 - \rho)(r_\pi \hat{\pi}_t + r_y (\hat{y}_t - \hat{y}_t^*)) \\ & + r_{\Delta y} ((\hat{y}_t - \hat{y}_t^*) - (\hat{y}_{t-1} - \hat{y}_{t-1}^*)) + \varepsilon_t^r. \end{aligned} \quad (\text{A.14})$$

The exogenous shocks evolve according to

$$\varepsilon_t^a = \rho_a \varepsilon_{t-1}^a + \eta_t^a \quad (\text{A.15})$$

$$\varepsilon_t^b = \rho_b \varepsilon_{t-1}^b + \eta_t^b \quad (\text{A.16})$$

$$\varepsilon_t^g = \rho_g \varepsilon_{t-1}^g + \rho_{ga} \eta_t^a + \eta_t^g \quad (\text{A.17})$$

$$\varepsilon_t^i = \rho_i \varepsilon_{t-1}^i + \eta_t^i \quad (\text{A.18})$$

$$\varepsilon_t^r = \rho_r \varepsilon_{t-1}^r + \eta_t^r \quad (\text{A.19})$$

$$\varepsilon_t^p = \rho_p \varepsilon_{t-1}^p + \eta_t^p - \mu_p \eta_{t-1}^p \quad (\text{A.20})$$

$$\varepsilon_t^w = \rho_w \varepsilon_{t-1}^w + \eta_t^w - \mu_w \eta_{t-1}^w. \quad (\text{A.21})$$

The counterfactual no-rigidity prices and quantities evolve according to

$$\hat{y}_t^* = c_y \hat{c}_t^* + i_y \hat{l}_t^* + z_y \hat{z}_t^* + \varepsilon_t^g \quad (\text{A.22})$$

$$\begin{aligned} \hat{c}_t^* &= \frac{h/\gamma}{1+h/\gamma} \hat{c}_{t-1}^* + \frac{1}{1+h/\gamma} E_t \hat{c}_{t+1}^* + \frac{wl_c(\sigma_c - 1)}{\sigma_c(1+h/\gamma)} (\hat{l}_t^* - E_t \hat{l}_{t+1}^*) \\ &\quad - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} r_t^* - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} \varepsilon_t^b \end{aligned} \quad (\text{A.23})$$

$$\hat{i}_t^* = \frac{1}{1+\beta\gamma^{(1-\sigma_c)}} \hat{i}_{t-1}^* + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}} E_t \hat{i}_{t+1}^* + \frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})} \hat{q}_t^* + \varepsilon_t^i \quad (\text{A.24})$$

$$\hat{q}_t^* = \beta(1-\delta)\gamma^{-\sigma_c} E_t \hat{q}_{t+1}^* - r_t^* + (1-\beta(1-\delta)\gamma^{-\sigma_c}) E_t r_{t+1}^{k*} - \varepsilon_t^b \quad (\text{A.25})$$

$$\hat{y}_t^* = \Phi(\alpha k_t^{s*} + (1-\alpha)\hat{l}_t^* + \varepsilon_t^a) \quad (\text{A.26})$$

$$\hat{k}_t^{s*} = k_{t-1}^* + z_t^* \quad (\text{A.27})$$

$$\hat{z}_t^* = \frac{1-\psi}{\psi} \hat{r}_t^{k*} \quad (\text{A.28})$$

$$\hat{k}_t^* = \frac{(1-\delta)}{\gamma} \hat{k}_{t-1}^* + (1-(1-\delta)/\gamma) \hat{i}_t^* + (1-(1-\delta)/\gamma) \varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)}) \varepsilon_t^i \quad (\text{A.29})$$

$$\hat{w}_t^* = \alpha(\hat{k}_t^{s*} - \hat{l}_t^*) + \varepsilon_t^a \quad (\text{A.30})$$

$$\hat{r}_t^{k*} = \hat{l}_t^* + \hat{w}_t^* - \hat{k}_t^* \quad (\text{A.31})$$

$$\hat{w}_t^* = \sigma_l \hat{l}_t^* + \frac{1}{1-h/\gamma} (\hat{c}_t^* + h/\gamma \hat{c}_{t-1}^*). \quad (\text{A.32})$$

The steady state (ratios) that appear in the measurement equation or the log-linearized equilibrium conditions are given by

$$\gamma = \bar{\gamma}/100 + 1 \quad (\text{A.33})$$

$$\pi^* = \bar{\pi}/100 + 1 \quad (\text{A.34})$$

$$\bar{r} = 100(\beta^{-1}\gamma^{\sigma_c}\pi^* - 1) \quad (\text{A.35})$$

$$r_{ss}^k = \gamma^{\sigma_c}/\beta - (1-\delta) \quad (\text{A.36})$$

$$w_{ss} = \left(\frac{\alpha^\alpha(1-\alpha)^{(1-\alpha)}}{\Phi r_{ss}^{k\alpha}} \right)^{\frac{1}{1-\alpha}} \quad (\text{A.37})$$

$$i_k = (1-(1-\delta)/\gamma)\gamma \quad (\text{A.38})$$

$$l_k = \frac{1-\alpha}{\alpha} \frac{r_{ss}^k}{w_{ss}} \quad (\text{A.39})$$

$$k_y = \Phi l_k^{(\alpha-1)} \quad (\text{A.40})$$

$$i_y = (\gamma - 1 + \delta)k_y \quad (\text{A.41})$$

$$c_y = 1 - g_y - i_y \quad (\text{A.42})$$

$$z_y = r_{ss}^k k_y \quad (\text{A.43})$$

$$wl_c = \frac{1}{\lambda_w} \frac{1 - \alpha}{\alpha} \frac{r_{ss}^k k_y}{c_y}. \quad (\text{A.44})$$

The measurement equations take the form:

$$YGR_t = \bar{\gamma} + \hat{y}_t - \hat{y}_{t-1} \quad (\text{A.45})$$

$$INF_t = \bar{\pi} + \hat{\pi}_t$$

$$FFR_t = \bar{r} + \hat{R}_t$$

$$CGR_t = \bar{\gamma} + \hat{c}_t - \hat{c}_{t-1}$$

$$IGR_t = \bar{\gamma} + \hat{i}_t - \hat{i}_{t-1}$$

$$WGR_t = \bar{\gamma} + \hat{w}_t - \hat{w}_{t-1}$$

$$HOURS_t = \bar{l} + \hat{l}_t.$$

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: ORIGINAL 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 $p(\sigma|\nu, s) \propto \sigma^{-\nu-1} e^{-\nu s^2/2\sigma^2}$.

A.2 Leeper-Plante-Traum Fiscal Policy Model

The log-linearized equilibrium conditions of the Leeper, Plante, and Traum (2010) are given by:

$$\hat{u}_t^b - \frac{\gamma(1+h)}{1-h}\hat{C}_t + \frac{\gamma h}{1-h}\hat{C}_{t-1} - \frac{\tau^c}{1+\tau^c}\hat{\tau}_t^c \quad (\text{A.46})$$

$$= \hat{R}_t - \frac{\tau^c}{1+\tau^c}E_t\hat{\tau}_{t+1}^c + E_t u_{t+1}^b - \frac{\gamma}{1-h}E_t\hat{C}_{t+1}$$

$$\hat{u}_t^l + (1+\kappa)\hat{I}_t + \frac{\tau^c}{1+\tau^c}\hat{\tau}_t^c \quad (\text{A.47})$$

$$= \hat{Y}_t - \frac{\tau^l}{1+\tau^l}\hat{\tau}_t^l - \frac{\gamma}{1-h}\hat{C}_t + \frac{\gamma h}{1-h}\hat{C}_{t-1}$$

$$\begin{aligned} \hat{q}_t &= E_t\hat{u}_{t+1}^b - \frac{\gamma}{1-h}E_t\hat{C}_{t+1} + \frac{\gamma(1+h)}{1-h}\hat{C}_t - \frac{\tau^c}{1+\tau^c}E_t\tau_{t+1}^c - \hat{u}_t^b - \frac{\gamma h}{1-h}\hat{C}_{t-1} \\ &\quad + \frac{\tau^c}{1+\tau^c}\hat{\tau}_t^c + \beta(1-\tau^k)\alpha\frac{Y}{K}E_t\hat{Y}_{t+1} - \beta(1-\tau^k)\alpha\frac{Y}{K}\hat{K}_t - \beta\tau^k\alpha\frac{Y}{K}E_t\hat{\tau}_{t+1}^k \\ &\quad - \beta\delta_1E_t\hat{\nu}_{t+1} + \beta(1-\delta_0)E_t\hat{q}_{t+1} \end{aligned} \quad (\text{A.48})$$

$$Y_t = \frac{\tau^k}{1-\tau^k}\hat{\tau}_t^k + \hat{K}_{t+1} + \hat{q}_t + \left(1 + \frac{\delta_2}{\delta_0}\right)\hat{\nu}_t \quad (\text{A.49})$$

$$0 = \frac{1}{s''(1)}\hat{q}_t + (1-\beta)\hat{I}_t + \hat{I}_{t-1} + \beta E_t\hat{u}_t^i + \beta E_t\hat{u}_{t+1}^i \quad (\text{A.50})$$

$$Y\hat{Y}_t = C\hat{C}_t + G\hat{G}_t + I\hat{I}_t \quad (\text{A.51})$$

$$\hat{K}_t = (1-\delta_0)K_{t-1} + \delta_1\hat{\nu}_t + \delta_0I_t \quad (\text{A.52})$$

$$\begin{aligned} B\hat{B}_t + \tau^k\alpha Y(\hat{\tau}_t^k + \hat{Y}_t) + \tau^l(1-\alpha)Y(\hat{\tau}_t^l + \hat{Y}_t) + \tau^c C(\hat{\tau}_t^c + \hat{C}_t) \\ = \frac{B}{\beta}\hat{R}_{t-1} + \frac{B}{\beta}\hat{B}_{t-1} + G\hat{G}_t + Z\hat{Z}_t \end{aligned} \quad (\text{A.53})$$

$$\hat{Y}_t = \hat{u}_t^a + \alpha\nu_t + \alpha\hat{K}_{t-1} + (1-\alpha)\hat{L}_t. \quad (\text{A.54})$$

The fiscal policy rules and the law of motion of the exogenous shock processes are provided in the main text (Chapter 6.3).

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 Lubik and Schorfheide (2006). Here we detail the construction of the extended sample (2003:I to 2013:IV) for Chapter 8.6.

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,

$$\text{Per Capita Real Output Growth} = 100 \left[\ln \left(\frac{GDP_t}{POP_t} \right) - \ln \left(\frac{GDP_{t-1}}{POP_{t-1}} \right) \right].$$

2. **Annualized Inflation.** Take the CPI price level, (FRED mnemonic “CPIAUCSL”), call it CPI_t . Then,

$$\text{Annualized Inflation} = 400 \ln \left(\frac{CPI_t}{CPI_{t-1}} \right).$$

3. **Federal Funds Rate.** Take the effective federal funds rate (FRED mnemonic “FEDFUNDS”), call it FFR_t . Then,

$$\text{Federal Funds Rate} = FFR_t.$$

B.2 Smets-Wouters Model

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,

$$\text{Per Capita Real Output Growth} = 100 \left[\ln \left(\frac{GDP_t}{POP_t} \right) - \ln \left(\frac{GDP_{t-1}}{POP_{t-1}} \right) \right].$$

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

$$\begin{aligned} &\text{Per Capita Real Consumption Growth} \\ &= 100 \left[\ln \left(\frac{CONS_t}{GDPP_t POP_t} \right) - \ln \left(\frac{CONS_{t-1}}{GDPP_{t-1} POP_{t-1}} \right) \right]. \end{aligned}$$

3. **Per Capita Real Investment Growth.** Take the level of fixed private investment (FRED mnemonic “FPI”), call it INV_t . Then,

$$\begin{aligned} &\text{Per Capita Real Investment Growth} \\ &= 100 \left[\ln \left(\frac{INV_t}{GDPP_t POP_t} \right) - \ln \left(\frac{INV_{t-1}}{GDPP_{t-1} POP_{t-1}} \right) \right]. \end{aligned}$$

4. **Per Capita Real Wage Growth.** Take the BLS measure of compensation per hour for the nonfarm business sector (FRED mnemonic “COMPFB” / BLS series “PRS85006103”), call it W_t . Then

$$\text{Per Capita Real Wage Growth} = 100 \left[\ln \left(\frac{W_t}{GDPP_t} \right) - \ln \left(\frac{W_{t-1}}{GDPP_{t-1}} \right) \right].$$

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,

$$\text{Per Capita Hours} = 100 \ln \left(\frac{HOURS_t EMP_t}{POP_t} \right).$$

The series is then demeaned.

6. **Inflation.** Take the GDP price deflator, then

$$\text{Inflation} = 100 \ln \left(\frac{GDPP_t}{GDPP_{t-1}} \right).$$

7. **Federal Funds Rate.** Take the effective federal funds rate (FRED mnemonic “FED-FUNDS”), call it FFR_t . Then,

$$\text{Federal Funds Rate} = FFR_t/4.$$

B.3 Leeper-Plante-Traum Fiscal Policy Model

The data covers 1960:Q1 to 2008:Q1. The construction follows that of Leeper, Plante, and Traum (2010). Output data come from the NIPA; other sources are noted in the exposition. Each series has a (seperate) linear trend removed prior to estimation.

1. **Real Investment.** Take nominal personal consumption on durable goods (FRED mnemonic “PCDG”) call it $PCED_t$ and deflate it by the GDP deflator for personal consumption (FRED mnemonic “DPCERD3Q086SBEA”) call it P_t . Take the number of employed civilians (FRED mnemonic “CE16OV”), normalized so that its 1992Q3 value is 1, call it POP_t . Then,

$$\text{Real Investment} = 100 \ln \left(\frac{PCED_t/P_t}{POP_t} \right).$$

2. **Real Consumption.** Take nominal personal consumption on durable goods (FRED mnemonic “PCNG”) call it PCE_t and deflate it by the GDP deflator for personal consumption (FRED mnemonic “DPCERD3Q086SBEA”) call it P_t . Take the number of employed civilians (FRED mnemonic “CE16OV”), normalized so that its 1992Q3 value is 1, call it POP_t . Then,

$$\text{Real Consumption} = 100 \ln \left(\frac{PCE_t/P_t}{POP_t} \right).$$

3. **Real Hours worked.** 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,

$$\text{Real Hours Worked} = 100 \ln \left(\frac{HOURS_t EMP_t}{POP_t} \right).$$

4. **Real Consumption Tax Revenues.** Take Federal government current tax receipts from production and imports (FRED mnemonic “W007RC1Q027SBEA”), call it $CTAX_t$. Then,

$$\text{Real Consumption Tax Revenues} = 100 \ln \left(\frac{CTAX_t/P_t}{POP_t} \right).$$

5. **Real Labor Tax Revenues.** Take personal current tax revenues (FRED mnemonic “A074RC1Q027SBEA”), call it IT_t , take wage and salary accruals (FRED mnemonic “WASCUR”), call it W_t , and take proprietors’ incomes (FRED mnemonic “A04RC1Q027SBEA”), call it PRI_t . Take rental income (FRED mnemonic “RENTIN”), call it $RENT_t$, take corporate profits (FRED mnemonic “CPROFIT”), call it $PROF_t$, and take interest income (FRED mnemonic “W255RC1Q027SBEA”), call it INT_t . Define capital income, CI_t , as

$$CI_t = RENT_t + PROF_t + INT_t + PRI_t/2.$$

Define the average personal income tax rate as:

$$\tau_t^p = \frac{IT_t}{W_t + PRI_t/2 + CI_t}.$$

Take contributions for government social insurance (FRED mnemonic “W780RC1Q027SBEA”) call it CSI_t and take compensation for employees (FRED mnemonic “A57RC1Q027SBEA”) call it EC_t . Define the average labor income tax rate as

$$\tau_t^l = \frac{\tau_t^p(W_t + PRI_t) + CSI_t}{EC_t + PRI_t/2}.$$

Take the tax base $BASE_t = PCE_t + PCED_t$. Then

$$\text{Real Labor Tax Revenues} = 100 \ln \left(\frac{\tau_t^l BASE_t/P_t}{POP_t} \right).$$

6. **Real Capital Tax Revenues.** Take taxes on corporate income (FRED mnemonic “B075RC1Q027SBEA”) call it CT_t and take property taxes (FRED mnemonic “B249RC1Q027SBEA”) and call it PT_t . Define the average capital income tax rate as

$$\tau_t^k = \frac{\tau_t^p CI_t + CT_t}{CI_t + PT_t}.$$

Then

$$\text{Real Capital Tax Revenues} = 100 \ln \left(\frac{\tau_t^k BASE_t/P_t}{POP_t} \right).$$

7. **Real Government Expenditure.** Take government consumption expenditure (FRED mnemonic “FGEXPND”) call it GC_t . Take government gross investment (FRED mnemonic “A787RC1Q027SBEA”) call it GI_t . Take government net purchases of non-produced assets (FRED mnemonic “AD08RC1A027NBEA”) call it GP_t . Finally, take government consumption of fixed capital (FRED mnemonic “A918RC1Q027SBEA”) call it GCK_t . Define $G_t = GC_t + GI_t + GP_t - GCK_t$. Then

$$\text{Real Government Expenditure} = 100 \ln \left(\frac{G_t/P_t}{POP_t} \right).$$

8. **Real Government Transfers.** Take current transfer payments (FRED mnemonic “A084RC1Q027SBEA”) call $TRANSPAY_t$ and take current transfer receipts (FRED mnemonic “A577RC1Q027SBEA”) call it $TRANSREC_t$. Define net current transfers as

$$CURRTRANS_t = TRANSPAY_t - TRANSREC_t.$$

Take capital transfer payments (FRED mnemonic “W069RC1Q027SBEA”) call it $CAPTRANSPAY_t$ and take capital transfer receipts (FRED mnemonic “B232RC1Q027SBEA”) call it $CAPTRANSREC_t$. Define net capital transfers as

$$CAPTRANS_t = CAPTRANSPAY_t - CAPTRANSREC_t.$$

Take current tax receipts (FRED mnemonic “W006RC1Q027SBEA”), call it $TAXREC_t$, take income receipts on assets (FRED mnemonic “W210RC1Q027SBEA”), call it $INCREC_t$, and take the current surplus of government enterprises (FRED mnemonic “A108RC1Q027SBEA”), call it, $GOVSRP_t$. Define the total tax revenue, T_t , as the sum on consumption, labor, and capital tax revenues. Define the tax residual as

$$TAXRESID_t = TAXREC + INCREC + CSI_t + GOVSRP_t - T_t.$$

Define

$$TR_t = CURRTRANS_t + CAPTRANS_t - TAXRESID_t.$$

Then

$$\text{Real Government Transfers} = 100 \ln \left(\frac{TR_t/P_t}{POP_t} \right).$$

9. **Real Government Debt.** Take interest payments (FRED mnemonic “A091RC1Q027SBEA”), call it $INTPAY_t$. Define net borrowing as

$$NB_t = G_t + INT_t + TR_t - T_t.$$

Take the adjusted monetary base (FRED mnemonic “AMBSL”), take quarterly averages, then call it M_t . Then define government debt—starting in 1947—as

$$B_t = NB_t - \Delta M_t + B_{t-1}.$$

The value B_{1947Q1} is from Cox and Hirschhorn (1983). Then,

$$\text{Real Government Debt} = 100 \ln \left(\frac{B_t/P_t}{POP_t} \right)$$