Estimating Nonlinear Economic Models Using Surrogate Transitions

Matthew E. Smith*
Federal Reserve Board

First draft: July 8, 2010
This draft: October 3, 2011
Comments Welcome

*Federal Reserve Board, 20th and C Street NW, Washington, DC. 20551; Email: matthew.e.smith@frb.gov; Tel: (202) 452-3606. I would like to thank Tom Sargent for guidance during the early stages of this project, David Backus, Tim Cogley, Mark Gertler, Sydney Ludvigson and especially Arnaud Doucet for helpful comments on previous drafts. The views expressed in this paper, and all errors and omissions, should be regarded as those solely of the author and not necessarily those of the individuals listed above, the Federal Reserve Board of Governors or the Federal Reserve System.

Estimating Nonlinear Economic Models Using Surrogate Transitions

Abstract

We propose a novel combination of algorithms for jointly estimating parameters and unobservable states in a nonlinear state space system. We exploit an approximation to the marginal likelihood to guide a Particle Marginal Metropolis-Hastings algorithm. While this algorithm seemingly targets reduced dimension marginal distributions, it draws from a joint distribution of much higher dimension. The algorithm is demonstrated on a stochastic volatility model and a Real Business Cycle model with robust preferences.

1 Introduction

Before writing down the probability density function associated with an economic model, one needs to solve the functional equations that define an equilibrium. Typically, these functions are unavailable in closed form. If the solution is unavailable, so to is its associated likelihood function and one is left with an unfortunate dilemma. Solving a model using a linear or log-linear approximation, some higher order perturbation method, Chebyshev polynomials, or value function iteration will ultimately induce a different probability distribution over the same sequence. In this paper we propose combining such solution methods to aid in estimating the structural parameters of an economic model.

After the solution method is chosen, economic models have a state space representation. Estimating both parameters and unobservable states in nonlinear state space models is an active area of research. One technique recently proposed is Particle Markov Chain Monte Carlo (PMCMC) by Andrieu, Doucet, and Holenstein (2010). PMCMC involves embedding a particle filter within a Metropolis-Hastings algorithm to draw from the joint posterior distribution of parameters, θ, and unobservable states $X_{1:T}$ given the data $Y_{1:T}$. One algorithm in this class, the Particle Marginal Metropolis-Hastings algorithm (PMMH), uses a particle filter to approximate the marginal likelihood $p(Y_{1:T}|\theta)$ while drawing from the joint distribution of parameters and states. The main drawback of this algorithm is that the evaluation of the marginal likelihood using a particle filter is computationally intensive. In this paper, we approximate the marginal likelihood and use it to guide our PMMH algorithm. The approximate (surrogate) marginal likelihood guides the PMMH algorithm by pre-screening proposed parameter configurations. Only once the proposed set of parameters are accepted in this first step, do we attempt to evaluate the likelihood using a particle filter. While only computing marginal distributions, we show that this algorithm, like PMMH, targets a distribution of much higher dimension.

While focused on the estimation of nonlinear economic structures, the use of a surrogate marginal likelihood can be applied in a wide array of nonlinear state space systems. As such, we first demonstrate the algorithm on a stochastic volatility model. We then turn our attention to estimating

nonlinear Dynamic Stochastic General Equilibrium (DSGE) models, although doing so still falls under the broad category of creating surrogate marginal likelihoods. We fix the model of interest as one being solved with a third order perturbation solution. First, keeping fixed a nonlinear solution method, we use a marginal likelihood function computed using an Unscented Kalman filter as our surrogate marginal likelihood. Second, we linearly approximate a model then use a Kalman filter to evaluate the marginal likelihood. While only exploring these two types of approximations, the methodology proposed is flexible enough to handle many combination of approximate solution methods and surrogate likelihoods.[1]

When estimating DSGE models, one typically uses perturbation to solve for the policy and value functions, although our method could in practice be used with any solution method. If a linear solution is obtained, a Kalman filter can be used to evaluate the likelihood. Combining this technique with an optimization routine or embedding it within a Metropolis-Hastings algorithm has yielded much new insight into the US and other economies. See Sargent (1989) or more recently Smets and Wouters (2003), Christiano, Eichenbaum, and Evans (2005), Del Negro, Schorfheide, Smets, and Wouters (2007) and Justiniano and Primiceri (2008).

Sometimes one needs a nonlinear solution method to answer particular questions of interest, as in Fernández-Villaverde and Rubio-Ramírez (2007), Fernández-Villaverde, Guerrón-Quintana, Rubio-Ramírez, and Uribe (2009), Koijen, Fernández-Villaverde, Rubio-Ramírez, and van Binsbergen (2008) and Bidder and Smith (2010b). These models were solved using higher order perturbation methods which deliver quick and fairly accurate solutions. As a by-product, this solution method also delivers a linearized version of the model which until now has not been exploited as a computational aid when trying to estimate the full nonlinear model. As explored by Fernández-Villaverde and Rubio-Ramírez (2005), nonlinear solutions give more accurate estimates of the underlying structural parameters, but the authors did not explore combining the two approximations within one algorithm. When a linearly approximated model differs substantially from the nonlinear model of interest, an Extended or Unscented Kalman filter can be used to evaluate the surrogate marginal likelihood.

[1]For example, a surrogate likelihood computed using a lower tolerance level for convergence of value function iteration, or, finite elements with less elements than the model of interest.

Combining Extended or Unscented Kalman filters, and even the Surrogate Transition method, with particle filters has been done before, for example, Liu and Chen (1998), Doucet, Godsill, and Andrieu (2000) van der Merwe, de Freitas, Doucet, and Wan (2000), and Andreasen (2010). However, they have typically been used in the importance sampling step of particle filters and not as an over-arching approximation to the entire marginal likelihood. Further, given one has already programmed an Extended or Unscented Kalman filter, it requires only marginal additional effort to add it as a surrogate guide in the PMCMC routine.

The idea of embedding a particle filter within a Metropolis-Hastings algorithm comes from Fernández-Villaverde and Rubio-Ramírez (2007) in the context of estimating DSGE models. More recently, Andrieu, Doucet, and Holenstein (2010) prove convergence of such an algorithm and show that it targets a distribution of much higher dimension. They show that for any number of particles $N \geq 1$, the algorithm has the correct limiting distribution. The number of particles will affect the variance of the estimated marginal likelihood function and the speed of convergence to the invariant distribution. There are no special restrictions placed on the proposal distribution, $\theta' \sim q(\theta, \cdot)$ above the usual needed for convergence of an standard Metropolis-Hastings routine where one can exactly compute the marginal likelihood function.

The use of a surrogate when the target density is computationally demanding was suggested by Liu (2001),[2] who demonstrates that the transition kernel associated with surrogate transitions satisfies detailed balance with the target density.[3] Later, Christen and Fox (2005) suggest using a state dependent approximation to the true posterior distribution to estimate parameters in a Bayesian context.[4] Rasmussen (2003) approximates the posterior distribution with a normal distribution and uses that as a surrogate to guide a Hamiltonian Monte Carlo algorithm. Extending this idea, Fielding, Nott, and Liong (2011) embed an approximate density within a parallel tempering scheme for posterior distributions that are multimodal. While the above "surrogate" papers share a commonality with this paper, using an approximation but asymptotically drawing from the correct distribution, the algorithm presented in this paper differs by approximating (and comput-

[2] Chapter 9.4.3

[3] See also Efendiev, Hou, and Luo (2006).

[4] Cui, Fox, and O'Sullivan (2011) add an adaptive proposal distribution to this two-step algorithm.

ing) a marginal distribution but drawing from a much more complicated joint distribution as in Andrieu, Doucet, and Holenstein (2010). Alternatively, one could forgo the expensive computation altogether by approximating the target distribution and carrying out analysis using only this approximation.[5] While we focus on creating a surrogate marginal likelihood through the use of a Kalman or Unscented Kalman filter, one could use many methods to approximate $p(Y_{1:T}|\theta)$.

Our algorithm can be viewed as a special case of Andrieu, Doucet, and Holenstein (2010), with the insight that the transition kernel in the surrogate step can be used as a proposal density within a PMCMC framework. Alternatively, we show how to incorporate auxiliary variables within a surrogate transition framework, but only compute marginal distributions. Our description of the surrogate steps follows that of Liu (2001), Christen and Fox (2005), and Efendiev, Hou, and Luo (2006).

2 A Tour of Monte Carlo Algorithms for Bayesian Computation

In a Bayesian estimation context, Monte Carlo methods are used to draw from a posterior distribution of parameters given the data. Using draws from this posterior distribution, one can then compute various quantities of interest. See Robert and Casella (2004) for a thorough treatment of this subject. In some simple cases, one can compute and draw exactly from the posterior distribution. In many cases, this simple sampling is impossible and usually either an importance sampling or a Metropolis-Hastings algorithm is used.[6]

2.1 Importance Sampling

Suppose one wishes to calculate the following integral,

$$E_p[h(X)] = \int h(x)p(x)dx.$$

[5] See for example, Meyer, Fournier, and Berg (2003),Bliznyuk, Ruppert, Shoemaker, Regis, Wild, and Mugunthan (2008), Skaug and Yu (2008), and Frangos, Marzouk, Willcox, and van loeman Waanders (2010).

[6] See Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953) and Hastings (1970).

If one could draw X from the distribution $p(\cdot)$ with density $p(x)$, $X \sim p(\cdot)$, one could estimate the integral as,

$$\widehat{E_p[h(x)]} = \frac{1}{L}\sum_{m=1}^{L} h(x^{(m)}).$$

In cases when $p(\cdot)$ is hard to sample from, but there exists another distribution with density $g(x)$ which is easy to sample from, one could compute

$$E_p[h(x)] = \int h(x)\frac{p(x)}{g(x)}g(x)dx = E_g\left[h(x)\frac{p(x)}{g(x)}\right],$$

by drawing L times from $g(\cdot)$ then computing

$$\widehat{E_p[h(x)]} = \frac{1}{L}\sum_{m=1}^{L} h(x^{(m)})\frac{p(x^{(m)})}{g(x^{(m)})}.$$

When a natural candidate $g(\cdot)$ does not exist one can build an importance distribution sequentially.

2.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm seeks to reverse engineer a Markov chain whose invariant distribution is the distribution from which you want to sample from, $p(\cdot)$. The algorithm requires a density, $p(x)$ which one can compute up to a normalizing constant, and a proposal distribution $q(x,\cdot)$ with density $q(x,y)$, such that, given one is at some point $x^{(m)} = x$, one proposes a move $y \sim q(x,\cdot)$ and sets $x^{(m+1)} = y$ with probability

$$\alpha(x,y) = \min\left\{1, \frac{p(y)q(y,x)}{p(x)q(x,y)}\right\}$$

and $x^{(m+1)} = x$ with probability $(1-\alpha(x,y))$. Under some regularity conditions, this algorithm produces a Markov chain whose invariant distribution is $p(\cdot)$. The transition kernel of the Markov chain is

$$K(x,dy) = K(x,y)dy + (1-A(x))\delta_x(dy),$$

where $A(x) = \int K(x,y)dy$ is the probability of accepting a draw, conditional on being at x, $\delta_x(\cdot)$ is the point mass at x, $K(x,y) = \alpha(x,y)q(x,y), x \neq y$ and $K(x,x) = 0$.[7]

2.2.1 Two Common Examples

There are many choices of proposal densities $q(x,y)$. One common choice of proposal density is a random walk, where $q(x,\cdot) = N(x,\Sigma)$. We then have $q(x,y) = q(y,x)$ and the acceptance probability becomes

$$\alpha(x,y) = \min\left\{1, \frac{p(y)}{p(x)}\right\}.$$

When a good approximation to the distribution of interest is known, and is easy to sample from, one can set $q(x,\cdot) = q(\cdot)$, and propose new configurations from a fixed distribution. This algorithm is called an Independent Metropolis-Hastings algorithm, and the acceptance probability is

$$\alpha(x,y) = \min\left\{1, \frac{p(y)q(x)}{p(x)q(y)}\right\}.$$

2.3 The Surrogate Transition Method

Suppose that our target density, $p(x)$ is time consuming to calculate but that we have access to an approximation of $p(x)$, call it $\tilde{p}(x)$. Then the following algorithm, the Surrogate Transition Method (Liu (2001)), simulates draws from $p(\cdot)$.

Starting at some point $x^{(m)} = x$, we propose $y \sim \tilde{q}(x,\cdot)$ and set some intermediate point $z = y$ to with probability

$$\alpha_1(x,y) = \min\left\{1, \frac{\tilde{p}(y)\tilde{q}(y,x)}{\tilde{p}(x)\tilde{q}(x,y)}\right\},$$

and $z = x$ with probability $(1 - \alpha_1(x,y))$. Denote the transition kernel of this Markov chain as $S(x,\cdot)$.[8]

[7] See Tierney (1994) for more details.

[8] Since we are performing a Metropolis-Hastings update targeting $\tilde{p}(\cdot)$ with proposal distribution $\tilde{q}(x,\cdot)$, the transition kernel $S(x,\cdot)$ has the same form as in the previous section with p and q replaced by $\tilde{p}$ and $\tilde{q}$.

If $z = y$ after this accept/reject step, we set $x^{(m+1)} = y$ with probability

$$\alpha_2(x, y) = \min\left\{1, \frac{p(y)S(y, x)}{p(x)S(x, y)}\right\} = \min\left\{1, \frac{p(y)\tilde{p}(x)}{p(x)\tilde{p}(y)}\right\}.$$

If y is rejected at the first stage or second stage, we set $x^{(m+1)} = x$.

This algorithm produces Markov chain that has $p(\cdot)$ as its invariant distribution. Liu (2001) shows how to extend this idea to k intermediate steps. Let $S(x, \cdot)$ be a reversible Markov transition kernel with $\tilde{p}(x)$. That is, $S(x, y)$ and $\tilde{p}(x)$ satisfy the detailed balance condition,

$$\tilde{p}(x)S(x, y) = \tilde{p}(y)S(y, x).$$

Starting at some $x^{(m)}$, let $y_0 = x^{(m)}$, and sequentially draw $y_i \sim S(y_{i-1}, \cdot)$ k times. Let $S^k(x, \cdot)$ denote this k-step transition kernel. Set $x^{(m+1)} = y_k$ with probability

$$min\left\{1, \frac{p(y_k)S^k(y_k, x^{(m)})}{p(x^{(m)})S^k(x^{(m)}, y_k)}\right\} = min\left\{1, \frac{p(y_k)}{p(x^{(m)})}\frac{\tilde{p}(x^{(m)})}{\tilde{p}(y_k)}\right\}$$

and set $x^{(m+1)} = x^{(m)}$ otherwise. The surrogate transition method can be viewed as a special case of the Metropolis-Hastings algorithm where we use the surrogate Markov chain to generate proposals, $q(x, \cdot) = S^k(x, \cdot)$.[9]

2.4 Filtering and Inference in State Space Models

A state space model is made up of an unobservable sequence $\{X_t\}_{t=1}^T$, that evolves according to $X_t \sim p(\cdot|X_{t-1})$. Although we do not directly observe the sequence X_t, we do observe a sequence $\{Y_t\}_{t=1}^T$ which is related to the unobservable sequence as $Y_t|X_t \sim p(\cdot|X_t)$. It may be the case that the densities depend on a vector of parameters, θ that are also unknown, but, conditional on these parameters, we have expressions for p$(X_t|X_{t-1}, \theta)$, and $p(Y_t|X_t, \theta)$. In this joint estimation/signal extraction problem, we do not observe $X_{1:T}$, nor θ. We want to draw from the joint posterior distribution of states and parameters, with density $p(\theta, X_{1:T}|Y_{1:T})$. We can write the state space

[9]Some authors explicitly define the Metropolis-Hastings algorithm with $q(x, \cdot)$ being a Markov chain, for example, Roberts and Rosenthal (2004).

model as a transition equation

$$X_t = g(X_{t-1}, \epsilon_t, \theta),$$

and an observation equation

$$Y_t = f(X_t, \nu_t, \theta),$$

where $\epsilon_t \sim p(\cdot, \theta)$, and $\nu_t \sim p(\cdot, \theta)$. In the special case where the functions $g()$ and $f()$ are linear in (X_t, ϵ_t, ν_t) and ϵ_t and ν_t are normally distributed, the Kalman filter gives the optimal estimate of the posterior mean and variance of $p(dX_t|Y_{1:t}, \theta)$. When $g()$ and $f()$ are no longer linear or the disturbances are non-normal, approximations must be made.

A particle filter is a Sequential Monte Carlo technique that can approximate this solution, for a given θ, by drawing from a sequence of target densities, $\{p(X_{1:t}|Y_{1:t}, \theta)\}_{t=1}^T$. It is comprised of an initial distribution of $X_0 \sim p(\cdot, \theta)$ (and the ability to draw from that distribution), as well as two steps: importance sampling and resampling. The algorithm is as follows: Given a set of draws $\{X_{1:t-1}^i\}_{i=1}^N \sim p(dX_{1:t-1}|Y_{1:t-1}, \theta)$ with weights $\{W_{t-1}^i\}_{i=1}^N$

- draw $X_t^i \sim q(\cdot|X_{t-1}^i, Y_t)$
- compute weight $W_t^i = \frac{p(Y_t|X_t^i, \theta)p(X_t^i|X_{t-1}^i, \theta)}{q(X_t^i|X_{t-1}^i, Y_t, \theta)} W_{t-1}^i$
- resample $X_{1:t}^i = (X_{1:t-1}^i, X_t^i)$ with probability $\propto W_t^i$ and set $W_t^i = 1/N$

This algorithm produces a set of points $\{X_{1:t}^i\}_{t=1}^T$ that are distributed approximately according to $p(dX_{1:t}|Y_{1:t}, \theta)$ for each time period. When we use as our importance function $q(X_t|X_{t-1}, Y_t, \theta) = p(X_t|X_{t-1}, \theta)$ we have the Bootstrap filter (Gordon, Salmond, and Smith (1993)) and the weights are just proportional to the likelihood $W_t^i \propto p(Y_t|X_t^i)$. Because of the complex set of nested integrals in the marginal likelihood, $p(Y_{1:T}|\theta) = \prod_{t=1}^T p(Y_t|Y_{1:t-1}, \theta)$, one cannot compute this quantity exactly. This quantity can be estimated using a particle filter. When $X_{1:t}^i \sim p(dX_{1:t}|Y_{1:t})$, then by adding $X_{t+1}^i = g(X_t^i, \epsilon_{t+1})$ to $X_{1:t}^i$ to get $X_{1:t+1}^i$, we have a draw $X_{1:t+1}^i \sim p(dX_{1:t+1}|Y_{1:t}, \theta)$, and the

marginal likelihood can be estimated by

$$\begin{aligned}\hat{p}(Y_{1:T}|\theta) &= \prod_{t=1}^{T} p(Y_t|Y_{1:t-1},\theta) \\ &= \prod_{t=1}^{T} \int p(Y_t|X_t,\theta)p(X_t|Y_{1:t-1},\theta)dX_t \\ &= \prod_{t=1}^{T} \left(\frac{1}{N}\sum_{i=1}^{N} p(Y_t|X_t^i,\theta)\right).\end{aligned}$$

To perform inference in this nonlinear state space model, we can use the Particle Marginal Metropolis-Hastings algorithm of Andrieu, Doucet, and Holenstein (2010), which embeds a particle filter to evaluate (an approximation to) the marginal likelihood within a Metropolis-Hastings algorithm.

2.4.1 Particle Markov Chain Monte Carlo - the Particle Marginal Metropolis Hastings (PMMH) Algorithm

Given some $\theta^{(m)}, X_{1:T}^{(m)}$

- Draw $\theta' \sim q(\theta^{(m)}, \cdot)$
- Use a particle filter to evaluate $p(Y_{1:T}|\theta)$, call it $\hat{p}(Y_{1:T}|\theta)$
- Draw $X'_{1:T}$ from particles in particle filter
- Set $\theta^{(m+1)} = \theta'$, $X_{1:T}^{(m+1)} = X'_{1:T}$ with probability

$$\min\left\{1, \frac{\hat{p}(Y_{1:T}|\theta')p(\theta')q(\theta',\theta^{(m)})}{\hat{p}(Y_{1:T}|\theta^{(m)})p(\theta^{(m)})q(\theta^{(m)},\theta')}\right\}$$

As shown by Andrieu, Doucet, and Holenstein (2010), this algorithm produces draws $(\theta, X_{1:T}) \sim p(\cdot|Y_{1:T})$ and the marginal distribution of θ is obtained when the $X_{1:T}$ draws are discarded. The key insight is to view the algorithm as drawing from the joint distribution of parameters θ and a vector of random variables, u, that are used to evaluate the particle filter.[10] That is, write the estimate of

[10] See Beaumont (2003) and Andrieu and Roberts (2009) for earlier related work. Our characterization of Andrieu, Doucet, and Holenstein (2010) follows that of Flury and Shephard (2008) and Silva, Giordani, Kohn, and Pitt (2009)

the marginal likelihood evaluated using the particle filter as $\hat{p}(Y_{1:T}|\theta) = p(Y_{1,T}|\theta, u)$ with,

$$\int p(Y_{1:T}|\theta, u)p(u|\theta)du = p(Y_{1:T}|\theta).$$

Then write the proposal density $q(\{\theta, u\}, \{\theta', u'\})$ as,

$$q(\{\theta, u\}, \{\theta', u'\}) = q(\theta, \theta')p(u'|\theta').$$

The acceptance probability is,

$$\begin{aligned}
\alpha(\{\theta, u\}, \{\theta', u'\}) &= \min\left\{1, \tfrac{p(\theta',u'|Y_{1:T})q(\{\theta',u'\},\{\theta,u\})}{p(\theta,u|Y_{1:T})q(\{\theta,u\},\{\theta',u'\})}\right\} \\
&= \min\left\{1, \tfrac{p(\theta',u'|Y_{1:T})q(\theta',\theta)p(u|\theta)}{p(\theta,u|Y_{1:T})q(\theta,\theta')p(u'|\theta')}\right\} \\
&= \min\left\{1, \tfrac{p(Y_{1:T}|\theta',u')p(\theta',u')q(\theta',\theta)p(u|\theta)}{p(Y_{1:T}|\theta,u)p(\theta,u)q(\theta,\theta')p(u'|\theta')}\right\} \\
&= \min\left\{1, \tfrac{p(Y_{1:T}|\theta',u')p(u'|\theta')p(\theta')q(\theta',\theta)p(u|\theta)}{p(Y_{1:T}|\theta,u)p(u|\theta)p(\theta)q(\theta,\theta')p(u'|\theta')}\right\} \\
&= \min\left\{1, \tfrac{p(Y_{1:T}|\theta',u')p(\theta')q(\theta',\theta)}{p(Y_{1:T}|\theta,u)p(\theta)q(\theta,\theta')}\right\} \\
&= \min\left\{1, \tfrac{\hat{p}(Y_{1:T}|\theta')p(\theta')q(\theta',\theta)}{\hat{p}(Y_{1:T}|\theta)p(\theta)q(\theta,\theta')}\right\}.
\end{aligned}$$

Thus, this Particle Marginal Metropolis Hastings algorithm is actually a Metropolis-Hastings algorithm on an extended space (θ, u).

3 PMMH with Surrogate Transitions

In this section, we suggest a specific proposal density for our parameters, namely one in which uses a simple to compute likelihood function, $\tilde{p}(Y_{1:T}|\theta)$ to perform a surrogate transition step within the PMMH algorithm. Our algorithm is as follows: Given some $\theta^{(m)}, X_{1:T}^{(m)}$

- sample $\theta' \sim q(\theta^{(m)}, \cdot)$
- set $z = \theta'$ with probability,

$$\alpha_1(\theta^{(m)}, \theta') = min\left\{1, \frac{\tilde{p}(\theta'|Y_{1:T})q(\theta', \theta^{(m)})}{\tilde{p}(\theta^{(m)}|Y_{1:T})q(\theta^{(m)}, \theta')}\right\}$$

- If $z \neq \theta^{(m)}$,
  - run a particle filter
  - sample $X'_{1:T}$ from the particles in the particle filter
  - set $(\theta^{(m+1)}, X_{1:T}^{(m+1)}) = (\theta', X'_{1:T})$ with probability

$$\begin{aligned} \alpha_2(\theta^{(m)}, \theta') &= min\left\{1, \frac{\hat{p}(\theta'|Y_{1:T})\tilde{p}(\theta^{(m)}|Y_{1:T})}{\hat{p}(\theta^{(m)}|Y_{1:T})\tilde{p}(\theta'|Y_{1:T})}\right\} \\ &= min\left\{1, \frac{\hat{p}(Y_{1:T}|\theta')p(\theta')\tilde{p}(Y_{1:T}|\theta^{(m)})\tilde{p}(\theta^{(m)})}{\hat{p}(Y_{1:T}|\theta^{(m)})p(\theta^{(m)})\tilde{p}(Y_{1:T}|\theta')\tilde{p}(\theta')}\right\} \end{aligned}$$

- Else set $(\theta^{(m+1)}, X_{1:T}^{(m+1)}) = (\theta^{(m)}, X_{1:T}^{(m)})$

Even though the above was motivated as a surrogate step only in the θ dimension, it is actually a surrogate transition (which is a particular Metropolis-Hastings update) in the larger space (θ, u). The appendix formally sets up and proves this result, but below we sketch the main ideas.

Define the surrogate target density as $\tilde{p}(\theta, u|Y_{1:T}) \propto \tilde{p}(Y_{1:T}|\theta, u)p(u|\theta)\tilde{p}(\theta)$, where $\tilde{p}(Y_{1:T}|\theta, u)$ is an approximation to the likelihood function, $p(u|\theta)$ is the density of the vector of random variables in u that are used to evaluate the particle filter given our vector of structural parameters θ, and $\tilde{p}(\theta)$ is some alternate density of our parameters θ, not necessarily equal to the prior. We will create an approximate likelihood function does not depend on the vector u, or that $\tilde{p}(Y_{1:T}|\theta, u) = \tilde{p}(Y_{1:T}|\theta)$. Further, we assume that the support of the surrogate target density contains the support of the target density,

$$supp\left\{p(\theta, u|Y_{1:T})\right\} \subset supp\left\{\tilde{p}(\theta, u|Y_{1:T})\right\}.$$

This guarantees that we do not automatically reject any point in the support of $p(\theta, u|Y_{1:T})$.

The proposal density for the surrogate transition step is,

$$\tilde{q}(\{\theta, u\}, \{\theta', u'\}) = q(\theta, \theta')p(u'|\theta').$$

The acceptance probability in the surrogate chain becomes,

$$\begin{aligned}
\alpha_1(\{\theta,u\},\{\theta',u'\}) &= \min\left\{1, \tfrac{\tilde{p}(\theta',u'|Y_{1:T})q(\{\theta',u'\},\{\theta,u\})}{\tilde{p}(\theta,u|Y_{1:T})q(\{\theta,u\},\{\theta',u'\})}\right\} \\
&= \min\left\{1, \tfrac{\tilde{p}(Y_{1:T}|\theta',u')p(u'|\theta')\tilde{p}(\theta')q(\{\theta',u'\},\{\theta,u\})}{\tilde{p}(Y_{1:T}|\theta,u)p(u|\theta)\tilde{p}(\theta)q(\{\theta,u\},\{\theta',u'\})}\right\} \\
&= \min\left\{1, \tfrac{\tilde{p}(Y_{1:T}|\theta',u')p(u'|\theta')\tilde{p}(\theta')q(\theta',\theta)p(u|\theta)}{\tilde{p}(Y_{1:T}|\theta,u)p(u|\theta)\tilde{p}(\theta)q(\theta,\theta')p(u'|\theta')}\right\} \\
&= \min\left\{1, \tfrac{\tilde{p}(Y_{1:T}|\theta',u')\tilde{p}(\theta')q(\theta',\theta)}{\tilde{p}(Y_{1:T}|\theta,u)\tilde{p}(\theta)q(\theta,\theta')}\right\} \\
&= \min\left\{1, \tfrac{\tilde{p}(Y_{1:T}|\theta')\tilde{p}(\theta')q(\theta',\theta)}{\tilde{p}(Y_{1:T}|\theta)\tilde{p}(\theta)q(\theta,\theta')}\right\}.
\end{aligned}$$

By having our target and proposal density including $p(u|\theta)$, and our approximation of the marginal likelihood a deterministic function of θ, we can accept or reject the pair (θ, u) without reference to u.

The transition kernel of the Markov Chain constructed using this Metropolis-Hastings update with $\tilde{p}(\theta, u|Y_{1:T})$ as a target density and proposal distribution $\tilde{q}(\{\theta,u\},\cdot)$ is,

$$S(\{\theta,u\},\cdot) = \alpha(\{\theta,u\},\cdot)\tilde{q}(\{\theta,u\},\cdot) + (1 - A(\theta,u))\delta_{\theta,u}(\cdot).$$

By construction, our kernel satisfies detailed balance with $\tilde{p}(\theta, u|Y_{1:T})$, that is,

$$\begin{aligned}
\tilde{p}(\theta,u|Y_{1:T})S(\{\theta,u\},\{\theta',u'\}) &= S(\{\theta',u'\},\{\theta,u\})\tilde{p}(\theta',u'|Y_{1:T}) \\
\tilde{p}(Y_{1:T}|\theta,u)p(u|\theta)\tilde{p}(\theta)S(\{\theta,u\},\{\theta',u'\}) &= S(\{\theta',u'\},\{\theta,u\})\tilde{p}(Y_{1:T}|\theta',u')p(u'|\theta')\tilde{p}(\theta'),
\end{aligned}$$

or alternately,

$$\tfrac{\tilde{p}(Y_{1:T}|\theta,u)p(u|\theta)\tilde{p}(\theta)}{\tilde{p}(Y_{1:T}|\theta',u')p(u'|\theta')\tilde{p}(\theta')} = \tfrac{S(\{\theta',u'\},\{\theta,u\})}{S(\{\theta,u\},\{\theta',u'\})}.$$

If we define as a proposal distribution,

$$q(\{\theta,u\},\cdot) = S(\{\theta,u\},\cdot),$$

we have an acceptance probability for $\theta' \neq \theta$ of,

$$\begin{aligned}
\alpha_2(\{\theta,u\},\{\theta',u'\}) &= \min\left\{1, \frac{p(\theta',u'|Y_{1:T})q(\{\theta',u'\},\{\theta,u\})}{p(\theta,u|Y_{1:T})q(\{\theta,u\},\{\theta',u'\})}\right\} \\
&= \min\left\{1, \frac{p(\theta',u'|Y_{1:T})S(\{\theta',u'\},\{\theta,u\})}{p(\theta,u|Y_{1:T})S(\{\theta,u\},\{\theta',u'\})}\right\} \\
&= \min\left\{1, \frac{p(Y_{1:T}|\theta',u')p(u'|\theta')p(\theta')S(\{\theta',u'\},\{\theta,u\})}{p(Y_{1:T}|\theta,u)p(u|\theta)p(\theta)S(\{\theta,u\},\{\theta',u'\})}\right\} \\
&= \min\left\{1, \frac{p(Y_{1:T}|\theta',u')p(u'|\theta')p(\theta')\tilde{p}(Y_{1:T}|\theta,u)p(u|\theta)\tilde{p}(\theta)}{p(Y_{1:T}|\theta,u)p(u|\theta)p(\theta)\tilde{p}(Y_{1:T}|\theta',u')p(u'|\theta')\tilde{p}(\theta')}\right\} \\
&= \min\left\{1, \frac{p(Y_{1:T}|\theta',u')p(\theta')\tilde{p}(Y_{1:T}|\theta,u)\tilde{p}(\theta)}{p(Y_{1:T}|\theta,u)p(\theta)\tilde{p}(Y_{1:T}|\theta',u')\tilde{p}(\theta')}\right\} \\
&= \min\left\{1, \frac{\hat{p}(Y_{1:T}|\theta')p(\theta')\tilde{p}(Y_{1:T}|\theta)\tilde{p}(\theta)}{\hat{p}(Y_{1:T}|\theta)p(\theta)\tilde{p}(Y_{1:T}|\theta')\tilde{p}(\theta')}\right\}.
\end{aligned}$$

If we have rejected (θ', u') in the surrogate phase, that is, $\theta' = \theta$, we do not need to re-run the particle filter. When $\theta' \neq \theta$ at the end of the surrogate phase, we then need to evaluate the particle filter and compute what the associated u' is. That is, we can delay computation of u' until we evaluate the particle filter.

4 Discussion

4.1 Balancing Statistical and Computational Efficiency

Dynamic samplers, such as the Metropolis-Hastings algorithm, generate correlated draws that are approximately distributed according to the target distribution. Thus the estimate of

$$E_p[h(X)] = \int h(x)p(x)dx,$$

by

$$\hat{h} = \frac{1}{L}\sum_{i=1}^{L} h(x^{(i)}),$$

has a variance

$$var(\hat{h}) = var_p(h(X))\frac{\tau}{L},$$

where τ is the integrated autocorrelation time defined as

$$\tau = 1 + 2\sum_{j=1}^{\infty} \rho_j.$$

Here, ρ_j is the correlation between $h(X_t)$ and $h(X_{t-j})$.[11] The goal of designing good dynamic samplers is to design Markov chains with low τ. The closer τ is to unity, the closer the variance of the Monte Carlo estimate of the quantity h is to one using independent draws from $p(\cdot)$. However, it is only fair to use τ to compare two algorithms if they take the same time to compute the same number of draws. Thus a natural extension is to look at the seconds per effectively independent sample. Let τ^{rt} be the running time in seconds to produce a chain of length L. The seconds per effectively independent draw of a chain of length L is computed as,

$$s/Eff = \frac{\tau^{rt}\tau}{L}.$$

The speedup from using algorithm 2 relative to a base algorithm, 1, is then the ratio of the seconds per effectively independent draw of each algorithm,

$$Speedup = \frac{\tau_1^{rt}\tau_1}{\tau_2^{rt}\tau_2}.$$

We can see two forces playing a role in the computed speedup. By foregoing filter evaluations on points which have a very small probability of acceptance, we can decrease the running time of our algorithm. By sampling multiple times from our surrogate density, or designing better samplers in the surrogate step, we can potentially decrease τ_2 relative to τ_1. If we consider the running time of a PMMH algorithm of length L,[12]

$$\tau_{MH}^{rt} \approx (\text{seconds to evaluate particle filter}) * L,$$

[11] In our experiments, we estimate τ by the following, $\hat{\tau} = 1 + 2\sum_{j=1}^{2000}(1 - \frac{j}{2000})\hat{\rho}_j$.

[12] This assumes that the time to draw from and compute $q(x, y)$ is negligible.

whereas for the PMMH with surrogate transitions, with K surrogate steps,

$$\begin{aligned}\tau_{ST}^{rt} \approx &(\text{seconds to evaluate surrogate likelihood}) * K * L \\ &+ (\text{seconds to evaluate particle filter}) * (\text{average probability of move from surrogate}) * L\end{aligned}$$

For DSGE models that requires one to both solve for the various policy functions and evaluate the likelihood, the running times become,

$$\tau_{MH}^{rt} \approx (\text{seconds to solve full model} + \text{seconds to evaluate particle filter}) * L,$$

and,

$$\begin{aligned}\tau_{ST}^{rt} \approx &(\text{seconds to solve surrogate model} + \text{seconds to evaluate surrogate likelihood}) * K * L \\ &+ (\text{seconds to solve full model} + \text{seconds to evaluate particle filter}) \\ &* (\text{average probability of move from surrogate}) * L\end{aligned}$$

From this decomposition of running times, we can see where the surrogate method might see advantages relative to the Metropolis-Hastings algorithm, as well as how one might be able to design surrogates to decrease the running times. Analysis of the ratio of integrated autocorrelation times is less clear. Following the insights contained in Cui, Fox, and O'Sullivan (2011), consider moving from a point x to y obtained from a 1 step surrogate move. Because our algorithm uses two accept/reject tests, it will accept less often than if we were able to directly sample from our surrogate model,

$$\alpha_1(x, y)\alpha_2(x, y) \leq \alpha_2(x, y).$$

Further we have that,

$$\begin{aligned}\alpha_1(x,y)\alpha_2(x,y) &= \min\left\{1, \frac{\tilde{p}(y)\tilde{q}(y,x)}{\tilde{p}(x)\tilde{q}(x,y)}\right\} \min\left\{1, \frac{p(y)\tilde{p}(x)}{p(x)\tilde{p}(y)}\right\} \\ &\leq \min\left\{1, \frac{p(y)\tilde{q}(y,x)}{p(x)\tilde{q}(x,y)}\right\} \\ &= \alpha_{MH}(x,y)\end{aligned}$$

Thus multiple acceptance steps, while required to maintain detailed balance, decreases the acceptance probability relative to sampling directly from $\tilde{p}(\cdot)$ as well as a Metropolis-Hastings algorithm with target $p(\cdot)$ and proposal distribution $q(x,\cdot)$. When $\alpha_1(x,y)$ and $\alpha_2(x,y)$ both agree on the same move from x to y in the sense of $\alpha_1(x,y) = \alpha_2(x,y) = 1$ or, $\alpha_1(x,y) < 1, \alpha_2(x,y) < 1$, we have $\alpha_1(x,y)\alpha_2(x,y) = \alpha_{MH}(x,y)$. From this we can see that the closer our surrogate $\tilde{p}(\cdot)$ is to our model of interest, $p(\cdot)$, the closer the 1 step surrogate algorithm mimics a Metropolis Hastings algorithm with the proposal distribution $\tilde{q}(x,\cdot)$ and target $p(\cdot)$. By avoiding expensive likelihood evaluations, an accurate surrogate should be able to produce draws faster than such a Metropolis algorithm with minimal loss from higher integrated autocorrelation times by requiring multiple acceptance steps. Inefficiencies arise when proposed moves from x to y, are rejected in the first step, but certainly accepted in the second, $\alpha_1(x,y) < 1$ and $\alpha_2(x,y) = 1$. In words, if y was proposed sampling directly from $\tilde{p}(\cdot)$, we would move to y, but since we are not drawing directly from $\tilde{p}(\cdot)$, we may reject this move before having an opportunity to test it with $\alpha_2(x,y)$. If we focus on symmetric surrogate proposal densities, $q(x,y) = q(y,x)$, this happens when moving from x to y is costly in terms of $\tilde{p}(\cdot)$ but favored (strongly) according to $p(\cdot)$. When $\alpha_1(x,y) = 1$, but $\alpha_2(x,y) < 1$, that is, our surrogate phase accepts the move with certainty, but the second step rejects the point with some probability, we have, $\alpha_1(x,y)\alpha_2(x,y) = \alpha_2(x,y)$, and we accept the move with the same probability as if we were sampling from $\tilde{p}(\cdot)$. We expect that the surrogate method will have a larger integrated autocorrelation time when only one update is used and speedup (if any) will likely come from a lower computational time. When multiple surrogate steps are performed with a surrogate which closely matches our target, we would expect lower correlation in our Markov chain and similar computational time.

4.2 Tuning

There are many aspects of this algorithm that one can choose, such as the choice of the surrogate model, choice of proposal density within the surrogate phase and number of steps. Rather than searching for an optimal surrogate proposal, we will strive to learn in what circumstances can using the method improve performance over a random walk sampler.

We have seen that conditional on reaching the second stage, the acceptance probability matches that of an Independent Metropolis-Hastings algorithm. A necessary and sufficient condition the Independent Metropolis-Hastings algorithm to sample approximately from $p(\theta, u|Y_{1:T})$ is that the support of proposal distribution contains that of the target distribution.[13] That is, $p(\theta, u|Y_{1:T}) > 0 \rightarrow \tilde{p}(\theta, u|Y_{1:T}) > 0$, which we have assumed for our surrogate transitions sampler. However, the rate of convergence of the IMH algorithm depends on the tails of the two densities through the ratio $\frac{p(x)}{\tilde{p}(x)}$. We use this insight to alleviate potential problems in the tails our our surrogate distribution, we can scale our surrogate log density by a temperature term, T. That is, we can use $\tilde{p}^{\frac{1}{T}}(\theta|Y_{1:T}) \propto \tilde{p}(\theta|Y_{1:T})^{\frac{1}{T}}$ as our surrogate.[14] Increasing the temperature increases the chance that the algorithm accepts moves which would normally have been rejected, thus over sampling the tails relative to sampling from $\tilde{p}(\theta, u|Y_{1:T})$. Following the discussion in the previous section, this should decrease the cases when $\alpha_1(x, y) < 1$, and $\alpha_2(x, y) = 1$, but the proposal was rejected using $\alpha_1(x, y)$.

Although motivated as an improvement for estimating parameters in a Bayesian context through a simpler to compute likelihood, the surrogate target density has another degree of freedom, its priors. While our forthcoming examples will use $\tilde{p}(\theta) = p(\theta)$, one can choose a surrogate prior density $\tilde{p}(\theta)$ that differs from $p(\theta)$. This might be useful when the surrogate likelihood is nearly flat, or differs substantially from the target likelihood in some dimension.

The number of intermediate steps in the surrogate transition phase will have two effects. First, increasing the number of intermediate steps will increase the probability when we actually run the expensive particle filter. Second, more intermediate steps decreases the dependence of that draw from the current state. If the approximations to the true marginal likelihood are accurate, one can

[13] See Robert and Casella (2004) for a discussion of the IMH algorithm and convergence properties.

[14] Explicitly taking into account the random variables u, the surrogate would be $\tilde{p}^{\frac{1}{T}}(\theta, u|Y_{1:T}) \propto (\tilde{p}(Y_{1:T}|\theta)\tilde{p}(\theta))^{\frac{1}{T}} p(u|\theta)$.

expect to accept the proposal value with probability one as the acceptance criteria is,

$$\min\left\{1, \frac{\hat{p}(Y_{1:T}|\theta')}{\tilde{p}(Y_{1:T}|\theta')}\frac{\tilde{p}(Y_{1:T}|\theta^{(m)})}{\hat{p}(Y_{1:T}|\theta^{(m)})}\right\} \quad = \quad \min\left\{1, \approx 1\right\}.$$

However, in cases where $\tilde{p}(Y_{1:T}|\theta)$ is crude and perhaps tempered, the acceptance probability will not be close to one.

In the examples that follow, we use a random walk proposal distribution in the surrogate phase. Following the results of the optimal tuning literature (Roberts and Rosenthal (2001)), we choose $q(\theta, \cdot) = N(\theta, \Sigma')$ where $\Sigma' = \frac{2.38^2}{d}\Sigma$, with Σ an estimate of the true covariance of the resulting posterior distribution and d is the dimension of θ. In situations where the surrogate likelihood differs from the likelihood of interest and substantial differences in computational time exist, it is not clear what sort of acceptance probability one should strive for in the surrogate phase. As we will keep the proposal distribution the same, even as we temper the surrogate density, the probability of accepting a draw in the surrogate phase will increase as the temperature rises. Since we are choosing the surrogate likelihoods that are fast to compute, one could design more elaborate samplers in the surrogate transition step as opposed to a simple random walk, allowing for example, tempered or HMC transitions as in Rasmussen (2003) and Fielding, Nott, and Liong (2011). Lastly, computational resources will determine the effectiveness of using a surrogate as a guide. Each chain was run on a single core of a Xeon X5670 processor, 2.93GHz with 12MB Cache.[15] Faster processors that decreases running times of the particle filter relative to the surrogate model decreases the value of forgoing filter evaluations.[16]

[15] Previous drafts of this paper included results from Xeon E5345 and E5410 (2.33GHz) processors performed while a graduate student at NYU.

[16] We omit sampling and storing the unobservable series $X_{1:T}$.

5 Examples

5.1 A stochastic volatility model

Consider the stochastic volatility model,

$$\begin{aligned} Y_t &= e^{(\frac{\mu+X_t}{2})}\nu_t \\ X_t &= \rho X_{t-1} + \tau\epsilon_t. \end{aligned}$$

Here, $\{Y_t\}_{t=1}^T$ is a sequence of observables, say asset returns or log consumption growth, and $\{X_t\}_{t=1}^T$ is an unobservable component that dictates the current period's volatility. The goal is to estimate the sequence of unobservables $X_{1:T}$ as well as the vector of parameters $\theta = (\mu, \rho, \tau)$. The likelihood function for this stochastic volatility model is unavailable in closed form. We form the surrogate model by first squaring and taking logarithms of the observation equation to get the following state space system,

$$\begin{aligned} log(Y_t^2) &= \mu + X_t + log(\nu_t^2) \\ X_t &= \rho X_{t-1} + \tau\epsilon_t. \end{aligned}$$

While linear, the term $log(v_t^2)$ no longer is normally distributed, but has mean approximately -1.27 and variance 4.93. Here we replace $log(v_t^2)$ with a normally distributed term $\hat{v}_t \sim N(-1.27, 4.93)$. Also, for notational sake, let $\hat{Y}_t = log(Y_t^2)$ and form the linear Gaussian state space system:

$$\begin{aligned} \hat{Y}_t &= \mu + X_t + \hat{\nu}_t \\ X_t &= \rho X_{t-1} + \tau\epsilon_t \end{aligned}$$

Note that one could potentially maximize the likelihood and estimate θ using (say) a Laplace approximation to evaluate the likelihood following Meyer, Fournier, and Berg (2003). Instead, we use this as an surrogate marginal likelihood. To see the surrogate transition method in action, we simulate the model for 1000 time periods at parameter values $\mu = 1$, $\rho = 0.9$, $\tau = 0.5$. We use 5000 particles in the evaluation of the particle filter, and run our algorithm to generate 100000 draws. We run our surrogate transitions algorithms for various levels of the temperature, T, from 1 to 4, and number of intermediate steps, K, from 1 to 6. The constant across algorithms covariance

matrix of the random walk chain was computed as the covariance matrix of an initial Random Walk Metropolis-Hastings chain.[17]

5.1.1 Results

We first examine the results from our standard Random Walk Metropolis-Hastings chain. We compute the acceptance rate of the chain, the number of seconds it took to run the 100000 draws, as well as the integrated autocorrelation time, and finally the number of seconds per effectively independent sample from the chain.

[INSERT TABLE 1 ABOUT HERE]

As we see in table 1, the acceptance rate for our Random Walk Metropolis chain is close to the oft cited optimal 0.234. As discussed in Roberts and Rosenthal (2001), an acceptance rate of about 25% maximizes the efficiency of the Markov chain for certain target distributions. However, for acceptance rates between $15 - 40\%$ the chain is still upwards of 80% efficient.[18]

We take the draws from the posterior distribution and calculate the integrated autocorrelation time for mean estimate of the parameters. The numbers range from $10.8 - 12.6$, meaning that the variance of the estimate is $10.8 - 12.6$ times larger than if we had independent draws from the posterior. Alternatively, the variance of the estimate is about the same as if we only had a little under 10000 independent draws from the posterior distribution. Combining these results with the running time, we compute the time per effectively independent sample. As we see, the numbers range from $2.5 - 3$ seconds for each variable.

[INSERT TABLE 2 ABOUT HERE]

As we see in table 2, for our surrogate transition chain, the acceptance rate varies from a low of 0.2 to a high of 0.4, depending on the temperature (T) and number of intermediate steps (K) used. The lower acceptance rates come from combinations of temperature/intermediate steps (T, K) that

[17] While we refer to this as Random Walk Metropolis-Hastings, or just RWMH, we could alternatively call it the PMMH with a random walk proposal density.

[18] Also note that the 0.234 number is an asymptotic result for a large parameter space for certain target distributions. In one dimension, the actual optimal acceptance rate is closer to 50%.

are either both high or both low. This is most likely due to differences in the surrogate and true posterior distribution.

[INSERT TABLE 3 ABOUT HERE]

Our surrogate chain runs anywhere from a little more then twice as fast when $(T, K) = (1, 1)$, to slightly longer for T and K both high. The decrease in running time is due to the avoidance of the relatively more expensive particle filter evaluation. When T and K are both low, the probability of rejecting the intermediate point it quite high and we avoid the expensive evaluation. With K and T higher, we most likely move far away from the starting point and almost certainly evaluate the particle filter.

The overall effect of the surrogate method depends on the integrated autocorrelation time, or how correlated the draws from our posterior distribution are. As we see, for low K we actually have a larger integrated autocorrelation time than the Metropolis chain. This drops substantially as K is increased to 5 and 6. For low K, any speedup with then be achieved through speedup of the running time rather than a speedup from a lower variance of the resulting estimate. For high K the opposite is true and the speedup will be due to lower integrated autocorrelation time.

[INSERT TABLE 4 ABOUT HERE]

[INSERT TABLE 5 ABOUT HERE]

[INSERT TABLE 6 ABOUT HERE]

To compare the surrogate transitions chain to the Random Walk Metropolis-Hastings chain, we compute the seconds per effectively independent draw of each of the algorithms. We then take the ratio of this time of the PMMH and divide it by that of PMMH with Surrogate Transitions to get the speedup (or slowdown) of the new algorithm. For example, the number 2 would state that surrogate transitions produces effectively independent samples twice as fast as the RWMH algorithm, or that our new algorithm is 100% more efficient.

As we can see, for this very simplistic model where the computation of the likelihood via particle filtering is quick and not many particles are needed to get an accurate approximation, we still see

speedups. Rather than searching for an optimal (T, K) combination, we observe that the numbers, while having a large range $(1.11 - 3.10)$, are all bigger than 1, and on average are quite substantial, about 1.7 averaging over the parameters and (T, K) combinations.

5.2 An Asset Pricing RBC Model with Robust Preferences

Here we estimate a simple real business cycle model with an agent who fears his model is misspecified. In Koijen, Fernández-Villaverde, Rubio-Ramírez, and van Binsbergen (2008), the authors estimate a slightly richer model with Epstein and Zin (1989) preferences using both macro and yield curve data. The authors do maximum likelihood estimation but suggest that one could use a Random Walk Metropolis-Hastings algorithm to estimate parameters in a Bayesian setting. We demonstrate the benefits of PMMH with Surrogate Transitions over a random walk algorithm on a similar model using simulated data.

Agent's utility satisfy the risk sensitive recursion,

$$U_t = (1-\beta)(log(C_t)) + \frac{\beta}{1-\gamma} log\left(E[e^{(1-\gamma)U_{t+1}}]\right).$$

Here, C_t is consumption of the agent at time t, β is the discount factor, and γ is a parameter governing either risk aversion (Tallarini (2000)), or aversion to model uncertainty (Hansen and Sargent (2007), Barillas, Hansen, and Sargent (2009), Bidder and Smith (2010a) and Bidder and Smith (2010b)).

Aggregate feasibility is

$$C_t + X_t \quad = \quad e^{Z_t} K_t^{\alpha},$$

where X_t is investment in capital and K_t is current level of capital in the economy. Capital accumulates as

$$K_{t+1} = (1-\delta)K_t + \left(1 - S\left(\frac{X_t}{X_{t-1}}\right)\right) X_t,$$

where $S()$ is an investment adjustment cost function specified as

$$S\left(\frac{X_t}{X_{t-1}}\right) = \frac{\kappa}{2}\left(\frac{X_t}{X_{t-1}} - e^{\frac{\phi}{1-\alpha}}\right)^2.$$

Finally, technology, Z_t evolves as a random walk with drift

$$Z_t = \phi + Z_{t-1} + \tau\epsilon_t.$$

These preferences can be interpreted as a special case of Epstein and Zin (1989) preferences where the intertemporal elasticity of substitution is set equal to 1. As demonstrated by Koijen, Fernández-Villaverde, Rubio-Ramírez, and van Binsbergen (2008), it is impossible to estimate γ through linearization or log-linearization. Terms involving γ show up in the second order solution as part of the constant risk-adjustment, as well as in the third and higher orders as part of the time varying adjustment for risk. We explore two choices of surrogate marginal likelihoods. The first surrogate marginal likelihood is constructed using a Square Root Unscented Kalman filter (van der Merwe and Wan (2001)), keeping fixed the nonlinear solution. A second surrogate is constructed from linearizing the equilibrium conditions and using a Kalman filter to compute the marginal likelihood.[19]

The planner chooses an allocation (C_t, X_t, K_{t+1}) to maximize utility subject to aggregate feasibility and capital accumulation. Let V_t denote the optimal value of the problem. The solution of the planners problem yields the first order conditions.

$$\begin{aligned}
0 &= -\mu_t + \beta\frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]}\left(\alpha\lambda_{t+1}e^{Z_{t+1}}K_{t+1}^{\alpha-1} + \mu_{t+1}(1-\delta)\right) \\
0 &= -\lambda_t + \beta\frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]}\left(\mu_{t+1}S'\left(\frac{X_{t+1}}{X_t}\right)\left(\frac{X_{t+1}}{X_t}\right)^2\right) \\
&\quad +\mu_t\left(S\left(\frac{X_t}{X_{t-1}}\right) + S'\left(\frac{X_t}{X_{t-1}}\right)\left(\frac{X_t}{X_{t-1}}\right)\right) \\
0 &= -\lambda_t + \frac{(1-\beta)}{C_t}
\end{aligned}$$

Where λ_t, and μ_t are Lagrange multipliers on the resource and capital accumulation constraints respectively. If we let $q_t = \frac{\mu_t}{\lambda_t}$, and some algebra yields:

[19] In this second surrogate, the parameter γ is unidentified in the likelihood.

$$
\begin{aligned}
0 &= -q_t + \beta \frac{C_t}{C_{t+1}} \frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]} \left(\alpha e^{Z_{t+1}} K_{t+1}^{\alpha-1} + q_{t+1}(1-\delta)\right) \\
0 &= -1 + \beta \frac{C_t}{C_{t+1}} \frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]} \left(q_{t+1} S'\left(\frac{X_{t+1}}{X_t}\right)\left(\frac{X_{t+1}}{X_t}\right)^2\right) \\
&\quad + q_t \left(S\left(\frac{X_t}{X_{t-1}}\right) + S'\left(\frac{X_t}{X_{t-1}}\right)\left(\frac{X_t}{X_{t-1}}\right)\right)
\end{aligned}
$$

Now define the return on capital, R_{t+1}^k as:

$$
R_{t+1}^k = \frac{\left(\alpha e^{Z_{t+1}} K_{t+1}^{\alpha-1} + q_{t+1}(1-\delta)\right)}{q_t}
$$

And a risk free return R_t^f, which satisfies:

$$
1 = R_t^f E_t \left[\beta \frac{C_t}{C_{t+1}} \frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]}\right]
$$

Combining with our definitions of the value function, resource constraint, capital evolution, and first order conditions, we get the following set of equilibrium conditions.

$$
\begin{aligned}
1 &= E_t \left[\beta \frac{C_t}{C_{t+1}} \frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]} R_t^f\right] \\
1 &= E_t \left[\beta \frac{C_t}{C_{t+1}} \frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]} R_{t+1}^k\right] \\
1 &= E_t \left[\beta \frac{C_t}{C_{t+1}} \frac{e^{(1-\gamma)V_{t+1}}}{E_t[e^{(1-\gamma)V_{t+1}}]} \left(q_{t+1} S'\left(\frac{X_{t+1}}{X_t}\right)\left(\frac{X_{t+1}}{X_t}\right)^2\right)\right] \\
&\quad + q_t \left(1 - S\left(\frac{X_t}{X_{t-1}}\right) - S'\left(\frac{X_t}{X_{t-1}}\right)\left(\frac{X_t}{X_{t-1}}\right)\right) \\
R_{t+1}^k &= \frac{\left(\alpha e^{Z_{t+1}} K_{t+1}^{\alpha-1} + q_{t+1}(1-\delta)\right)}{q_t} \\
V_t &= (1-\beta) log(C_t) + \frac{\beta}{1-\gamma} log\left(E_t[e^{(1-\gamma)V_{t+1}}]\right) \\
C_t &= K_t^\alpha e^{Z_t} - X_t \\
K_{t+1} &= (1-\delta) K_t + \left(1 - S\left(\frac{X_t}{X_{t-1}}\right)\right) X_t \\
Z_t &= \phi + Z_{t-1} + \tau \epsilon_t
\end{aligned}
$$

Since technology is growing over time, we need to redefine some of our variables. Define:

$$\begin{aligned}
\tilde{X}_t &= \frac{X_t}{e^{\frac{Z_t}{1-\alpha}}} \\
\tilde{C}_t &= \frac{C_t}{e^{\frac{Z_t}{1-\alpha}}} \\
\tilde{K}_{t+1} &= \frac{K_{t+1}}{e^{\frac{Z_t}{1-\alpha}}} \\
\tilde{V}_{t+1} &= V_{t+1} - \frac{z_t}{1-\alpha}
\end{aligned}$$

Then the equilibrium conditions become:

$$\begin{aligned}
1 &= E_t\left[\beta \frac{\tilde{C}_t}{\tilde{C}_{t+1} e^{\frac{1}{1-\alpha}\Delta Z_{t+1}}} \frac{e^{(1-\gamma)\tilde{V}_{t+1}}}{E_t[e^{(1-\gamma)\tilde{V}_{t+1}}]} R_t^f\right] \\
1 &= E_t\left[\beta \frac{\tilde{C}_t}{\tilde{C}_{t+1} e^{\frac{1}{1-\alpha}\Delta Z_{t+1}}} \frac{e^{(1-\gamma)\tilde{V}_{t+1}}}{E_t[e^{(1-\gamma)\tilde{V}_{t+1}}]} R_{t+1}^k\right] \\
1 &= E_t[\beta \frac{\tilde{C}_t}{\tilde{C}_{t+1} e^{\frac{1}{1-\alpha}\Delta Z_{t+1}}} \frac{e^{(1-\gamma)\tilde{V}_{t+1}}}{E_t[e^{(1-\gamma)\tilde{V}_{t+1}}]} \\
&\quad \left(q_{t+1} S'\left(\frac{X_{t+1}}{X_t} e^{\frac{1}{1-\alpha}\Delta Z_{t+1}}\right)\left(\frac{X_{t+1}}{X_t} e^{\frac{1}{1-\alpha}\Delta Z_{t+1}}\right)^2\right)] \\
&\quad + q_t\left(1 - S\left(\frac{X_t}{X_{t-1}} e^{\frac{1}{1-\alpha}\Delta Z_t}\right) - S'\left(\frac{X_t}{X_{t-1}}\right)\left(\frac{X_t}{X_{t-1}} e^{\frac{1}{1-\alpha}\Delta Z_t}\right)\right) \\
R_t^k &= \frac{\left(\alpha e^{\Delta Z_{t+1}} \tilde{K}_t^{\alpha-1} + q_t(1-\delta)\right)}{q_{t-1}} \\
\tilde{V}_t &= (1-\beta) log(\tilde{C}_t) + \frac{1}{1-\alpha}\Delta Z_t + \frac{\beta}{1-\gamma} log\left(E_t[e^{(1-\gamma)\tilde{V}_{t+1}}]\right) \\
\tilde{C}_t &= \tilde{K}_t^{\alpha} e^{-\frac{\alpha}{1-\alpha}\Delta Z_t} - \tilde{X}_t \\
\tilde{K}_{t+1} &= (1-\delta)\tilde{K}_t e^{-\frac{\alpha}{1-\alpha}\Delta Z_t} + \left(1 - S\left(\frac{\tilde{X}_t}{\tilde{X}_{t-1}} e^{\frac{1}{1-\alpha}\Delta Z_t}\right)\right)\tilde{X}_t \\
\Delta Z_t &= \phi + \tau \epsilon_t
\end{aligned}$$

Let χ be the perturbation parameter, and let $v\hat{a}r$ denote the deviations of var_t from its deterministic steady state, $v\hat{a}r_t = var_t - var_{ss}$. Here it is assumed that $\chi = 1$, and $\chi_{ss} = 0$. We will let lowercase letters denote log of the variables, ie $c_t = log(\tilde{C}_t)$. The state is the vector $\hat{s}_t = (\hat{k}_t, \hat{x}_{t-1}, \hat{q}_{t-1}, \hat{\epsilon}_t, \hat{\chi})$,

The 3rd order approximation of variable $var \in \{V_t, c_t, x_t, k_{t+1}, r_t^k, r_t^f\}$ are as follows,

$$var(\hat{s}_t) \approx var_{ss} + \sum_i var_{i,ss}\hat{s}_t^i + \sum_i \sum_j \frac{1}{2} var_{ij,ss}\hat{s}_t^i s_t^j + \sum_i \sum_j \sum_l \frac{1}{6} var_{ijl,ss}\hat{s}_t^i \hat{s}_t^j \hat{s}_t^l.$$

The log-linearized version only include terms up to the first order, that is

$$var(\hat{s}_t) \approx var_{ss} + \sum_i var_{i,ss}\hat{s}_t^i.$$

We use as the observable series log investment growth, log consumption growth, log return on capital and log of the risk free rate, $Y_t = (\Delta log(X_t), \Delta log(C_t), log(R_t^k), log(R_t^f))$ and so we link them to our model by the following,

$$\begin{aligned}
\Delta log(X_t) &= x_t - x_{t-1} + \tfrac{1}{1-\alpha}(\phi + \tau\epsilon_t) + \sigma_1\nu_{1,t} \\
\Delta log(C_t) &= c_t - c_{t-1} + \tfrac{1}{1-\alpha}(\phi + \tau\epsilon_t) + \sigma_2\nu_{2,t} \\
log(R_t^k) &= log(R_{ss}^k) + r_t^k + \sigma_3\nu_{3,t} \\
log(R_t^f) &= log(R_{ss}^f) + r_t^f + \sigma_4\nu_{4,t}
\end{aligned}$$

where $\nu_t = (\nu_{1,t}, \nu_{2,t}, \nu_{3,t}, \nu_{4,t})'$ is a vector of normally distributed measurement error shocks, $\nu_t \sim N(0, I)$ and $\sigma_1, \ldots, \sigma_4$ are loadings on the measurement errors. That is, our measurement equation,

$$Y_t = f(S_t, \nu_t).$$

The unobservable states in our state space system, will now be denoted S_t instead of the more common X_t as written in the previous sections due to an unfortunate notation clash with investment. The unobservable state is $S_t = (\hat{s}_t, \hat{s}_{t-1})'$ with transition:

$$\begin{aligned}
\hat{k}_{t+1} &= \textstyle\sum_i k_{i,ss}\hat{s}_t^i + \sum_i \sum_j \frac{1}{2} k_{ij,ss}\hat{s}_t^i s_t^j + \sum_i \sum_j \sum_l \frac{1}{6} k_{ijl,ss}\hat{s}_t^i \hat{s}_t^j \hat{s}_t^l \\
\hat{x}_t &= \textstyle\sum_i x_{i,ss}\hat{s}_t^i + \sum_i \sum_j \frac{1}{2} x_{ij,ss}\hat{s}_t^i s_t^j + \sum_i \sum_j \sum_l \frac{1}{6} x_{ijl,ss}\hat{s}_t^i \hat{s}_t^j \hat{s}_t^l \\
\hat{q}_t &= \textstyle\sum_i q_{i,ss}\hat{s}_t^i + \sum_i \sum_j \frac{1}{2} q_{ij,ss}\hat{s}_t^i s_t^j + \sum_i \sum_j \sum_l \frac{1}{6} q_{ijl,ss}\hat{s}_t^i \hat{s}_t^j \hat{s}_t^l \\
\hat{\epsilon}_{t+1} &= \hat{\epsilon}_{t+1} \\
1 &= 1 \\
\hat{s}_t &= \hat{s}_t
\end{aligned}$$

or,

$$S_{t+1} = g(S_t, \epsilon_{t+1}).$$

We generate data of length 200 using the following parameter values.

[INSERT TABLE 7 ABOUT HERE]

We do not estimate the measurement error loadings and instead fix each $\sigma_i = 0.001$.[20] The prior distribution for the vector of structural parameters as follows:

[INSERT TABLE 8 ABOUT HERE]

The priors and parameters were chosen to be representative of what one might estimate from the data using more complicated model but not necessarily the parameters one would get if such a simple model was brought to the data. The particle filter was evaluated with 5000 particles and the Metropolis-Hastings was run for 100000 draws. The surrogate transitions algorithm was also run identical starting values, number of particles, and number of draws. Like the previous example, the covariance matrix for the random walk part of the chain was estimated as the covariance matrix of the resulting draws from a previous run of the RWMH algorithm. As before, we demonstrate the algorithm varying the temperature from 1 to 4, and number of surrogate steps from 1 to 6.

5.2.1 Nonlinear Transitions

Here we examine the results for a model that is solved using a third order approximation with a marginal likelihood evaluated using a (square root) Unscented Kalman filter as our surrogate.

[INSERT TABLE 9 ABOUT HERE]

The output for the RWMH algorithm is reported in table 9. The Random Walk chain was fairly well tuned, achieving an acceptance rate of about 40%, although perhaps a bit too high. The autocorrelations range from $16 - 26$ meaning that when one is estimating the mean of the posterior distribution, the variance of the resulting estimate is $16-26$ times larger than if one had independent draws. The particle filter took approximately 0.4 seconds per evaluation.

[20] We also do not estimate the discount factor β.

[INSERT TABLE 10 ABOUT HERE]

The acceptance rates for the surrogate transitions chain are presented in table 10. They range from 0.33 all the way up to 0.87. While both the Surrogate Transitions and RWMH chains have higher than the reference acceptance rates, it is unclear what effects changing the proposal variance in the random walk chain will have as this might improve the performance of algorithms.

[INSERT TABLE 11 ABOUT HERE]

As we can see from the running times, by avoiding running the particle filter on rejected parameter values in the surrogate phase, we are able to decrease running times relative to the RWMH algorithm. This is especially true when we perform just one surrogate proposal step for any temperature. As with the stochastic volatility model in the previous section, for high temperature/intermediate step combinations, the surrogate transitions algorithm actually takes longer than the RWMH algorithm. And as before, this is because we are very likely to accept a draw in the surrogate phase and thus run the resulting particle filter at a new draw. Whether or not these produce a speedup with then depend greatly on the decrease in integrated autocorrelation time produced from our Markov chain.

[INSERT TABLE 12 ABOUT HERE]

[INSERT TABLE 13 ABOUT HERE]

In terms of autocorrelations, the surrogate transitions sampler yields a wider range of estimates of the integrated autocorrelation time, depending on the (T, K) combination. For a low number of intermediate steps, the sampler produces a higher integrated autocorrelation time for the parameters than the Metropolis algorithm. Fortunately, the increase in autocorrelation time is offset by the substantial decrease in running time of the algorithm. For a fixed T, as we increase K, the number of surrogate steps, the autocorrelation time decreases substantially.

[INSERT TABLE 14 ABOUT HERE]

If we take both running times and integrated autocorrelation times into account, we can examine the overall speedup. The speedup is positive for most (T, K) combinations. For T low and more than 1 intermediate step, the sampler sees a speedup of $1.4 - 6$ over the simple RWMH chain. Thus, our new sampler is anywhere from 40 to 500% more efficient than the naive implementation of the Random Walk Metropolis-Hastings chain, depending on the parameter and (T,K) combination used. Only one estimate showed slowdown, estimating γ with low K. Averaged over parameters and (T, K) combinations, the speedup is approximately 2.24, and, focusing on the subblock with temperatures less than or equal to 2, and more than 1 interemediate step, speedup was on average 2.85.

5.2.2 Nonlinear Transitions: Adjusting the Surrogate Proposal Density

We notice that the speedup computed in the previous section was primarily due to decreases in integrated autocorrelation time when the number of intermediate steps is large. This section addresses whether there is room to scale the surrogate proposal density in such a way to trade higher autocorrelation time for less computational time. For this experiment, we scale the standard deviation of the surrogate proposal density of our parameters by the square root of the number of surrogate steps, $q(\theta, \cdot) = N(\theta, K\Sigma')$ for a surrogate sampler with K steps. Larger proposal variance should decrease the chance that we accept the move, but, when we do move, move us further in the state space. By decreasing the number of acceptances in the surrogate step, we should run our filter less often. When we do try to evaluate the particle filter, hopefully we will be further from our previous parameter configuration, maintaining a favorable integrated autocorrelation time.

[INSERT TABLE 15 ABOUT HERE]

Acceptance rates are lower than in the unscaled case, and running times in table 16 are all less than in the unscaled case. Thus, we are accepting less points in the surrogate phase and need to run the particle filter less often.

[INSERT TABLE 16 ABOUT HERE]

[INSERT TABLE 17 ABOUT HERE]

Integrated autocorrelation times are reported in table 17. For $K = 1$ they are the same as before. For larger $K's$ are slightly larger than in the unscaled case, but not by much.

[INSERT TABLE 19 ABOUT HERE]

In terms of overall speedup, we see in table 19 a dramatic improvement relative to the unscaled case. The mean speedup over all parameters and (T, K) combinations is now 2.97, while the mean speedup over parameters and the $T \leq 2$, $K \geq 2$ block of runs is 4.47. This shows that while varying the temperature and number of surrogate steps can and will affect the performance of the algorithm, large gains in efficiency can be achieved by taking care in designing the surrogate Markov chain.

5.2.3 Linear Transitions

A more extreme implementation of the surrogate transition method is to uses a linearized model as a surrogate guide. This linear model takes almost no additional coding given one is solving for a higher order approximation. Further, linearized models are still the primary tool for analysis in DSGE models and are able to fit quantities quite well. Unfortunately, linearization, as pointed out by Fernández-Villaverde and Rubio-Ramírez (2005) has its drawbacks. On major drawback is that the parameter γ is not identified when taking a first order approximation to the policy functions. Thus the surrogate phase merely samples from the prior on γ which is always valid so long as the prior is not improper. We also demean the data before evaluating the Kalman filter in the surrogate step. The results are contained in tables 20 - 24.

[INSERT TABLE 20 ABOUT HERE]

[INSERT TABLE 21 ABOUT HERE]

[INSERT TABLE 22 ABOUT HERE]

[INSERT TABLE 23 ABOUT HERE]

[INSERT TABLE 24 ABOUT HERE]

Relative to the nonlinear filter, the surrogate method with a linearized model shows much lower acceptance rates and similar running times for a wide array of (T, K) combinations.[21] Again, integrated autocorrelation times drop as the number of intermediate steps rise, but not as dramatically as with using the nonlinear surrogate likelihood. The integrated autocorrelation times for our linearized surrogate model are higher than those for the nonlinear one. In terms of overall speedup relative to the RWMH we see that it is general positive, except for very high (T, K) combinations and when $K = 1$. The algorithm shows slowdown when estimating γ, when $K = 1$ for any T, although this performance could have been expected given that it is unidentified in the surrogate likelihood. However, increases in efficiency are seen in the parameters that are shared across models. Even taking into account this slowdown, averaging speedup over the parameters remains positive. The average over all parameters and (T, K) combinations showed a speedup of 1.62. For lower T, and K greater than 2, it ranges from $1.11 - 2.86$. It is less efficient than using the nonlinear filter as a surrogate, suggesting that nonlinearities are important and differences exist between the posterior distribution induced by linear and nonlinear solution methods. As a word of caution, the results for unidentified parameters vary substantially if the true likelihood is far away from a tight prior, which suggests the use of fairly diffuse priors for such parameters.

6 Conclusion

This paper has proposed a novel combination of ideas for estimating parameters in a nonlinear state space system. By combining various approximation methods, we are able to substantially speedup the rate at which our Markov chain produces draws from its invariant distribution. This method is especially suited for the estimation of nonlinear DSGE models where different solution methods can be combined to form surrogate models. As we have explored the sampler for only the simplest of DSGE models, more complicated models with large state spaces and/or a lot of particles should see

[21] We compute the second stage acceptance rate for $K = 1$ to get an idea of how close the surrogate is to the likelihood of interest. For the nonlinear surrogate model, it ranged from approximately 0.9 when $T = 1$ to 0.55 when $T = 4$. For the linear surrogate model, the second stage acceptance rates ranged from $0.5 - 0.65$. This suggests that the nonlinear surrogate is very accurate for low T and we lose little from using multiple accept/reject steps with this approximation.

an even greater advantage. Another way of viewing our algorithm is that running the particle filter can be seen as a way of correcting the invariant distribution of a Markov chain that is computed by an approximate marginal likelihood function.

Although we generally find good performance with only a few intermediate steps, tuning the surrogate sampler and temperature level is an open question. What optimal acceptance rate one should strive for, in both the surrogate and overall algorithm is unknown and very likely problem dependent.

Exploiting the linear model allows an increase in efficiency over the baseline, especially for the parameters which are well identified in the linear model. Nonlinear approximations to the marginal likelihood increase the efficiency gains. This is the result of nonlinearities being important. However, as we show, we can still use the linear parts of the model in such a way to aid in characterizing these effects. One can certainly come up with examples in which using a linear state space model or an Unscented Kalman filter to evaluate the likelihood gives disastrous results, and one should design a surrogate model with care.

References

Andreasen, M. M. (2010): "Non-linear DSGE Models and The Optimized Particle Filter," Unpublished paper.

Andrieu, C., A. Doucet, and R. Holenstein (2010): "Particle Markov Chain Monte Carlo with discussion," *Journal of Royal Statistical Society Series B*, 72, 269 – 342.

Andrieu, C., and G. O. Roberts (2009): "The Pseudo-Marginal Approach for Efficient Computation," *Annals of Statistics*, 37, 697–725.

Barillas, F., L. P. Hansen, and T. J. Sargent (2009): "Doubts or Variability," *Journal of Economic Theory*, 144, 2388–2418.

Beaumont, M. A. (2003): "Estimation of Population Growth or Decline in Genetically Monitored Populations," *Genetics*, 164, 1139–1160.

Bidder, R., and M. Smith (2010a): "Doubts and Variability," Working paper, New York University.

——— (2010b): "Robust Control in a Nonlinear DSGE Model," Working paper, New York University.

Bliznyuk, N., D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan (2008): "Bayesian Calibration of Expensive Models Using Optimization and Radial Basis Function Approximation," *Journal of Computational and Graphical Statistics*, pp. 270–294.

Christen, J., and C. Fox (2005): "MCMC using an Approximation," *Journal of Computer and Graphical Statistics*, 14, 795–810.

Christiano, L. J., M. Eichenbaum, and C. L. Evans (2005): "Nominal Rigidities and the Dynamic Eect of a Shock to Monetary Policy," *The Journal of Political Economy*, 113, 1–45.

Cui, T., C. Fox, and M. O'Sullivan (2011): "Adaptive Error Modelling in MCMC Sampling for Large Scale Inverse Problems," Report, University of Auckland, Faculty of Engineering, no 687.

Del Negro, M., F. Schorfheide, F. Smets, and R. Wouters (2007): "On the Fit and Forecasting Performance of New Keynesian Models," *Journal of Business and Economic Statistics*, 25, 123–162.

Doucet, A., S. Godsill, and C. Andrieu (2000): "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, 10, 197208.

Efendiev, Y., T. Hou, and W. Luo (2006): "Preconditioning of MCMC simulations using coarse-scale models," *SIAM Journal on Scientific Computing*, 28, 776–803.

Epstein, L., and S. Zin (1989): "Substitution Risk Aversion and the Temporal Behavior of Consumption and Asset Returns: A Theoretical Framework," *Econometrica*, 57, 937–968.

Fernández-Villaverde, J., P. A. Guerrón-Quintana, J. F. Rubio-Ramírez, and M. Uribe (2009): "Risk Matters: The Real Effects of Volatility Shocks," Unpublished paper, University of Pennsylvania.

Fernández-Villaverde, J., and J. F. Rubio-Ramírez (2005): "Estimating Dynamic Equilibrium Economies: Linear Versus Nonlinear Likelihood," *Journal of Applied Econometrics*, 20, 891–910.

——— (2007): "Estimating Macroeconomic Models: A Likelihood Approach," *Review of Economic Studies*, 74, 1059–1087.

Fielding, M., D. J. Nott, and S.-Y. Liong (2011): "Efficient MCMC Schemes for Computationally Expensive Posterior Distributions," *Technometrics*, pp. 16–28.

Flury, T., and N. Shephard (2008): "Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models," Unpublished paper, University of Oxford.

Frangos, M., Y. Marzouk, K. Willcox, and B. van loeman Waanders (2010): "Surrogate and Reduced Order Modeling: A Comparison of Approaches for Large Scale Statistical Inverse

Problems," in *Large Scale Inverse Problems and Quantification of Uncertainty*, ed. by L. Biegler, G. Biros, O.Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, Y. Marzouk, L. Tenorio, B. van loeman Waanders, , and K. Willcox. John Wiley Sons, Chichester, UK.

Gordon, N. J., D. J. Salmond, and A. F. M. Smith (1993): "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F on Radar and Signal Processing*, 140, 107–113.

Hansen, L. P., and T. J. Sargent (2007): "Robust Estimation and Control Without Commitment," *Journal of Economic Theory*, 136, 1–27.

Hastings, W. K. (1970): "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, 97–109.

Justiniano, A., and G. Primiceri (2008): "The Time Varying Volatility of Macroeconomic Fluctuations," *American Economic Review*, 98, 604–641.

Koijen, R., J. Fernández-Villaverde, J. F. Rubio-Ramírez, and J. van Binsbergen (2008): "Computation and Likelihood Estimation of DSGE Models with Recursive Preferences," Unpublished paper, University of Pennsylvania.

Liu, J. (2001): *Monte Carlo Strategies in Scientific Computing*. Springer, New York, NY.

Liu, J. S., and R. Chen (1998): "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, 93, 1032–1044.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953): "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087 – 1092.

Meyer, R., D. A. Fournier, and A. Berg (2003): "Stochastic volatility: Bayesian computation using automatic differentiation and the extended Kalman filter," *The Econometrics Journal*, 6, 408–420.

Rasmussen, C. E. (2003): "Gaussian Process to Speed Up Hybrid Monte Carlo for Expensive Bayesian Integrals," *Bayesian Statistics*, pp. 651–659.

Robert, C. P., and G. Casella (2004): *Monte Carlo statistical methods*. Springer, New York, NY, 2 edn.

Roberts, G., and J. Rosenthal (2001): "Optimal scaling for various Metropolis-Hastings algorithms," *Statistical Science*, 16, 351–367.

Roberts, G., and J. Rosenthal (2004): "General State Space Markov Chains and MCMC Algorithms," *Probability Surveys*, pp. 20–71.

Sargent, T. J. (1989): "Two Models of Measurements and the Investment Accelerator," *Journal of Political Economy*, 97, 251–287.

Silva, R., P. Giordani, R. Kohn, and M. Pitt (2009): "Particle filtering within adaptive Metropolis Hastings sampling," Working Paper.

Skaug, H., and J. Yu (2008): "Automated Likelihood Based Inference for Stochastic Volatility Models," Working Paper, Singapore Management University.

Smets, F., and R. Wouters (2003): "An Estimated Stochastic Dynamic General Equilibrium Model of the Euro Area," *Journal of the European Economic Association*, 1, 1123–1175.

Tallarini, T. D. (2000): "Risk-Sensitive Real Business Cycle Models," *Journal of Monetary Economics*, 45, 507–532.

Tierney, L. (1994): "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, pp. 1701–1728.

van der Merwe, R., N. de Freitas, A. Doucet, and E. Wan (2000): "The Unscented Particle Filter," Discussion Paper CUED/F-INFENG/TR 380, Cambridge University Engineering Department, Cambridge, England.

van der Merwe, R., and E. A. Wan (2001): "The Square-Root Unscented Kalman Filter for State and Parameter-Estimation," in *International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah.

Table 1: SV Model: Random Walk Metropolis-Hastings

		ρ	τ	μ
AR	0.29			
Time	23874.50			
IAT		12.60	10.80	11.58
s/Eff		3.01	2.58	2.76

Table 2: SV Model: Surrogate Transitions Acceptance Rates

T\K	1	2	3	4	5	6
1	0.20	0.31	0.36	0.39	0.40	0.40
2	0.25	0.35	0.39	0.38	0.37	0.36
3	0.26	0.34	0.35	0.33	0.31	0.29
4	0.27	0.32	0.32	0.29	0.27	0.24

Table 3: SV Model: Surrogate Transitions Running time

T\K	1	2	3	4	5	6
1	9229.31	14832.38	18371.88	20555.92	21892.49	22747.26
2	12010.63	18048.08	21165.83	22638.38	23438.59	23814.77
3	14046.27	19985.97	22522.86	23621.78	24222.10	24227.45
4	15462.99	21094.90	23197.13	23895.08	24284.64	24371.16

Table 4: SV Model: Surrogate Transitions Integrated Autocorrelation Time

			ρ			
T\K	1	2	3	4	5	6
1	23.32	11.04	12.56	9.71	8.91	8.76
2	12.27	9.36	7.30	7.75	7.98	6.95
3	18.11	10.16	9.71	7.09	6.28	6.24
4	13.52	9.19	7.24	11.37	8.69	8.77
			τ			
T\K	1	2	3	4	5	6
1	24.61	14.47	11.96	9.35	7.24	8.46
2	14.04	7.64	7.58	6.11	8.84	7.02
3	16.59	7.56	7.86	7.63	5.51	5.25
4	13.28	7.86	7.98	7.61	7.65	8.06
			μ			
T\K	1	2	3	4	5	6
1	12.48	11.62	9.03	5.09	6.78	5.74
2	13.19	7.02	6.59	3.94	6.57	4.20
3	16.00	9.06	6.80	5.40	7.05	5.20
4	10.74	6.17	6.30	5.11	7.04	6.56

Table 5: SV Model: Surrogate Transitions Time per Effectively Independent Sample

ρ						
T\K	1	2	3	4	5	6
1	2.15	1.64	2.31	2.00	1.95	1.99
2	1.47	1.69	1.55	1.76	1.87	1.66
3	2.54	2.03	2.19	1.68	1.52	1.51
4	2.09	1.94	1.68	2.72	2.11	2.14
τ						
T\K	1	2	3	4	5	6
1	2.27	2.15	2.20	1.92	1.58	1.92
2	1.69	1.38	1.61	1.38	2.07	1.67
3	2.33	1.51	1.77	1.80	1.33	1.27
4	2.05	1.66	1.85	1.82	1.86	1.97
μ						
T\K	1	2	3	4	5	6
1	1.15	1.72	1.66	1.05	1.49	1.31
2	1.58	1.27	1.39	0.89	1.54	1.00
3	2.25	1.81	1.53	1.28	1.71	1.26
4	1.66	1.30	1.46	1.22	1.71	1.60

Table 6: SV Model: Speedup Relative to Random Walk Metropolis-Hastings

			ρ			
T\K	1	2	3	4	5	6
1	1.40	1.84	1.30	1.51	1.54	1.51
2	2.04	1.78	1.95	1.71	1.61	1.82
3	1.18	1.48	1.38	1.80	1.98	1.99
4	1.44	1.55	1.79	1.11	1.43	1.41
			τ			
T\K	1	2	3	4	5	6
1	1.13	1.20	1.17	1.34	1.63	1.34
2	1.53	1.87	1.61	1.86	1.24	1.54
3	1.11	1.71	1.46	1.43	1.93	2.03
4	1.26	1.55	1.39	1.42	1.39	1.31
			μ			
T\K	1	2	3	4	5	6
1	2.40	1.60	1.67	2.64	1.86	2.12
2	1.74	2.18	1.98	3.10	1.80	2.76
3	1.23	1.53	1.80	2.17	1.62	2.20
4	1.66	2.12	1.89	2.26	1.62	1.73

Table 7: RBC Model: Parameters

γ	50
α	0.4
δ	0.03
κ	4
ϕ	0.005
$\log(\tau)$	-5.3

Table 8: RBC Model: Priors

γ	Normal(50, 10^2)
α	Normal(0.3, 0.05^2)
δ	Normal(0.025, 0.005^2)
κ	Normal(4, 1.5^2)
ϕ	Normal(0.3, 0.05^2)
$\log(\tau)$	Normal(-5.3, 2^2)

Table 9: RBC Model: Random Walk Metropolis-Hastings

		γ	α	δ	κ	ϕ	τ
AR	0.40						
Time	43452.28						
IAT		16.31	18.51	21.04	25.94	21.72	20.59
s/Eff		7.09	8.04	9.14	11.27	9.44	8.95

Table 10: RBC Model: Nonlinear Surrogate Transitions Acceptance Rates

T\K	1	2	3	4	5	6
1	0.37	0.59	0.72	0.80	0.85	0.87
2	0.40	0.55	0.61	0.62	0.61	0.59
3	0.40	0.50	0.50	0.48	0.45	0.42
4	0.40	0.47	0.44	0.40	0.36	0.33

Table 11: RBC Model: Nonlinear Surrogate Transitions Running time

T\K	1	2	3	4	5	6
1	19692.80	32292.76	39792.75	45780.85	49597.33	50241.67
2	25964.66	38934.23	45257.38	51278.58	51411.03	53320.81
3	29564.54	40926.98	46061.55	48388.76	50029.13	51387.92
4	32613.70	42968.98	46838.05	48895.27	50325.81	51418.05

Table 12: RBC Model: Nonlinear Surrogate Transitions Integrated Autocorrelation Time

γ						
T\K	1	2	3	4	5	6
1	36.07	9.68	8.69	5.37	5.11	4.04
2	27.76	11.96	9.25	7.83	8.36	5.71
3	24.06	13.01	13.73	9.52	9.19	5.27
4	18.93	14.31	11.58	9.66	10.43	12.76
α						
T\K	1	2	3	4	5	6
1	18.71	14.25	6.10	8.26	4.13	3.96
2	20.39	14.39	7.12	5.22	6.63	5.80
3	22.78	11.59	8.35	7.20	8.25	5.90
4	18.21	14.26	10.29	12.41	7.66	8.90
δ						
T\K	1	2	3	4	5	6
1	25.28	16.60	11.94	6.36	6.01	5.28
2	20.19	14.71	8.29	6.54	6.03	3.85
3	19.97	14.01	7.36	7.66	6.77	5.48
4	25.68	15.92	10.77	9.28	10.40	8.83
κ						
T\K	1	2	3	4	5	6
1	20.55	11.53	8.28	9.26	4.44	3.73
2	17.73	17.57	12.09	7.02	5.82	4.86
3	30.27	13.04	11.00	11.41	7.48	6.81
4	20.54	19.56	13.10	10.12	9.33	8.35
ϕ						
T\K	1	2	3	4	5	6
1	30.85	11.57	7.67	6.65	4.84	4.80
2	30.73	11.55	8.57	7.00	6.96	4.63
3	22.68	14.78	11.97	8.34	9.16	6.83
4	16.92	14.50	12.45	8.92	9.75	11.98
τ						
T\K	1	2	3	4	5	6
1	34.70	9.83	8.13	5.68	5.92	4.00
2	22.15	10.30	8.30	9.14	8.62	4.43
3	18.70	14.00	11.53	8.78	7.10	7.05
4	20.26	16.25	14.48	12.08	9.13	9.97

Table 13: RBC Model: Nonlinear Surrogate Transitions Time per Effectively Independent Sample

γ						
T\K	1	2	3	4	5	6
1	7.10	3.13	3.46	2.46	2.54	2.03
2	7.21	4.66	4.18	4.02	4.30	3.04
3	7.11	5.33	6.32	4.61	4.60	2.71
4	6.17	6.15	5.42	4.72	5.25	6.56
α						
T\K	1	2	3	4	5	6
1	3.68	4.60	2.43	3.78	2.05	1.99
2	5.29	5.60	3.22	2.68	3.41	3.09
3	6.74	4.74	3.85	3.48	4.13	3.03
4	5.94	6.13	4.82	6.07	3.86	4.58
δ						
T\K	1	2	3	4	5	6
1	4.98	5.36	4.75	2.91	2.98	2.65
2	5.24	5.73	3.75	3.35	3.10	2.05
3	5.90	5.73	3.39	3.71	3.39	2.81
4	8.37	6.84	5.04	4.54	5.24	4.54
κ						
T\K	1	2	3	4	5	6
1	4.05	3.72	3.29	4.24	2.20	1.87
2	4.60	6.84	5.47	3.60	2.99	2.59
3	8.95	5.34	5.07	5.52	3.74	3.50
4	6.70	8.41	6.13	4.95	4.70	4.29
ϕ						
T\K	1	2	3	4	5	6
1	6.08	3.74	3.05	3.04	2.40	2.41
2	7.98	4.50	3.88	3.59	3.58	2.47
3	6.71	6.05	5.51	4.04	4.58	3.51
4	5.52	6.23	5.83	4.36	4.91	6.16
τ						
T\K	1	2	3	4	5	6
1	6.83	3.18	3.24	2.60	2.94	2.01
2	5.75	4.01	3.76	4.69	4.43	2.36
3	5.53	5.73	5.31	4.25	3.55	3.62
4	6.61	6.98	6.78	5.91	4.60	5.13

Table 14: RBC Model: Nonlinear Surrogate Speedup Relative to Random Walk Metropolis-Hastings

			γ			
T\K	1	2	3	4	5	6
1	1.00	2.27	2.05	2.88	2.80	3.49
2	0.98	1.52	1.69	1.76	1.65	2.33
3	1.00	1.33	1.12	1.54	1.54	2.62
4	1.15	1.15	1.31	1.50	1.35	1.08
			α			
T\K	1	2	3	4	5	6
1	2.18	1.75	3.31	2.13	3.93	4.05
2	1.52	1.44	2.50	3.00	2.36	2.60
3	1.19	1.70	2.09	2.31	1.95	2.65
4	1.35	1.31	1.67	1.33	2.09	1.76
			δ			
T\K	1	2	3	4	5	6
1	1.84	1.71	1.92	3.14	3.07	3.45
2	1.74	1.60	2.44	2.73	2.95	4.46
3	1.55	1.59	2.70	2.47	2.70	3.25
4	1.09	1.34	1.81	2.02	1.75	2.01
			κ			
T\K	1	2	3	4	5	6
1	2.79	3.03	3.42	2.66	5.12	6.01
2	2.45	1.65	2.06	3.13	3.77	4.35
3	1.26	2.11	2.23	2.04	3.01	3.22
4	1.68	1.34	1.84	2.28	2.40	2.63
			ϕ			
T\K	1	2	3	4	5	6
1	1.55	2.53	3.09	3.10	3.93	3.92
2	1.18	2.10	2.43	2.63	2.64	3.83
3	1.41	1.56	1.71	2.34	2.06	2.69
4	1.71	1.52	1.62	2.16	1.92	1.53
			τ			
T\K	1	2	3	4	5	6
1	1.31	2.82	2.77	3.44	3.05	4.45
2	1.56	2.23	2.38	1.91	2.02	3.79
3	1.62	1.56	1.68	2.11	2.52	2.47
4	1.35	1.28	1.32	1.52	1.95	1.74

Table 15: RBC Model: Nonlinear Surrogate Transitions Acceptance Rates, $\sqrt{K}scale$

T\K	1	2	3	4	5	6
1	0.37	0.40	0.39	0.37	0.34	0.32
2	0.40	0.39	0.36	0.34	0.32	0.29
3	0.40	0.36	0.32	0.29	0.26	0.24
4	0.40	0.33	0.27	0.24	0.22	0.19

Table 16: RBC Model: Nonlinear Surrogate Transitions Running time, $\sqrt{K}scale$

T\K	1	2	3	4	5	6
1	18883.73	21589.49	22051.98	21875.95	21725.40	21652.62
2	25468.33	29137.78	29874.37	30091.10	29832.27	29485.74
3	31119.73	35687.11	37385.15	39256.09	38219.48	38447.14
4	34327.98	40368.05	40925.25	42536.11	43748.29	42742.30

Table 17: RBC Model: Nonlinear Surrogate Transitions Integrated Autocorrelation Time, $\sqrt{K}scale$

			γ			
T\K	1	2	3	4	5	6
1	36.07	12.03	9.09	7.61	6.49	7.32
2	27.76	12.28	8.51	8.13	8.72	9.03
3	24.06	13.72	11.52	8.46	11.98	9.79
4	18.93	15.65	11.37	10.15	11.09	13.18
			α			
T\K	1	2	3	4	5	6
1	18.71	12.01	7.45	7.44	8.40	6.57
2	20.39	11.51	9.89	8.27	7.19	8.34
3	22.78	14.76	12.92	8.73	10.08	8.13
4	18.21	16.37	7.05	11.82	10.14	13.72
			δ			
T\K	1	2	3	4	5	6
1	25.28	8.23	9.45	6.01	5.91	8.31
2	20.19	11.22	6.93	8.21	8.08	6.32
3	19.97	17.40	7.27	8.80	9.06	12.06
4	25.68	12.79	11.36	11.69	10.70	15.55
			κ			
T\K	1	2	3	4	5	6
1	20.55	13.31	8.00	8.88	7.13	6.55
2	17.73	12.66	6.32	7.52	6.99	8.31
3	30.27	15.38	10.23	8.75	10.71	9.81
4	20.54	14.50	10.88	12.37	11.24	17.19
			ϕ			
T\K	1	2	3	4	5	6
1	30.85	14.30	9.10	6.40	7.81	5.71
2	30.73	13.65	10.86	7.31	7.66	9.25
3	22.68	16.72	12.76	8.54	12.06	11.06
4	16.92	16.40	9.86	9.02	11.79	10.08
			τ			
T\K	1	2	3	4	5	6
1	34.70	9.10	6.99	7.98	6.37	6.84
2	22.15	13.72	7.18	7.46	7.18	7.87
3	18.70	17.80	8.97	10.07	9.71	9.73
4	20.26	13.08	11.59	9.08	11.23	12.96

Table 18: RBC Model: Nonlinear Surrogate Transitions Time per Effectively Independent Sample, $\sqrt{K}scale$

			γ			
T\K	1	2	3	4	5	6
1	6.81	2.60	2.00	1.66	1.41	1.59
2	7.07	3.58	2.54	2.44	2.60	2.66
3	7.49	4.90	4.31	3.32	4.58	3.76
4	6.50	6.32	4.65	4.32	4.85	5.64
			α			
T\K	1	2	3	4	5	6
1	3.53	2.59	1.64	1.63	1.83	1.42
2	5.19	3.35	2.96	2.49	2.15	2.46
3	7.09	5.27	4.83	3.43	3.85	3.13
4	6.25	6.61	2.88	5.03	4.44	5.86
			δ			
T\K	1	2	3	4	5	6
1	4.77	1.78	2.08	1.31	1.28	1.80
2	5.14	3.27	2.07	2.47	2.41	1.86
3	6.21	6.21	2.72	3.45	3.46	4.64
4	8.81	5.16	4.65	4.97	4.68	6.65
			κ			
T\K	1	2	3	4	5	6
1	3.88	2.87	1.76	1.94	1.55	1.42
2	4.52	3.69	1.89	2.26	2.09	2.45
3	9.42	5.49	3.82	3.44	4.09	3.77
4	7.05	5.85	4.45	5.26	4.92	7.35
			ϕ			
T\K	1	2	3	4	5	6
1	5.83	3.09	2.01	1.40	1.70	1.24
2	7.83	3.98	3.24	2.20	2.28	2.73
3	7.06	5.97	4.77	3.35	4.61	4.25
4	5.81	6.62	4.03	3.84	5.16	4.31
			τ			
T\K	1	2	3	4	5	6
1	6.55	1.96	1.54	1.74	1.38	1.48
2	5.64	4.00	2.14	2.25	2.14	2.32
3	5.82	6.35	3.35	3.95	3.71	3.74
4	6.95	5.28	4.74	3.86	4.91	5.54

Table 19: RBC Model: Nonlinear Surrogate Speedup Relative to Random Walk Metropolis-Hastings, $\sqrt{K}scale$

			γ			
T\K	1	2	3	4	5	6
1	1.05	2.75	3.56	4.29	5.07	4.50
2	1.01	2.00	2.81	2.92	2.75	2.68
3	0.95	1.46	1.66	2.15	1.56	1.90
4	1.10	1.13	1.54	1.65	1.47	1.27
			α			
T\K	1	2	3	4	5	6
1	2.29	3.13	4.93	4.98	4.44	5.69
2	1.56	2.42	2.74	3.26	3.78	3.30
3	1.14	1.54	1.68	2.37	2.10	2.59
4	1.30	1.23	2.81	1.61	1.83	1.38
			δ			
T\K	1	2	3	4	5	6
1	1.93	5.19	4.42	7.01	7.18	5.12
2	1.79	2.82	4.45	3.73	3.82	4.95
3	1.48	1.48	3.39	2.67	2.66	1.99
4	1.05	1.78	1.98	1.85	1.97	1.39
			κ			
T\K	1	2	3	4	5	6
1	2.93	3.95	6.44	5.85	7.33	8.00
2	2.52	3.08	6.02	5.02	5.44	4.64
3	1.21	2.07	2.97	3.31	2.78	3.01
4	1.61	1.94	2.55	2.16	2.31	1.55
			ϕ			
T\K	1	2	3	4	5	6
1	1.63	3.08	4.74	6.80	5.61	7.70
2	1.22	2.39	2.93	4.32	4.16	3.49
3	1.35	1.59	1.99	2.84	2.06	2.24
4	1.64	1.44	2.36	2.48	1.84	2.21
			τ			
T\K	1	2	3	4	5	6
1	1.38	4.59	5.85	5.17	6.52	6.09
2	1.60	2.26	4.20	4.02	4.21	3.89
3	1.55	1.42	2.69	2.28	2.43	2.41
4	1.30	1.71	1.90	2.33	1.83	1.63

Table 20: RBC Model: Linear Surrogate Transitions Acceptance Rates

T\K	1	2	3	4	5	6
1	0.32	0.45	0.50	0.51	0.49	0.47
2	0.37	0.45	0.45	0.42	0.39	0.36
3	0.38	0.42	0.39	0.34	0.30	0.27
4	0.39	0.40	0.34	0.29	0.25	0.22

Table 21: RBC Model: Linear Surrogate Transitions Running time

T\K	1	2	3	4	5	6
1	23069.16	34762.59	41329.10	44819.00	46775.13	48249.95
2	29304.32	40313.62	44776.56	46852.80	48014.90	50022.72
3	32793.09	42812.45	46149.10	48090.15	48442.30	49216.49
4	35176.48	44087.98	46755.08	47726.13	48536.65	49232.81

Table 22: RBC Model: Linear Surrogate Transitions Integrated Autocorrelation Time

γ						
T\K	1	2	3	4	5	6
1	35.33	18.34	10.99	11.35	8.57	8.71
2	34.02	15.19	12.60	10.72	10.79	9.81
3	28.02	12.29	11.48	9.55	9.68	13.43
4	28.01	16.57	14.53	11.50	14.62	15.49
α						
T\K	1	2	3	4	5	6
1	25.04	13.68	11.73	9.25	11.29	6.95
2	33.19	12.91	11.60	8.92	8.31	9.12
3	27.69	14.56	11.46	9.66	12.73	11.38
4	20.70	12.76	13.01	10.07	16.81	12.40
δ						
T\K	1	2	3	4	5	6
1	27.29	13.91	14.26	9.94	9.68	8.24
2	17.65	14.48	11.05	8.18	9.84	7.41
3	22.79	21.20	12.05	8.65	8.10	11.30
4	17.97	15.23	12.65	13.71	12.06	15.87
κ						
T\K	1	2	3	4	5	6
1	34.41	15.85	10.30	9.46	13.01	10.91
2	23.20	11.09	9.82	9.22	9.36	8.63
3	22.71	12.90	11.75	12.25	10.55	12.34
4	29.02	18.53	8.91	14.16	10.76	14.61
ϕ						
T\K	1	2	3	4	5	6
1	25.83	13.80	14.90	7.35	9.14	8.05
2	28.41	16.29	12.81	10.90	8.14	9.71
3	25.82	12.55	14.23	10.31	9.74	15.19
4	17.68	15.84	16.93	12.86	14.69	12.46
τ						
T\K	1	2	3	4	5	6
1	33.51	17.96	13.26	12.94	7.31	10.87
2	27.43	15.31	12.45	11.62	9.72	11.48
3	26.28	17.45	10.46	11.43	13.67	10.24
4	23.67	16.41	15.94	14.94	11.40	10.53

Table 23: RBC Model: Linear Surrogate Transitions Time per Effectively Independent Sample

			γ			
T\K	1	2	3	4	5	6
1	8.15	6.37	4.54	5.08	4.01	4.20
2	9.97	6.12	5.64	5.02	5.18	4.91
3	9.19	5.26	5.30	4.59	4.69	6.61
4	9.85	7.30	6.79	5.49	7.10	7.63
			α			
T\K	1	2	3	4	5	6
1	5.78	4.76	4.85	4.14	5.28	3.36
2	9.73	5.20	5.20	4.18	3.99	4.56
3	9.08	6.23	5.29	4.65	6.17	5.60
4	7.28	5.63	6.08	4.81	8.16	6.10
			δ			
T\K	1	2	3	4	5	6
1	6.29	4.84	5.89	4.45	4.53	3.98
2	5.17	5.84	4.95	3.83	4.73	3.71
3	7.47	9.08	5.56	4.16	3.93	5.56
4	6.32	6.71	5.91	6.54	5.85	7.81
			κ			
T\K	1	2	3	4	5	6
1	7.94	5.51	4.26	4.24	6.09	5.27
2	6.80	4.47	4.40	4.32	4.49	4.32
3	7.45	5.52	5.42	5.89	5.11	6.07
4	10.21	8.17	4.17	6.76	5.22	7.19
			ϕ			
T\K	1	2	3	4	5	6
1	5.96	4.80	6.16	3.30	4.27	3.89
2	8.32	6.57	5.73	5.10	3.91	4.86
3	8.47	5.37	6.57	4.96	4.72	7.48
4	6.22	6.99	7.92	6.14	7.13	6.14
			τ			
T\K	1	2	3	4	5	6
1	7.73	6.24	5.48	5.80	3.42	5.24
2	8.04	6.17	5.57	5.45	4.67	5.74
3	8.62	7.47	4.83	5.49	6.62	5.04
4	8.33	7.23	7.45	7.13	5.53	5.18

Table 24: RBC Model: Linear Surrogate Speedup Relative to Random Walk Metropolis-Hastings

			γ			
T\K	1	2	3	4	5	6
1	0.87	1.11	1.56	1.39	1.77	1.69
2	0.71	1.16	1.26	1.41	1.37	1.44
3	0.77	1.35	1.34	1.54	1.51	1.07
4	0.72	0.97	1.04	1.29	1.00	0.93
			α			
T\K	1	2	3	4	5	6
1	1.39	1.69	1.66	1.94	1.52	2.40
2	0.83	1.55	1.55	1.92	2.02	1.76
3	0.89	1.29	1.52	1.73	1.30	1.44
4	1.10	1.43	1.32	1.67	0.99	1.32
			δ			
T\K	1	2	3	4	5	6
1	1.45	1.89	1.55	2.05	2.02	2.30
2	1.77	1.57	1.85	2.39	1.93	2.47
3	1.22	1.01	1.64	2.20	2.33	1.64
4	1.45	1.36	1.55	1.40	1.56	1.17
			κ			
T\K	1	2	3	4	5	6
1	1.42	2.05	2.65	2.66	1.85	2.14
2	1.66	2.52	2.56	2.61	2.51	2.61
3	1.51	2.04	2.08	1.91	2.21	1.86
4	1.10	1.38	2.71	1.67	2.16	1.57
			ϕ			
T\K	1	2	3	4	5	6
1	1.58	1.97	1.53	2.86	2.21	2.43
2	1.13	1.44	1.65	1.85	2.42	1.94
3	1.11	1.76	1.44	1.90	2.00	1.26
4	1.52	1.35	1.19	1.54	1.32	1.54
			τ			
T\K	1	2	3	4	5	6
1	1.16	1.43	1.63	1.54	2.62	1.71
2	1.11	1.45	1.61	1.64	1.92	1.56
3	1.04	1.20	1.85	1.63	1.35	1.78
4	1.07	1.24	1.20	1.25	1.62	1.73

A Appendix

This section proves PMMH with surrogate transitions has the correct limiting distribution. The notation, definitions, and description follow almost exactly from Andrieu, Doucet, and Holenstein (2010), with the theorem and proof are augmented in the appropriate way. We included this section here for completeness. Much like how Andrieu, Doucet, and Holenstein (2010), show PMMH is a Metropolis-Hastings algorithm on an extended state space, we show here that PMMH with marginal surrogate transitions is a surrogate transition (which can be viewed as a specific Metropolis-Hastings update) with appropriately defined target, surrogate, and proposal densities.

A.1 Preliminaries

We let the parameters be denoted by θ, where $\theta \in \Theta$. The unobserved states are the random variable $X \in \mathcal{X}$, and denote a vector of unobserved states of length P as $X_{1:P} \in \mathcal{X}^P$. Define $\bar{X}_n$ as an N vector of states at time n, $\bar{X}_n = (X_n^1, \cdots, X_n^N)$. We want to sample from the density,

$$\pi(\theta, x_{1:P}) = \gamma(\theta, x_{1:P})/Z.$$

Here, $\gamma : \Theta \times \mathcal{X}^P \to R^+$, and Z is a normalizing constant. We write,

$$\pi(\theta, x_{1:P}) = \pi(\theta)\pi_\theta(x_{1:P})$$

where,

$$\pi_\theta(x_{1:P}) = \gamma(\theta, x_{1:P})/\gamma(\theta),$$

$$\gamma(\theta) = \int_{\mathcal{X}^\mathcal{P}} \gamma(\theta, x_{1:P}) dx_{1:P},$$

and,

$$\pi(\theta) = \gamma(\theta)/Z.$$

For any $\theta \in \Theta$, we will define a family of importance sampling densities as

$$\left\{ M_n^\theta(x_n|x_{1:n-1}) \right\}; n = 1, \ldots, P\},$$

and a family of bridging densities,

$$\left\{ \pi_n^\theta(x_{1:n}); n = 1, \ldots, P \right\}.$$

Here, $\pi_0^\theta(x_{1:0}) = 1$ and $M_1^\theta(x_1|x_{1:0}) = M_1^\theta(x_1)$. Each bridging density can be written $\pi_n^\theta(x_{1:n}) = \gamma_n(\theta, x_{1:n})/\gamma_n(\theta)$, and $\pi_P^\theta(x_{1:P}) = \gamma(\theta, x_{1:P})/\gamma(\theta)$. We will call $\hat{\pi}^N(dx_{1:P})$ and $\hat{\gamma}^N(\theta)$ the SMC approximation using N particles to $\pi_\theta(dx_{1:P})$ and $\gamma(\theta)$.

Table 25: Notation Relation

General	State Space
$\pi(\theta, x_{1:P})$	$p(\theta, x_{1:T}|y_{1:T})$
$\pi_\theta(x_{1:P})$	$p(x_{1:T}|\theta, y_{1:T})$
$\pi(\theta)$	$p(\theta|y_{1:t})$
$\gamma(\theta, x_{1:P})$	$p(\theta, x_{1:T}, y_{1:T})$
$\gamma(\theta)$	$p(y_{1:T}|\theta)p(\theta)$
Z	$p(y_{1:T})$

A.2 Ancestral lineage

The resampling procedure is the operation by which offspring at time n choose their parents at time $n-1$. Let A_{n-1}^k denote the index of the parent at time $n-1$ of particle $X_{1:n}^k$. Further, let the vector of parent indicies at time $n-1$ be $\mathbf{A}_{n-1} = (A_{n-1}^1, \ldots, A_{n-1}^N)$. This vector, $\mathbf{A}_{n-1}$, parameterizes the random mapping $\{1, \ldots, N\} \to \{1, \ldots, N\}^N$.

Parents are chosen according to probability distribution $r(\cdot|\mathbf{W}_{n-1})$. Here $\mathbf{W}_{n-1}$ is the N dimensional vector of the normalized particle weights at time $n-1$, that is, $\mathbf{W}_{n-1} = (W_{n-1}^1, W_{n-1}^1, \ldots, W_{n-1}^N)$.

For a given vector of normalized weights, parents are chosen according to

$$r(\mathbf{A}_{n-1}|\mathbf{W}_{n-1}) = \prod_{k=1}^{N} \mathcal{F}(A_{n-1}^k|\mathbf{W}_{n-1}),$$

where $\mathcal{F}(A_{n-1}^k|\mathbf{W}_{n-1})$ is the density of a multinomial distribution with probabilities given by $\mathbf{W}_{n-1}$.

We let B_n^k be the index of the ancestor particle of $X_{1:P}^k$ at time n. That is, $B_P^k = k$, and,

$$B_n^k = A_n^{B_{n+1}^k}.$$

Then the vector $B_{1:P}^k$ is,

$$B_{1:P}^k = (B_1^k, \cdots, B_{P-1}^k, k).$$

The k^{th} vector from the N particle approximation of $\hat{p}^N(dx_{1:P})$ is then,

$$X_{1:P}^k = (X_1^{B_1^k}, X_2^{B_2^k}, \ldots, X_{P-1}^{B_{P-1}^k}, X_P^k).$$

Let the number of offspring of particle k at time n be $O_n^k = \sum_{m=1}^N I(A_n^m = k)$.

A.3 Generic Sequential Monte Carlo Algorithm

For any $\theta \in \Theta$, the following algorithm can be used to construct an approximation to $\pi_\theta(dx_{1:P})$. We omit dependence of the weight on θ.

- Step 1
  - Sample $X_1^k \sim M_1^\theta(\cdot)$
  - Compute and Normalize the weights

$$w_1(X_1^k) = \frac{\gamma_1(\theta, X_1^k)}{M_1^\theta(X_1^k)}$$

- Step 2, for $n = 2, \cdots, P$:
  - Sample $\mathbf{A}_{n-1} \sim r(\cdot|\mathbf{W}_{n-1})$

– Sample $X_n^k \sim M_n^\theta(\cdot|X_{1:n-1}^{A_{n-1}^k})$

– Compute and normalize the weights

$$w_n(X_{1:n}^k) = \frac{\gamma_n(\theta, X_{1:n}^k)}{\gamma_{n-1}(\theta, X_{1:n-1}^{A_{n-1}^k}) M_n^\theta(X_n^k|X_{1:n-1}^{A_{n-1}^k})}$$

$$W_n^k = \frac{w_n(X_{1:n}^k)}{\sum_{m=1}^N w_n(X_{1:n}^m)}$$

This algorithm yields particle approximation's to $\pi_\theta(dx_{1:P})$ and $\gamma(\theta)$ as

$$\hat{\pi}^N(dx_{1:P}) = \sum_{k=1}^N W_P^k \delta_{X_{1:P}^k}(dx_{1:P})$$

and

$$\hat{\gamma}^N(\theta) = \prod_{n=1}^P \frac{1}{N} \sum_{k=1}^N w_n(X_{1:n}^k)$$

respectively.

A.4 Support Definitions

For any $\theta \in \Theta$, define the support of importance density and the support of the target density as,

$$\mathcal{Q}_n^\theta = \left\{ x_{1:n} \in \mathcal{X}^n : \pi_{n-1}^\theta(x_{1:n-1}) M_n^\theta(x_n|x_{1:n-1}) > 0 \right\}$$

and,

$$\mathcal{S}_n^\theta = \left\{ x_{1:n} \in \mathcal{X}^n : \pi_n^\theta(x_{1:n}) > 0 \right\}$$

respectively. The support of the marginal density is

$$\mathcal{S} = \{\theta \in \Theta : \pi(\theta) > 0\},$$

while the support of the surrogate marginal density is

$$\tilde{\mathcal{S}} = \{\theta \in \Theta : \tilde{\pi}(\theta) > 0\}.$$

A.5 PMMH With Surrogate Transitions

For i = 1,. . . ,L

- sample $\theta' \sim q(\theta(i-1), \cdot)$
- set $z = \theta'$ with probability,

$$\alpha_1(\theta(i-1), \theta') = min\left\{1, \frac{\tilde{\gamma}(\theta')q(\theta', \theta(i-1))}{\tilde{\gamma}(\theta(i-1))q(\theta(i-1), \theta')}\right\}$$

  where $\tilde{\gamma}$ is an approximation to $\gamma(\theta)$.

- If $z \neq \theta(i-1)$,
  - Run a SMC targeting $\pi_\theta(dx_{1:P})$
  - sample $X'_{1:P}$ from $\hat{\pi}^N_{\theta'}(\cdot)$
  - set $(\theta(i), X_{1:P}(i)) = (\theta', X'_{1:P})$ with probability

$$\alpha_2(\theta(i-1), \theta') = min\left\{1, \frac{\hat{\gamma}^N(\theta')\tilde{\gamma}(\theta(i-1))}{\hat{\gamma}^N(\theta(i-1))\tilde{\gamma}(\theta')}\right\}$$

- Else set $(\theta(i), X_{1:P}(i)) = (\theta(i-1), X_{1:P}(i-1))$

A.6 Assumptions

Here we give the assumptions that we will use in the proof of the theorem.

- Assumption 1: For any $k = 1, \ldots, N$, and $n = 1, \ldots, P$, the resampling scheme satisfies

$$E(O^k_n | \mathbf{W}_n) = NW^k_n$$

and

$$r(A_n^k = m|\mathbf{W}_n) = W_n^m.$$

- Assumption 2a: $\mathcal{S} \subseteq \tilde{\mathcal{S}}$
- Assumption 2b: For any $\theta \in \tilde{\mathcal{S}}$, we have $\mathcal{S}_n^\theta \subseteq \mathcal{Q}_n^\theta$

  Note that assumptions 2a,b imply for any $\theta \in \mathcal{S}$, we have $\mathcal{S}_n^\theta \subseteq \mathcal{Q}_n^\theta$.
- Assumption 3: The Surrogate Transitions algorithm with target density $\pi(\theta)$, surrogate target density $\tilde{\pi}(\theta)$, surrogate proposal density $q(\theta, \theta')$ is irreducible and aperiodic.

Assumption 1 is the same as in Andrieu et al. It is a restriction on the resampling procedure such that it is unbiased. The second part of the assumption is needed in the proof of the theorem below. It can be enforced by a random permutation of the particles at each time period. Assumption 2a states that the support of the surrogate marginal density contains that of the true marginal density, which will be needed for the us to be able to sample from $\tilde{\pi}(\cdot)$ to draw from $\pi(\cdot)$ using surrogate transitions. Assumption 2b says that for any of the parameters contained in the support of the surrogate marginal target density, our SMC is valid to sample from $\pi_\theta(\cdot)$. Assumptions 2a,b imply the assumption 2 used in Andrieu et al. Assumption 3 states that if we were able to compute the true target marginal density, our surrogate method would create a Markov chain whose invariant distribution is $\pi(\theta)$ for almost any starting point. This is the surrogate transition analog of Assumption 3 used in Andrieu et al.

A.7 Some simplifying notation

We will define the vector $\mathbf{u}$ of random variables that make up the particle filter as

$$\mathbf{U} = (\bar{\mathbf{X}}_1, \ldots, \bar{\mathbf{X}}_P, \mathbf{A}_1, \ldots, \mathbf{A}_{P-1})$$

is a random variable defined on $\mathcal{U}$, where,

$$\mathcal{U} = \mathcal{X}^{PN} \times \{1, \cdots, N\}^{(P-1)N}.$$

Define the extended state space $(\theta, k, \mathbf{u}) \in \mathbf{X}$ as

$$\mathbf{X} = \Theta \times \{1, \cdots, N\} \times \mathcal{U}.$$

A.8 Theorem

1. Assume assumption 1, then PMMH with Surrogate Transitions is a surrogate transition on an extended space, $\mathbf{X}$ with the following definitions:

   - Surrogate target density,

     $$\tilde{\pi}(\theta, k, \mathbf{u}) = w_P^k \tilde{\pi}(\theta) \varphi^\theta(\mathbf{u})$$

     where $w_P^{k'}$ is the realized weight associated with k',

     $$\tilde{\pi}(\theta) = \tilde{\gamma}(\theta)/\tilde{Z}$$

     and,

     $$\varphi^\theta(\mathbf{u}) = \prod_{m=1}^{N} M_1^\theta(x_1^m) \prod_{n=2}^{P} r(\mathbf{a}_{n-1}|\mathbf{w}_{n-1}) \prod_{m=1}^{N} M_n^\theta(x_n^m | x_{1:n-1}^{a_{n-1}^m}).$$

   - Surrogate proposal density,

     $$\tilde{q}(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}') = w_P^{k'} \varphi^{\theta'}(\mathbf{u}') q(\theta, \theta').$$

   - Proposal density,

     $$q(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}') = S(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}')$$

     where $S(\theta, k, \mathbf{u}, \cdot)$ is the transition kernel of the surrogate Markov Chain.

   - Target density,

     $$\pi(\theta, k, \mathbf{u}) = \frac{\pi(\theta, x_{1:P}^k)}{N^P} \frac{\varphi^\theta(\mathbf{u})}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^{P} r(b_{n-1}^k | w_{n-1}) M_n^\theta(x_n^{b_n^k} | x_{1:n-1}^{b_{n-1}^k})}$$

2. Assume assumptions 2a,b and 3, then for any $N \geq 1$, then PMMH with Surrogate Transitions

generates a sequence $\{\theta(i), X_{1:P}(i)\}$ whose marginal distributions $\left\{\mathcal{L}^N\left\{(\theta(i), X_{1:P}(i)) \in \cdot\right\}\right\}$ satisfy

$$\left\|\mathcal{L}^N\left\{(\theta(i), X_{1:P}(i)) \in \cdot\right\} - \pi(\cdot)\right\| \to 0$$

as $i \to \infty$

A.9 Proof of Theorem

To prove the theorem, we need to show that the acceptance probabilities correspond to those given in the algorithm.

In the surrogate phase, we have, a standard MH update on the extended space, using the surrogate target and proposal density,

$$\tilde{\pi}(\theta, k, \mathbf{u}) = w_P^k \tilde{\gamma}(\theta)/\tilde{Z}\varphi^\theta(\mathbf{u})$$

and,

$$\tilde{q}(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}') = w_P^{k'}\varphi^{\theta'}(\mathbf{u}')q(\theta, \theta')$$

respectively.

The acceptance probability of a Metropolis-Hastings algorithm targeting $\tilde{\pi}(\theta, k, \mathbf{u})$ using proposal $\tilde{q}(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}')$ is,

$$\begin{aligned}
\alpha_1(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}') &= min\left\{1, \tfrac{\tilde{\pi}(\theta', k', \mathbf{u}')\tilde{q}(\theta', k', \mathbf{u}', \theta, k, \mathbf{u})}{\tilde{\pi}(\theta, k, \mathbf{u})\tilde{q}(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}')}\right\} \\
&= min\left\{1, \tfrac{w_P^{k'}\tilde{\gamma}(\theta')/\tilde{Z}\varphi^{\theta'}(\mathbf{u}')w_P^k\varphi^\theta(\mathbf{u})q(\theta', \theta)}{w_P^k\tilde{\gamma}(\theta)/\tilde{Z}\varphi^\theta(\mathbf{u})w_P^{k'}\varphi^{\theta'}(\mathbf{u}')q(\theta, \theta')}\right\} \\
&= min\left\{1, \tfrac{\tilde{\gamma}(\theta')q(\theta', \theta)}{\tilde{\gamma}(\theta)q(\theta, \theta')}\right\} \\
&= \alpha_1(\theta, \theta')
\end{aligned}$$

Transition kernel for the surrogate Markov Chain is,

$$S(\theta, k, \mathbf{u}, \cdot) = \alpha_1(\theta, k, \mathbf{u}, \cdot)\tilde{q}(\theta, k, \mathbf{u}, \cdot) + (1 - A(\theta, k, \mathbf{u}))\delta_{\theta, k, \mathbf{u}}(\cdot).$$

By construction, this kernel satisfies detailed balance with $\tilde{\pi}$,

$$\tilde{\pi}(\theta, k, \mathbf{u})S(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}') = \tilde{\pi}(\theta', k', \mathbf{u}')S(\theta', k', \mathbf{u}', \theta, k, \mathbf{u}).$$

This gives us a second stage proposal density of

$$q(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}') = S(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}').$$

We form the Metropolis-Hastings acceptance probability using proposal density $q(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}')$ and target density $\pi(\theta, k, \mathbf{u})$. This acceptance probability is,

$$\begin{aligned}
\alpha_2(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}') &= min\left\{1, \tfrac{\pi(\theta',k',\mathbf{u}')q(\theta',k',\mathbf{u}',\theta,k,\mathbf{u})}{\pi(\theta,k,\mathbf{u})q(\theta,k,\mathbf{u},\theta',k',\mathbf{u}')}\right\} \\
&= min\left\{1, \tfrac{\pi(\theta',k',\mathbf{u}')S(\theta',k',\mathbf{u}',\theta,k,\mathbf{u})}{\pi(\theta,k,\mathbf{u})S(\theta,k,\mathbf{u},\theta',k',\mathbf{u}')}\right\} \\
&= min\left\{1, \tfrac{\pi(\theta',k',\mathbf{u}')\tilde{\pi}(\theta,k,\mathbf{u})}{\pi(\theta,k,\mathbf{u})\tilde{\pi}(\theta',k',\mathbf{u}')}\right\} \\
&= min\left\{1, \tfrac{\pi(\theta',k',\mathbf{u}')w_P^k\tilde{\pi}(\theta)\varphi^{\theta}(\mathbf{u})}{\pi(\theta,k,\mathbf{u})w_P^{k'}\tilde{\pi}(\theta')\varphi^{\theta'}(\mathbf{u}')}\right\} \\
&= min\left\{1, \tfrac{\pi(\theta',k',\mathbf{u}')w_P^k\varphi^{\theta}(\mathbf{u})\tilde{\gamma}(\theta)}{\pi(\theta,k,\mathbf{u})w_P^{k'}\varphi^{\theta'}(\mathbf{u}')\tilde{\gamma}(\theta')}\right\} \\
&= min\left\{1, \tfrac{\hat{\gamma}^N(\theta')\tilde{\gamma}(\theta)}{\hat{\gamma}^N(\theta)\tilde{\gamma}(\theta')}\right\} \\
&= \alpha_2(\theta, \theta')
\end{aligned}$$

Where, going from lines 1 to 2 we've used the definition of $q(\theta, k, \mathbf{u}, \theta', k', \mathbf{u}')$, from 2 to 3, detailed balance of the surrogate transition, lines 3 to 5, the definition of $\tilde{\pi}(\theta, k, \mathbf{u})$ and $\tilde{\pi}(\theta)$. Finally, line 5 to 6 follows from Andrieu et al. 2010 (Theorems 2 and 4), but redo the algebra below.

$$
\begin{aligned}
\frac{\pi(\theta,k,\mathbf{u})}{w_P^k \varphi^\theta(\mathbf{u})} &= \frac{\pi(\theta,x_{1:P}^k) N^{-P} \varphi^\theta(\mathbf{u})}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^P r(b_{n-1}^k|\mathbf{w}_{n-1}) M_n^\theta(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k}) \varphi^\theta(\mathbf{u})} \\
&= \frac{\pi(\theta,x_{1:P}^k) N^{-P}}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^P r(b_{n-1}^k|\mathbf{w}_{n-1}) M_n^\theta(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k}) w_P^k} \\
&= \frac{\pi(\theta,x_{1:P}^k) N^{-P}}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^P M_n^\theta(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k}) \prod_{n=1}^{P-1} w_n^{b_n^k} w_P^k} \\
&= \frac{\pi(\theta,x_{1:P}^k) N^{-P}}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^P M_n^\theta(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k}) \prod_{n=1}^P w_n^{b_n^k}} \\
&= \frac{\pi(\theta,x_{1:P}^k) N^{-P} \prod_{n=1}^P \left\{\sum_{m=1}^M w_n(x_{1:n}^m)\right\}}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^P M_n^\theta(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k}) \prod_{n=1}^P w(x_n^{b_n^k})} \\
&= \frac{\pi(\theta,x_{1:P}^k) \prod_{n=1}^P \frac{1}{N} \left\{\sum_{m=1}^M w_n(x_{1:n}^m)\right\}}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^P M_n^\theta(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k}) \prod_{n=1}^P w(x_n^{b_n^k})} \\
&= \frac{\pi(\theta,x_{1:P}^k) \prod_{n=1}^P \frac{1}{N} \left\{\sum_{m=1}^M w_n(x_{1:n}^m)\right\}}{M_1^\theta(x_1^{b_1^k}) \prod_{n=2}^P M_n^\theta(x_n^{b_n^k}|x_{1:n-1}^{b_{n-1}^k}) \prod_{n=1}^P \left(\frac{\gamma_n(\theta,X_{1:n}^{b_n^k})}{\gamma_{n-1}(\theta,X_{1:n-1}^{b_{n-1}^k}) M_n^\theta(X_n^k|X_{1:n-1}^{b_{n-1}^k})} \right)} \\
&= \frac{\pi(\theta,x_{1:P}^k) \prod_{n=1}^P \frac{1}{N} \left\{\sum_{m=1}^M w_n(x_{1:n}^m)\right\}}{\gamma_P(\theta,x_{1:P}^{b_P^k})} \\
&= \frac{1}{Z} \frac{\gamma(\theta,x_{1:P}^k) \prod_{n=1}^P \frac{1}{N} \left\{\sum_{m=1}^M w_n(x_{1:n}^m)\right\}}{\gamma(\theta,x_{1:P}^k)} \\
&= \frac{\hat{\gamma}^N(\theta)}{Z}
\end{aligned}
$$

The proof of the second part of the theorem follows directly from Andrieu and Roberts (2009), Theorem 1.