Institute for
Monetary and
Financial
Stability

MÁTYÁS FARKAS, BALINT TATAR

Bayesian Estimation of DSGE Models with Hamiltonian Monte Carlo

Institute for Monetary and Financial Stability
GOETHE UNIVERSITY FRANKFURT

WORKING PAPER SERIES NO. 144 (2020)

Institute for Monetary and Financial Stability
Goethe University Frankfurt
House of Finance
Theodor-W.-Adorno-Platz 3
D-60629 Frankfurt am Main
www.imfs-frankfurt.de **|** info@imfs-frankfurt.de

Bayesian Estimation of DSGE Models with Hamiltonian Monte Carlo[*]

Mátyás Farkas[†]
European Central Bank

Balint Tatar[‡]
IMFS and Goethe-University Frankfurt

August 31, 2020

Abstract

In this paper we adopt the Hamiltonian Monte Carlo (HMC) estimator for DSGE models by implementing it into a state-of-the-art, freely available high-performance software package. We estimate a small scale textbook New-Keynesian model and the Smets-Wouters model on US data. Our results and sampling diagnostics confirm the parameter estimates available in existing literature. In addition we combine the HMC framework with the Sequential Monte Carlo (SMC) algorithm which permits the estimation of DSGE models with ill-behaved posterior densities.

Keywords: DSGE Estimation, Bayesian Analysis, Hamiltonian Monte Carlo

JEL-Codes: C11,C15,E10

[*] We thank Nils Bertschinger from FIAS, Michael Binder, Alexander Mayer-Gohde, Volker Wieland from IMFS, and Michele Lenza, Luca Dedola, Péter Karádi and participants of RCC5 seminar 11th July 2019 in the ECB the for very helpful comments and discussions. This paper builds in parts on the PhD thesis of Balint Tatar. All errors are our own. The views expressed in this paper belong to the authors and are not necessarily shared by the European Central Bank.

[†] matyas.farkas@ecb.europa.eu

[‡] balint.tatar@hof.uni-frankfurt.de

1 Introduction

Dynamic Stochastic General Equilibrium (DSGE) models have been shaping modern macroeconomic theory since the seminal contribution of Kydland and Prescott (1982). During the past decades DSGE models have become the workhorse framework for the analysis of economic fluctuations and were extended to achieve a sufficiently proper fit of empirical data. Finding the right link of the model to the data became an increasingly challenging and complex task. Fernandez-Villaverde et al. (2016) provides an excellent summary of the methodology and the transition from small scale calibrated models to the state-of-the-art likelihood-based estimation of medium to large scale DSGE models. The pioneer work on Bayesian DSGE model estimation in the form as it is conducted today dates back to Schorfheide (2000) and Otrok (2001). The original estimation framework was built around the Kalman filter and the Metropolis-Hastings algorithm, both originally developed as tools for applied physics. These methods also prepared the ground for the Markov Chain Monte Carlo (MCMC) algorithms. With the popularity of the DSGE literature and the increasing complexity of the models also more sophisticated estimation methods had to be developed, e.g. the Sequential Monte Carlo Method (SMC) which was first used for posterior inference by Creal (2007) and then formalized by Herbst and Schorfheide (2014).

A main criticism of the meanwhile established baseline framework using MCMC algorithms, also readily available in Dynare, remained unaddressed: the simulated sample draw often suffers from considerably high autocorrelations, and thus will have a very small effective sample size. A common approach to tackle this shortcoming is to run longer chains and to consider only each n-th draw by discarding the rest to obtain uncorrelated samples.[1] In theory the MCMC algorithm converges under certain regularity conditions asymptotically to the target density, in practice the convergence might occur at a very slow pace. Thinning the Markov Chain will render an efficient sampling impractical, as it can easily become time consuming, particularly when the dimension of the model to be estimated is high. A key question with respect to the MCMC algorithm, whether the Markov chain has already converged to the target distribution, remained unanswered. Unfortunately, there

[1] where n usually equals to a multiple of 100

is no single way to address the latter issue, as pointed out by Brooks and Gelman (1998), instead "The idea is to use a wide variety of diagnostics so that if all appear to suggest that convergence has been achieved, then the user can have some confidence in that conclusion". However, even if standard diagnostics suggests that convergence has not been reached yet, it will be challenging to explore the reason for non-convergence. Therefore, as also suggested by Betancourt (2018) better methods are needed to explore the typical set by exploiting the geometry of the target distribution.

In higher dimensional spaces the standard random walk MCMC algorithm will explore the typical set only slowly. Large transitions from one point to the other in the typical set will not be possible, as the number of directions to move the chain increases exponentially with the dimension. A straightforward algorithm making use of the information in the geometry of the typical set is the Hybrid Monte Carlo algorithm. It became also known as the Hamiltonian Monte Carlo (HMC) algorithm and is the new standard in high dimensional numerical simulation where the gradient of the target density can be evaluated. Similarly to the Kalman filter and MCMC, it has its roots in physics, dating back to Duane et al. (1987) originally designed for the numerical simulation of lattice field theory simulations of quantum chromodynamics.

Due to the accessibility of an advanced software package for Bayesian estimations, STAN, which implemented the HMC algorithm, the methodology is presently used by many researchers in various fields. The current paper presents its first detailed implementation for macroeconomic modeling, and in particular for DSGE estimation. HMC has been shown to have significantly better sampling properties than the baseline algorithm used for the estimation of DSGE models which is well documented in the literature, see e.g. Neal (2011). In Herbst and Schorfheide (2015), a recent textbook on Bayesian DSGE estimation, the advantages of the HMC algorithm are also acknowledged and research to make progress into this direction is also encouraged. Fortunately the STAN software package is a suitable tool to deal with complex models and symbolic differentiation which make the accurate implementation feasible, therefore there is no need to rely on approximations. It also comes along with a set of powerful diagnostics readily available, which enables to verify whether the typical set has been explored appropriately.

The main purpose of this paper is to implement the HMC method in order to estimate log-linearized DSGE models and illustrate how convergence diagnostics can indicate model misspecification. As an extension, we will also turn to estimate pathological posterior densities in established DSGE models by combining the HMC algorithm with the SMC framework.

This is the first paper to present results of applying the HMC algorithm to DSGE models.[2] In particular we implement HMC using STAN because it comes with the following advantages: First, it is implemented in C++, a low-level high-performance programming language. Second, it includes automated differentiation enabling the calculation of complicated differentials. Third, it features a very powerful diagnostic and visualization toolkit.

Although the implementation of the HMC algorithm for DSGE models paves the way for a more sophisticated exploration of the typical set and provides access to powerful diagnostics it has also a significant drawback. It is well known that the HMC algorithm fails to deal with multimodal target densities which occur e.g. when less informative priors are used to estimate the Smets-Wouters model (Herbst and Schorfheide (2014)). In case the modes are separated by large energy-barriers from each other, in particular, the posterior likelihood function has no support between modes, the algorithm will get stuck in one mode and the chain will not be able to escape in a foreseeable time. Nevertheless, it is possible to include the HMC sampling algorithm into the SMC framework adopted by Herbst and Schorfheide (2014) and explore bimodal densities as well, as with computing power available today the algorithm is feasible.

The remaining part of this paper is organized as follows. In Section 2 we review briefly the workhorse Bayesian estimation framework. In Section 3 we present the HMC algorithm and summarize the underlying theoretical considerations. Section 4 describes the way to implement the DSGE estimation framework and discusses some computational issues. Section 5 presents the estimation results of a textbook small scale New-Keynesian DSGE model and the Smets and Wouters (2007) model. Section 6 extends the paper by combining algorithms to estimate ill-behaved posterior

[2] In contemporanous work Fernandez-Villaverde and Rubio-Ramirez (2020) propose the application of HMC for DSGE estimation, furthermore an incomplete working paper by Goodrich and Montes-Galdon (retrieved on 25 August, 2020) exists, although without results.

densities. Section 7 concludes the paper.

2 Bayesian Estimation of DSGE Models: A Brief Review

In this chapter we briefly review the main estimation framework used for MCMC-type Bayesian DSGE model estimation. A more extensive treatment can be also found in the excellent work of Herbst and Schorfheide (2015).

In order to estimate a Bayesian model, the first step is to specify the joint distribution of the data and the model parameters. The aim is to obtain the posterior density, that is, the distribution of the model parameters given the data, denoted by $p(\theta|Y)$ which can be also expressed by the means of Bayes' rule as follows:

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

$p(Y|\theta)$ is referred to as the likelihood function and $p(\theta)$ is the prior distribution. Typically, in a Bayesian estimation, the *a priori* beliefs about the parameter vector θ, being equivalent to the the prior distribution, are updated using the likelihood function. The posterior distribution then comprises the state of knowledge about θ consisting of the a priori beliefs and the information available in the data.

To specify a likelihood function conditioned on the parameters and turn a DSGE model into a Bayesian model, a formal representation of the DSGE model is needed. Hence we need to solve for the law of motion of the model variables. There exists a variety of solution methods to approximate locally the solution, e.g. Blanchard and Kahn (1980), Binder and Pesaran (1997), King and Watson (1998), Uhlig (1999), Anderson (2000), Kim (2000), Christiano (2002). A popular solution technique for a linearized DSGE model was proposed by Sims (2002) which starts with the following representation of the DSGE model:

$$\Gamma_0 s_t = \Gamma_1 s_{t-1} + \Psi \epsilon_t + \Pi \eta_t \tag{2}$$

where ϵ_t is the vector of structural shocks and η_t the one step ahead rational expectation forecast errors, $x_t - \mathbb{E}_{t-1} x_t$. The solution is based on the QZ-decomposition, also referred to as generalized eigenvalue problem $Av = \lambda Bv$ where A and B are square matrices. If the above system has a unique stable solution then it can be represented in the following VAR-form:

$$s_t = G_0(\theta)s_{t-1} + G_1(\theta)\epsilon_t \tag{3}$$

Applying the solution method proposed by Sims (2002), alternatively any other solution algorithm, a state space representation can be obtained to specify the likelihood function. In this setup the VAR-form from above represents the transition equation which is linked to the data by means of the measurement equation:

$$y_t = H_0(\theta) + H_1(\theta)t + H_2(\theta)s_t + u_t \tag{4}$$

The state space representation allows to express the joint density function for the the observed data and the DSGE-model variables where the latter are unobserved:

$$p(Y_{1:T}, S_{1:T}|\theta) = \prod_{t=1}^{T} p(y_t, s_t|Y_{1:t-1}, S_{1:t-1}, \theta) = \prod_{t=1}^{T} p(y_t|s_t, \theta)p(s_t|s_{t-1}, \theta) \tag{5}$$

where $p(y_t|s_t, \theta)$ and $p(s_t|s_{t-1}, \theta)$ are the conditioned probabilities given the observation and the state equation. To obtain the desired likelihood function the unobserved states, s_t, have to be integrated out. For log-linearized DSGE models with Gaussian disturbance one can use the Kalman filter to obtain the conditional expectations and variances of the observables and finally the log-likelihood function. Once the prior distribution of the parameters is specified one can set up the Random-Walk Metropolis Hastings Algorithm to sample from the posterior density. The algorithm is summarized below:[3]

Algorithm 1: Random-Walk Metropolis Hastings

1. Maximize $\ln p(Y|\theta) + \ln p(\theta)$ by a numerical algorithm to obtain the posterior mode, denoted by $\tilde{\theta}$. (This step involves the application of the solution algo-

[3] This code summarizes the main steps described also extensively in Herbst and Schorfheide (2015)

rithm of the DSGE model, setting up the state space representation and the calculation of the likelihood by means of the Kalman filter)

2. Compute $\tilde{\Sigma}$, the inverse of the Hessian at $\tilde{\theta}$
3. Initialize a starting value or draw $\theta^{(0)}$ from the proposal density $q(\theta^{(0)}|\tilde{\theta})$ (in this case $N(\tilde{\theta}, c_0^2\tilde{\Sigma})$)
4. For $i = 1, ..., N$ draw θ' from the proposal distribution $\mathcal{N}(\theta^{(n-1)}, c_0^2\tilde{\Sigma})$.
5. Solve the DSGE model for θ' and build the new state space representation.
6. Calculate $p(Y|\theta')$ and $p(\theta')$ (by means of the Kalman filter)
7. Accept θ', that is, $(\theta^{(n)} = \theta')$, with probability $\min\big\{1, f(\theta^{(n-1)}, \theta'|Y)\big\}$ and reject $(\theta^{(n)} = \theta^{(n-1)})$ otherwise where

$$f(\theta^{(n-1)}, \theta'|Y) = \frac{p(Y|\theta')p(\theta')q(\theta'|\theta^{(n-1)})}{p(Y|\theta^{(n-1)})p(\theta^{(n-1)})q(\theta^{(n-1)}|\theta')}$$

8. Estimate the posterior expected value of the function $h(\theta)$ by $\frac{1}{N}\sum_{i=1}^{N} h(\theta^{(i)})$

In the above case the proposal density $q(\cdot|\cdot)$ is chosen to be the normal distribution with expected value $\theta^{(n-1)}$ which implies that the proposals follow a random walk. In addition, as the density function of the normal distribution is symmetric, the proposal densities cancel. Also the scaling paramater, c_0, should be chosen in a way that the acceptance ratio equals to 23.4%, which was proven to be the optimal acceptance ratio, see Roberts et al. (1997). In practice however this parameter is chosen in a way that the acceptance ratio lies between 0.2 and 0.4.

There are several other modified versions of the MH algorithm. For example, the Block-MH algorithm breaks the parameter vector into blocks and as its name suggests it updates at most only one block of the parameters at once. This scheme can be further extended by randomizing the break-up of the parameter vector into blocks in each step. A further possibility to improve the algorithm is to apply a more sophisticated proposal density. In particular, the Metropolis-Adjusted Langevin (MAL) algorithm suggests to choose again a normal distribution or student distribution, where the latter would have been appropriate also in the above case, however the expected value should be adjusted by one step into the direction of the gradient of the negative log-posterior. Intuitively, this algorithm accounts for the shape of the posterior density and pushes the chain, thus the new proposal for the parameter, toward regions with higher probability density. It is common to choose a scaled version of the identity matrix as the variance. Both the step size into the direction of the gradient and the scaling of the variance are subject to fine-tuning. The MH-Newton algorithm only differs from the latter modified MAL-algorithm in that instead of the Hessian at the posterior mode the Hessian at $\theta^{(n-1)}$ is taken. For further discussion of estimation methods we refer to the work of Herbst and Schorfheide (2015).

Although first-order linear approximations around the non-stochastic steady state are popular, in a number of cases more elaborate estimation methods are required. For example, when higher order approximations are necessary to capture the impact of shocks on endogenous variables, then the state space will be non linear. To evaluate the likelihood in this more complex case particle filters were proposed in literature. At the same time, particle filters are also applied if the posterior likelihood is ill shaped, e.g. Sequential Monte Carlo Methods (SMC), which may even occur when standard models are estimated using first order linear approximations. Our extension of the HMC with SMC falls into the latter application.

3 The Hamiltonian Monte Carlo Method

This section of the paper will provide an introduction into the HMC framework and is aimed to offer some intuition to the reader while we also reveal some main theoretical aspects of the methodology.[4] Similarly to alternative approaches from above the HMC algorithm builds on the information provided by the gradient of the log-posterior density function. In particular, this algorithm uses the information in the geometry of the target distribution and its main advantage is that by means of the Hamiltonian equations, which concept was borrowed from physics, the algorithm enables to propose a new parameter draw θ' which is distant from the current θ while it maintains a sufficiently high acceptance rate.

[4] More extensive treatment is provided e.g. in Neal (2011) or Betancourt (2018).

In physics researchers usually model the evolution of a mechanical system over time given a particle's position and momentum by functions measuring its potential and kinetic energy. This can be visualized the easiest by a puck in a frictionless environment sliding over a surface. Here the potential energy of the puck is described by the function $U(q)$, depending proportionally on the position and its kinetic energy $K(p)$ which is equal to $|p|^2/(2m)$ where m corresponds to the mass of the puck. Further classical examples are a bouncing ball, a pendulum or an oscillating spring. In a classical physical system which is isolated from any outside force the energy of the puck, being the sum of potential and kinetic energy remains constant. Hence if the puck moves along a path with positive slope, its velocity p/m will decrease. The above isolated system in physics, among others the evolution of the position and velocity can be fully described by the Hamiltonian equation $H(q,p)$, also referred to as the total energy function. In classical mechanics the Hamiltonian equation is obtained from Lagrange's equation, a reformulation of the Newtonian mechanics, by a Legendre transformation, where $H : \mathbb{R}^{2d} \to \mathbb{R}$ with $\mathbb{R}^{2d}$ being the phase space and d the degrees of freedom. This Hamiltonian framework can be easily translated also to MCMC applications outside physics, by regarding the position of the puck q as the variables of interest of which posterior distribution a sample should be drawn. The main idea is to extend Bayes' Theorem $p(\theta|Y) \propto p(\theta)p(Y|\theta)$ by an auxiliary vector α of momentum variables to obtain the joint posterior density $p(\theta,\alpha|Y) \propto p(\theta,\alpha)p(Y|\theta,\alpha)$ of θ and α. To each parameter θ_i one momentum variable α_i is assigned. The auxiliary variables are a priori independent of θ and Y implying that $p(\theta,\alpha|Y) \propto p(\theta)p(\alpha)p(Y|\theta)$.

The change in the current position q and momentum p, being both of dimension d respectively, over time is characterized by the partial derivatives of the Hamiltonian equation:

$$\frac{dq_i}{dt} = \frac{\partial H(q,p)}{\partial p_i} \qquad \forall i = 1, ..., d \tag{6}$$

$$\frac{dp_i}{dt} = -\frac{\partial H(q,p)}{\partial q_i} \qquad \forall i = 1, ..., d \tag{7}$$

where $2d$ equals the full dimension of the system. The equations of motion can be presented also in a more compact way by defining $z := (q, p)$ such that

$$\frac{dz}{dt} = J\nabla H(z) \tag{8}$$

with $\nabla H(z)$ being the gradient of the Hamiltonian system and J a matrix of dimension $2d \times 2d$:

$$J = \begin{bmatrix} 0_{d\times d} & I_{d\times d} \\ -I_{d\times d} & 0_{d\times d} \end{bmatrix}$$

The solution to this system of differential equations can be regarded as a mapping $F_s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ with $(q, p)(t) \to (q, p)(t + s)$ such that the Hamiltonian equations describe the law of motion of the system from t to $t + s$.

The Hamiltonian measures total energy and for the HMC algorithm it takes an additive form

$$H(p, q) = U(q) + K(p). \tag{9}$$

The kinetic energy $K(p)$ is usually defined as

$$K(p) = p^T M^{-1} p/2 \tag{10}$$

where M is referred to as the "mass" matrix which is typically diagonal, and is often a scalar multiple of the identity matrix.

The Hamiltonian system has four key properties which allow for using it for the construction of an MCMC algorithm. Firstly, the Hamiltonian does not change over time, that is, $dH/dt = 0$, which is crucial to ensure that the acceptance probability equals always one.

Secondly, the Hamiltonian system preserves the volume in the phase space. Without going too deeply into details of volume measures of a phase space, this property is crucial in the sense that there is no need to account for a change in the volume in the acceptance probability.

Thirdly, the Hamiltonian system is symplectic. Formally this corresponds to the property that the Jacobian $B_s := \mathrm{D}T_s$ of the mapping T_s, satisfies the following equation:

$$B_s^T A B_s = A \tag{11}$$

where A is in general a fixed $2d \times 2d$, non-singular, skew symmetric matrix. Usually, the matrix J from above is chosen for A. The determinant of the matrix J is unity and it also holds that $J^{-1} = J^T = -J$. The symplecticness condition also implies that the mapping is volume preserving as from the equation above it immediately follows that $|det(B_s)| = 1$. Yet, the above property is stronger than just volume preservation if $d > 1$. This property is important, as in practice Hamiltonian equations can be solved only by numerical integration. Although a large number of numerical integrators exist, most of them are prone to accumulate approximation errors such that the accuracy of the solution will be significantly impaired. However, to solve for the Hamiltonian, symplectic integrators can be applied having the advantage that the approximated trajectory does not drift away from the true one.

Finally, the mapping T_s defined above, is reversible, that is T_s has an inverse T_{-s} which is exactly the negation of the time derivatives in the Hamiltonian equations. Considering again the example with the puck, one can imagine this as stopping the puck at $q(t+s)$ and hit it into the opposite direction with the same impulse. In case $K(p) = p^T M^{-1} p/2$ one can negate $K(p)$, apply T_s and then negate again $K(p)$ to obtain the original $(q,p)(t)$ where the puck departed from. The reversibility property is crucial when proving the detailed balanced condition in the probabilistic framework which ensures together with ergodicity that the HMC converges to the invariant distribution.

To apply this framework to a probabilistic setting borrowing one further concept from statistical mechanics is necessary referred to as the "cannonical" distribution at a given temperature. This concept describes possible states of a mechanical system which is at thermal equilibrium at temperature T. For the latter purpose the following distribution is used:

$$P(x) = \frac{1}{Z} e^{-E(x)/T} \tag{12}$$

where we assume that the energy $E(x)$ and its gradient can be evaluated. Any particular density $P(x)$ can be adopted to the above scheme by setting $E(x) =$

$-\log\, P(x) - \log Z$ and $T = 1$. The HMC algorithm translates this framework into an MCMC-sampling algorithm by applying the Hamiltonian equation as the total energy function for the joint state (p, q) which results in the following cannonical distribution:

$$P(q,p) = \frac{1}{Z} e^{-H(q,p)/T} \tag{13}$$

with $H(q,p) = U(q) + K(p)$ we obtain

$$P(q,p) = \frac{1}{Z} e^{-U(q)/T} e^{-K(p)/T} \tag{14}$$

Setting for $U(q)$ the target density $p(Y|\theta)p(\theta)$ and for $K(p)$ the kinetic energy function allows to define an algorthim which samples from the distribution of interest. The iteration is carried out in three steps:

Algorithm 2: Hamiltonian Monte Carlo

1. Draw a momentum vector p' from its multivariate normal distribution which can be carried out by Gibbs-sampling.
2. Draw the position vector θ' by applying the Hamiltonian equations deterministically.
3. Metropolis-Hastings step: accept the new proposal and set $\theta^{(n+1)} = \theta'$ with probability $\min[1, \exp(-(U(q') - U(q) + K(p') - K(p)))]$.

As the total energy in the system remains constant, in theory the proposal obtained by applying the Hamiltonian equations is always accepted. To obtain a sample from the target distribution one simply omits the sampled momenta. It is well known that to show that the resulting Markov chain converges to the target distribution it has to be ergodic and has to fulfill the detailed balance condition:

$$P(q,p)P_K((q,p) \to (q',p')) = P(q',p')P_K((q',p') \to (q,p)) \tag{15}$$

where P_K is the HMC kernel. The key property that allows to proof that the detailed balance condition holds is reversibility of the Hamiltonian system. In addition, the symplecticness of the numerical integrator to be used ensures that detailed balance holds even if the solution is approximated numerically. A formal proof is available in Duane et al. (1987). As regards ergodicity the original paper does not provide any insights, instead it assesses using an example in compact quantum electrodynamics "Whether or not this idea works in practice...". Proving ergodicity for the HMC algorithm involves deep knowledge in probability theory. Very loosely spoken ergodicity implies that the Markov chain will not be trapped in a subset of the parameter space, instead it will reach all possible states again and again, hence it will asymptotically converge to the invariant distribution. Neal (2011) also points out that in theory it is possible that ergodicity fails once as a fixed number of integration steps is used for the numerical approximation of the solution and illustrates this based on a short example. Mackenzie (1989) proposes that by randomizing the length of the Hamiltonian trajectory this issue can be eliminated while recently general conditions for the convergence of the HMC algorithm could be proved, see e.g. Livingstone et al. (2018) and Durmus et al. (2019).

4 Implementation into STAN

The STAN software package is a state-of-the-art probabilistic programming language for Bayesian inference written in C++ language. It allows users to set up hierarchical Bayesian models in a convenient statistical language and provides thereby an easy to apply interface to the HMC algorithm for complex models. C++ is a machine-oriented programming language and is also often applied to perform computationally highly intensive calculations due to its performance, also necessary to estimate a DSGE model. Yet, this comes at the cost of complexity in terms of the programming language which the STAN interface remedies and makes this powerful and complex concept available to researchers, working out-of-the-box.

4.1 Features and Calibration

The Hamiltonian equations typically describe the dynamics of a system in continuous time. However, in practice it will be necessary to apply a discrete-time approximation in order to calculate the new position, the momentum, the potential energy and the kinetic energy. One of the key challenges lies in the accurate solution of the Hamiltonian equations. As the Hamiltonian system is symplectic a dedicated class of simplectic integrators can be applied enabling the calculation of an accurate discrete time solution for the Hamiltonian trajectory in the phase space. The main advantage of the latter class of integrators is that the approximated trajectory does not drift away from the true one, even if integration is carried out over a long distance in time. STAN uses a simple implementation referred to as the "leapfrogging" algorithm to solve for the discrete-time approximation of the Hamiltonian equations which is summarized by the following algorithm:

Algorithm 3: Leapfrogging Algorithm

```
1. $p_i(t+\epsilon/2) = p_i(t) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t))$
2. $q_i(t+\epsilon) = q_i(t) + \epsilon\frac{p_i(t+\epsilon/2)}{m_i}$
3. $p_i(t+\epsilon) = p_i(t+\epsilon/2) - (\epsilon/2)\frac{\partial U}{\partial q_i}(q(t+\epsilon))$
```

Although at first glance the above algorithm is easy to implement, it generates a further challenge, especially when applied in the context of DSGE estimation. In general it requires to evaluate the gradient of the log-posterior which calculation might be extremely difficult and time intensive. Gradients obtained by numerical approximations can be inaccurate or computationally demanding when the parameter space is large. One of the main advantages of STAN is that it applies a reverse-mode automatic differentiation and C++ template metaprogramming. Automatic differentiation requires only a limited number of differentiation rules and the gradient is constructed via the chain rule by creating an expression tree backwards starting with the last expression in the likelihood function. For example, STAN is capable of differentiating any iterative algorithm which will turn out to be handy when implementing the estimation of DSGE models. Therefore, there is no need to specify any derivatives by the user, yet in theory it is possible to write wrappers if a closed form solution of the partial derivatives is available. Although the derivation of the log-likelihood function which also depends on the solution of the DSGE model is computationally involved for a mid-scale DSGE model, the latter is performed by STAN in a highly efficient way.

The performance of the algorithm is sensitive to the selection of two parameters: the stepsize, ϵ and the number of steps in time, L. The selection of the discrete time approximation to calculate the integral, ϵ, is of crucial importance. If the approximation is overly fine then the proposal to update θ will be accepted with very high probability, yet $||\theta' - \theta||$ will be small and the chain will explore the parameter space very slowly. If ϵ is too high, the approximation of the true solution to the Hamiltonian equation will become imprecise, or may even diverge, and θ' will be unlikely to be accepted. Furthermore, it can also occur that the Markov chain will fail completely to explore certain regions of the posterior. Usually the posterior likelihood function exhibits regions with both lower and larger curvature especially if the model is more complex, therefore one has to strike the right balance when setting ϵ. A further strength of STAN lies in the feature that ϵ is calibrated automatically during the warmup period and fixed afterwards, yet the user retains the option to set the parameter manually. STAN aims to calibrate ϵ in a way that the acceptance rate lies at 80% being significantly higher than 23.4% in the Random Walk Metropolis algorithm. In case the divergence rate remains still high the option remains to instruct the automated calibration mechanism in STAN manually to target higher acceptance ratios.

It is also crucial to select a suitable number of steps, L, to be conducted by the leapfrogging algorithm in order to explore the state space systematically as pointed out by Neal (2011). An inappropriately low L will cause θ' to be little distant from θ, hence the algorithm will exhibit random walk behavior and the Markov chain will explore the parameter space again inefficiently slowly, as also highlighted by Hoffman and Gelman (2014). If L is too large, computational resources are wasted as the acceptance rate does not depend systematically on the number of steps. A further built-in feature of STAN is that it optimizes automatically the number of steps by means of the No U-Turn Sampling (NUTS) algorithm, see Hoffman and Gelman (2014). The intuition of NUTS is to use the leapfrog integrator to iterate on θ

both in positive and negative direction, that is first running forwards or backwards 1 step, then forwards or backwards 2 steps, then forwards or backwards 4 steps and so on. The doubling process implicitly builds a balanced binary tree and will continue until some proposal moves backwards to its original point of departure, thus it would make a U-turn and moves again towards the point of departure, θ. STAN applies then a slice sampling algorithm to select randomly a point along the Hamiltonian trajectory which adds complexity, yet it is necessary to preserve certain crucial properties of the Markov chain. Finally, it accepts the proposal with the probability given in the Metropolis-Hastings step.

The mass matrix M, being typically a diagonal matrix is also tuned automatically during the warmup. Here, the user is also allowed to tune M manually, however the automated tuning process of STAN operates sufficiently well.

A further useful feature which is implemented into STAN is that it is able to remedy the weakness that the HMC algorithm works only if the support of the posterior density spans the entire parameter space. If a proposal is accepted in a region where the mass of the parameter space is zero, the gradient will also become zero or undefined and the chain will get stuck. A straightforward approach to avoid this issue is to restrict the parameter space and let the Markov Chain bounce back from the boundary by negating the momentum. However, instead STAN reparametrizes θ as a function of unbounded parameters. This occurs typically when standard deviations are estimated. The latter approach obviously involves the calculation of the Jacobian, however this is carried out again automatically by STAN.

As already pointed out, the main advantage of the HMC algorithm is that it uses gradient information to explore suitable paths on which the level of energy remains constant and finds new proposals θ' which are distant from the most recent draw θ. However, it comes along with the difficulty that the gradient of the log-likelihood function needs to be evaluated. Recall that the popular solution algorithm to DSGE models proposed by Sims (2002) uses a QZ-decomposition where the entries of the matrices can become complex. A main shortcoming of STAN is that it is not capable to execute calculation with complex numbers, hence a QZ decomposition cannot be implemented. Furthermore, it might be also challenging to build the derivatives when complex numbers are involved. To overcome this difficulty we need to rely on a DSGE model solution algorithm which makes it feasible to the automated differentiation implemented in STAN to calculate the gradient. The reverse-mode automatic differentiation relies on the chain rule when building the symbolic derivative, hence it is capable to handle any matrix iteration algorithm where no complex numbers are involved. A straightforward and easy to understand algorithm to be applied for this purpose is the Binder-Pesaran solution algorithm.

4.2 Binder-Pesaran Algorithm

The main idea of the Binder and Pesaran method is to tweak s_t such that the reshuffled form will not contain the s_{t-1} term and the system can be solved forward in case it has a unique stationary solution. A short recap of the main steps of the algorithm looks as follows. Without loss of generality let the system be given in a slightly different form:

$$\mathrm{M}_{00}s_t = \mathrm{M}_{10}s_{t-1} + \mathrm{M}_{01}\mathbb{E}_t s_{t+1} + \mathrm{M}_s\epsilon_t \tag{16}$$

In the following it is assumed that M_{00} is invertible which implies that

$$s_t = \mathrm{A}s_{t-1} + \mathrm{B}\mathbb{E}_t s_{t+1} + \mathrm{W}\epsilon_t \tag{17}$$

with $\mathrm{A} = \mathrm{M}_{00}^{-1}\mathrm{M}_{10}$, $\mathrm{B} = \mathrm{M}_{00}^{-1}\mathrm{M}_{01}$ and $\mathrm{W} = \mathrm{M}_{00}^{-1}\mathrm{M}_s$. The assumption that M_{00} has to be invertible is a little restrictive at first sight, yet it is less of a concern. In particular, the matrix can become only non-invertible when a linear combination of future expectations in $t+1$ depend only on linear combinations of past values of endogenous variables and shocks. However, this does not seem to be an issue in practice.[5] Now let $S_t := s_t - \mathrm{C}s_{t-1}$ with S_t and C to be determined. s_t can be expressed from the definition and substituted above to obtain

$$S_t + \mathrm{C}s_{t-1} = \mathrm{A}s_{t-1} + \mathrm{B}(\mathbb{E}_t S_{t+1} + \mathrm{C}s_t) + \mathrm{W}\epsilon_t \tag{18}$$

Collecting and rearranging terms yields

[5] Even if this feature of the algorithm postulated an issue a number slightly larger than machine precision could be added to the matrix which does not influence results.

$$(\mathrm{I} - \mathrm{BC})S_t = (\mathrm{BC}^2 - \mathrm{C} + \mathrm{A})s_{t-1} + \mathrm{B}(\mathbb{E}_t S_{t+1}) + \mathrm{W}\epsilon_t \tag{19}$$

The backward looking component will drop out of the equation if $\mathrm{BC}^2 - \mathrm{C} + \mathrm{A} = 0$. The solution of this quadratic matrix equation can be easily obtained by iterating on

$$\mathrm{C} = (\mathrm{I} - \mathrm{BC})^{-1}\mathrm{A} \tag{20}$$

The quadratic matrix equation could be also solved by other techniques from linear algebra, however this would again involve the calculation of generalized eigenvalues. After obtaining the solution for C the system of equation looks as follows:

$$S_t = \underbrace{(\mathrm{I} - \mathrm{BC})^{-1}\mathrm{B}}_{=:F}(\mathbb{E}_t S_{t+1}) + \underbrace{(\mathrm{I} - \mathrm{BC})^{-1}\mathrm{W}\epsilon_t}_{=:\zeta_t} \tag{21}$$

If all eigenvalues of the matrix F are stable the equation can be easily solved forward to obtain

$$S_t = \sum_{i=0}^{\infty} F^i \mathbb{E}_t \zeta_{t+i} \tag{22}$$

In this case one obtains a unique stable solution of the system. Plugging back into the definition of S_t the solution of the original model is immediately obtained:

$$s_t = \mathrm{C}s_{t-1} + \sum_{i=0}^{\infty} F^i (\mathrm{I} - \mathrm{BC})^{-1}\mathrm{W}\mathbb{E}_t \epsilon_{t+i} \tag{23}$$

If structural shocks are uncorrelated then the above formula boils down to:

$$s_t = \mathrm{C}s_{t-1} + (\mathrm{I} - \mathrm{BC})^{-1}\mathrm{W}\epsilon_t \tag{24}$$

For the vast majority of the DSGE models one can thus apply the following short algorithm to obtain the solution:

Algorithm 4: Binder-Pesaran DSGE Solution

1. Rewrite the DSGE model into the following form:
   $\mathrm{M}_{00}s_t = \mathrm{M}_{10}s_{t-1} + \mathrm{M}_{01}\mathbb{E}_t s_{t+1} + \mathrm{M}_s\epsilon_t$

2. Compute the matrices $\mathrm{A} = \mathrm{M}_{00}^{-1}\mathrm{M}_{10}$, $\mathrm{B} = \mathrm{M}_{00}^{-1}\mathrm{M}_{01}$ and $\mathrm{W} = \mathrm{M}_{00}^{-1}\mathrm{M}_{s}$
3. Iterate the equation $\mathrm{C} = (\mathrm{I} - \mathrm{BC})^{-1}\mathrm{A}$ with a suitable guess until the matrix C converges.
4. Calculate $\mathrm{D} := (\mathrm{I} - \mathrm{BC})^{-1}\mathrm{W}$ to obtain the solution form:
   $s_t = \mathrm{C}s_{t-1} + \mathrm{D}\epsilon_t$

Hence, by applying this algorithm one obtains the solution to a large class of DSGE models by simple matrix iterations and multiplications which can be differentiated such that the solution method can be implemented into the STAN software package.

4.3 Further Computational Issues

To find a solution to a DSGE model Binder and Pesaran (1997) proposes to iterate the solution to the C matrix using the following rule: $\mathrm{C} = (\mathrm{I} - \mathrm{BC})^{-1}\mathrm{A}$. Although STAN is able to cope with the latter iteration types the building of inverses is computationally one of the most expensive operations, therefore it should be generally avoided. Instead one can directly plug in any initial guess into the equation $\mathrm{C} = \mathrm{BC}^2 + \mathrm{A}$ until it converges.

Although the Binder and Pesaran (1997) algorithm is transparent and easy to implement it has a main drawback. While the solution method proposed by Sims (2002) provides conditions which are necessary and also sufficient to guarantee that the model has a unique stable solution, for the Binder-Pesaran algorithm only a set of sufficient conditions under which the a unique stable solution exists can be derived. In particular, the matrix iteration will also converge if the model has multiple equilibria however these solutions are commonly excluded when DSGE models are estimated. Therefore, to assess whether the model has a unique stable solution, we rely here on the Sims (2002) algorithm. Although STAN is not capable to deal with complex numbers, external functions can be included into the algorithm and also partial derivatives of external functions could be manually specified which are automatically used by STAN when applying the chain rule. Yet, this is not necessary as the Sims (2002) algorithm is used only to reject the sample draw in case the Hamiltonian sampler enters a point in the parameter space where the model has no unique stable solution. For the latter purpose no calculation of the derivative is needed. To implement several matrix decompositions to execute the Sims (2002) algorithm the Intel Math Kernel Library (Intel MKL), a collection of BLAS and LAPACK algorithms which also Matlab uses, is applied and linked into the STAN C++ code.

A further computational issue arises when the covariance matrix Σ is initialized for the Kalman filter. Hamilton (1994) proposes to use Kronecker products to solve for Σ which STAN is able to handle, however it is computationally very costly since the dimension of the problem grows quadratically with the number of equations the model consists of. Since the solution of the DSGE model has to be non-explosive we can obtain Σ again by an iterative procedure. However, as Σ has an impact on the log-likelihood the calculation of this part of the gradient is costly once a large number of iterations is necessary to achive convergence. The initial variance is generally obtained by solving the discrete Lyapunov equation which belongs to the class of Stein matrix equations. Several iterative procedures are proposed in Zhoua et al. (2009) which accalerate the iteration exponentially and enable to calculate parts of the derivative in one step.

Altering the iteration procedure in the Binder-Pesaran algorithm and the adoption of a more efficient calculation to initialize the Kalman filter speeded up the algorithm by a factor of 3-4 for a mid-scale NK-model. In general we can state that calculation of the gradient is costly therefore streamlining the model setup is necessary as far as possible to avoid additional computational burden which increases exponentially with the dimension of the model.

5 Estimation Results

In this section we present the results obtained by applying the HMC algorithm to the textbook small scale New-Keynesian model proposed in Herbst and Schorfheide (2015) and subsequently to the Smets and Wouters (2007) model, a medium scale model serving as the core for a wide range of applied policy models.

5.1 A Small Scale New Keynesian Model

The most basic DSGE model estimated in Herbst and Schorfheide (2015) is a slightly altered version of the standard three equation textbook New Keynesian model (see e.g. Clarida et al. (1999)) consisting of the dynamic IS curve, the New Keynesian Phillllips curve and a Taylor-type monetary policy rule. The Herbst and Schorfheide (2015) version of the model uses quadratic price adjustment instead of the Calvo (1983) scheme and adds also a government sector to the model. Both the technology shock and the government spending shock is $AR(1)$. The model can be described by the following equations:[6]

$$\hat{y}_t = \mathbb{E}_t[\hat{y}_{t+1}] - \frac{1}{\tau}\left(\hat{R}_t - \mathbb{E}_t[\hat{\pi}_{t+1}] - \mathbb{E}_t[\hat{z}_{t+1}]\right) + \hat{g}_t - \mathbb{E}_t[\hat{g}_{t+1}] \tag{25}$$

$$\hat{\pi}_t = \beta\mathbb{E}_t[\hat{\pi}_{t+1}] + \kappa(\hat{y}_t - \hat{g}_t) \tag{26}$$

$$\hat{R}_t = \rho_R \hat{R}_{t-1} + (1-\rho_R)\psi_1\hat{\pi}_t + (1-\rho_R)\psi_2(\hat{y}_t - \hat{g}_t) + \epsilon_{R,t} \tag{27}$$

$$\hat{g}_t = \rho_g \hat{g}_{t-1} + \epsilon_{g,t} \tag{28}$$

$$\hat{z}_t = \rho_g \hat{z}_{t-1} + \epsilon_{z,t} \tag{29}$$

To estimate the model three observables are used: GDP growth, inflation and the nominal interest rate. These are linked to the state equations as follows:

$$YGR_t = \gamma^{(Q)} + 100(\hat{y}_t - \hat{y}_{t-1} + \hat{z}_t) \tag{30}$$

$$INFL_t = \pi^{(A)} + 400\hat{\pi}_t \tag{31}$$

$$INT_t = \pi^{(A)} + 4\gamma^{(Q)} + 400\hat{R}_t \tag{32}$$

In this setup we do not allow for any measurement error. The small scale model thus has 13 structural parameters to be estimated:

$$\theta = [\tau, \kappa, \psi_1, \psi_2, \rho_r, \rho_g, \rho_z, \sigma_r, \sigma_g, \sigma_z, r^A, \pi^A, \gamma^Q] \tag{33}$$

The priors we assume are similar to those used in Herbst and Schorfheide (2015) and are summarized in the table below.

For the estimation we used 10 parallel chains with each 1,000 draws. Due to the

[6] For further details we direct the reader to (Herbst and Schorfheide, 2015, pp.15-28.).

Table 1: Prior Distributions

Name	Domain	Distribution	Parameter 1	Parameter 2
τ	[0,∞)	Gamma	2.00	0.50
κ	[0,1)	Uniform	0.00	1.00
ψ_1	[0,∞)	Gamma	1.50	0.25
ψ_2	[0,∞)	Gamma	0.50	0.25
$r^{(A)}$	[0,∞)	Gamma	0.50	0.50
$\pi^{(A)}$	[0,∞)	Gamma	7.00	2.00
$\gamma^{(Q)}$	(-∞,∞)	Normal	0.40	0.20
ρ_r	[0,1)	Uniform	0.00	1.00
ρ_g	[0,1)	Uniform	0.00	1.00
ρ_z	[0,1)	Uniform	0.00	1.00
$100\sigma_r$	[0,∞)	Inv. Gamma	0.40	4.00
$100\sigma_g$	[0,∞)	Inv. Gamma	1.00	4.00
$100\sigma_z$	[0,∞)	Inv. Gamma	0.50	4.00

Notes: For the Beta, Gamma and Normal distribution Parameter 1 and Parameter 2 stands for the mean and the standard deviation. For the Uniform distribution the parameters define the bounds of the interval. For the Inverse Gamma distribution they correspond to parameters s and ν, where $p_{IG}(\sigma) \propto \sigma^{-\nu-1}e^{-\nu s^2/2\sigma^2}$. See also Herbst and Schorfheide (2015).

efficiency of the HMC we settled for a burn in of 500 draws for each chain, and the diagnostics are confirming that a relatively low number of discarded initial draws is sufficient to ensure that the sampler finds regions of high probability. To visualize the diagnostics of the HMC method we used ShinyStan Version 3.0 (Gabry and Veen, 2020).

Table 2 shows the statistics describing the sampling efficiency of the HMC for each of the structural parameters and the log-posterior as well. Studying the numerical diagnostics of the sampling efficiency two of the main advantages of the HMC algorithm becomes visible: the high effective sample size, and the high accuracy of the simulation of the target density.

Recall, the first is due to the greatly reduced autocorrelation of the draws, introduced by the random variation in the total energy, i.e. by the random variation of the momentum. The latter is granted by the clever usage of the gradient to set the trajectory in the phase space along the Hamiltonian, i.e. the Hamiltonian equations ensures that all draws, after initial convergence, are from the target distribution.

This improvement in efficiency is why we consider HMC revolutionary for DSGE estimation. Herbst and Schorfheide (2015) report the inefficiency factor for the relative risk aversion parameter (τ) for the different Random-Walk Metropolis Hastings algorithms. The inefficiency factor is the inverse of $\mathrm{N}_{eff}/\mathrm{N}$, and note that the Random-Walk Metropolis Hastings suffers from high inefficiency due to its high autocorrelation. To grasp the leap in efficiency we highlight that the naive identity matrix based Metropolis proposal has an inefficiency that translates the "100,000 draws [...] is about as accurate as an approximation obtained from 5.5 *iid* draws" (Herbst and Schorfheide, 2015, p.119.), while the standard, benchmark Random-Walk Metropolis Hastings algorithm described in Chapter 2 has an inefficiency that increases the effective sample size to 1,137,[7] while the 3-Block Random Walk Metropolis Hastings algorithm results in an equivalent of 2,440 *iid* draws. In comparison the effective sample size (N_{eff}) for the 100,000 draws with HMC is 89,737 for the risk aversion parameter (τ). In other words the HMC estimation represents 78.60 efficiency improvement over the standard, 1-Block, Random Walk Metropolis Hastings algorithm and a 36.63 fold over the 3-Block Random Walk Metropolis Hastings algorithm. However, the efficiency gain comes at a cost in terms of computational time, as the gradient has to be evaluated.

Another advantage of weakly autocorrelated draws is the potential to run fully independent shorter chains in parallel, in other words STAN based HMC is highly parallelizable. The evaluation of the gradient and its computation for each transition is an increasingly difficult task in the number of structural parameters. The C++ level integration of the automated differentiation and the computational improvements discussed before renders HMC also for larger models feasible.[8]

Lastly, and probably most importantly, we need to highlight the fact that due to the higher convergence of the draws to the typical set, we can abandon the practice of a mode-estimation before sampling. This potentially also improves the reliability of our estimation method in higher dimensional models considerably, as discussed by Betancourt (2018). We are confident that future research will highlight the advantages of HMC in large DSGE models with irregularly shaped posterior.

The second column of Table 2 reports the ratio of the Monte Carlo standard error of the mean (MCSE) to the posterior standard deviation (SD). The former

[7] Herbst and Schorfheide (2015) report the inefficiency factor of 88 for the 1-Block Random Walk Metropolis Hastings algorithm for the parameter τ. In terms of inefficiency factor the HMC has a 1.12 inefficiency factor.

[8] With the advances of GPU computing in STAN and propagation of higher CPU core counts we anticipate another jump in the computational speed the coming years further advancing the applicability of our solution to estimate DSGE models.

Table 2: Sampling Efficiency of the Hamiltonian Monte Carlo

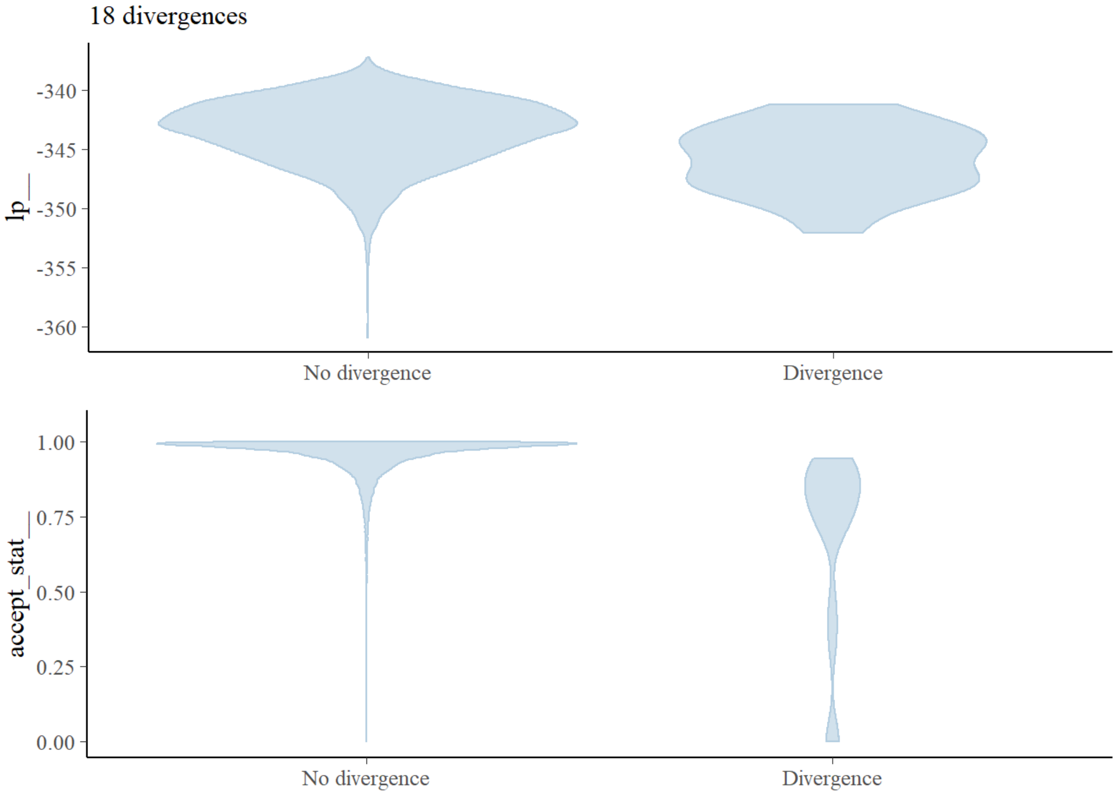
Parameter	N_{eff}/N	MCSE/SD	Parameter	N_{eff}/N	MCSE/SD
τ	89.37 %	1.05%	ρ_r	65.70 %	1.23%
κ	91.06 %	1.05%	ρ_g	94.94 %	1.03%
ψ_1	74.18%	1.16%	ρ_z	56.72%	1.33 %
ψ_2	67.16%	1.22%	$100\sigma_r$	74.72 %	1.16%
$r^{(A)}$	58.19%	1.31%	$100\sigma_g$	94.07 %	1.03%
$\pi^{(A)}$	50.44 %	1.47 %	$100\sigma_z$	89.45 %	1.06 %
$\gamma^{(Q)}$	55.57 %	1.34 %	Log-Posterior	36.24 %	1.66 %

Notes: The table summarizes the efficiency of the HMC sampling. The first column (N_{eff}/N) displays the effective sample size divided by the total number of draws for the structural parameters of the Small Scale DSGE model and its posterior in percentages (%). A higher number indicates more efficient sampling for the respective parameter. The second column (MCSE/SD) contains the ratio of the Monte Carlo standard error of the mean (MCSE) to the posterior standard deviation (SD), again in percentages (%). Here a lower number indicates a more efficient sampling.

is related to the accuracy of the simulation, the smaller the standard error, the loser the estimated parameter value is to the true value. The latter gives the total uncertainty around the structural parameter. The ratio is considered to be small if it is below 5%, thus the values around 1% are indicative of a highly efficient sampling.

Turning to the diagnostics of the sampling, starting with the number and properties of the divergent transitions. In general the existence of divergent transitions is a warning sign that the results might be invalid, however rejected transitions might be also false positive. In other words, if they do not display a common pattern, and are a low proportion, then they can be safely neglected. From 10,000 draws we observed approximately 18 divergent iterations, that is 0.2%. The existence of divergent transitions can indicate invalidity of the results, however the fact that their number is very low in relative terms, and they show no systematic pattern, we argue that the results are to be trusted. Figure 1 plots the frequency of divergent transitions against the log-posterior (*lp__*) in the top panel, and the acceptance of statistic (*accept_stat__*) in the bottom panel.

Figure 1: Small Scale DSGE Diagnostics: Divergence Information



Notes: Plots of the divergent transitions (x-axis) against the log-posterior (y-axis top panel) and against the acceptance statistic (y-axis bottom panel) of the Hamiltonian Monte Carlo sampling algorithm.

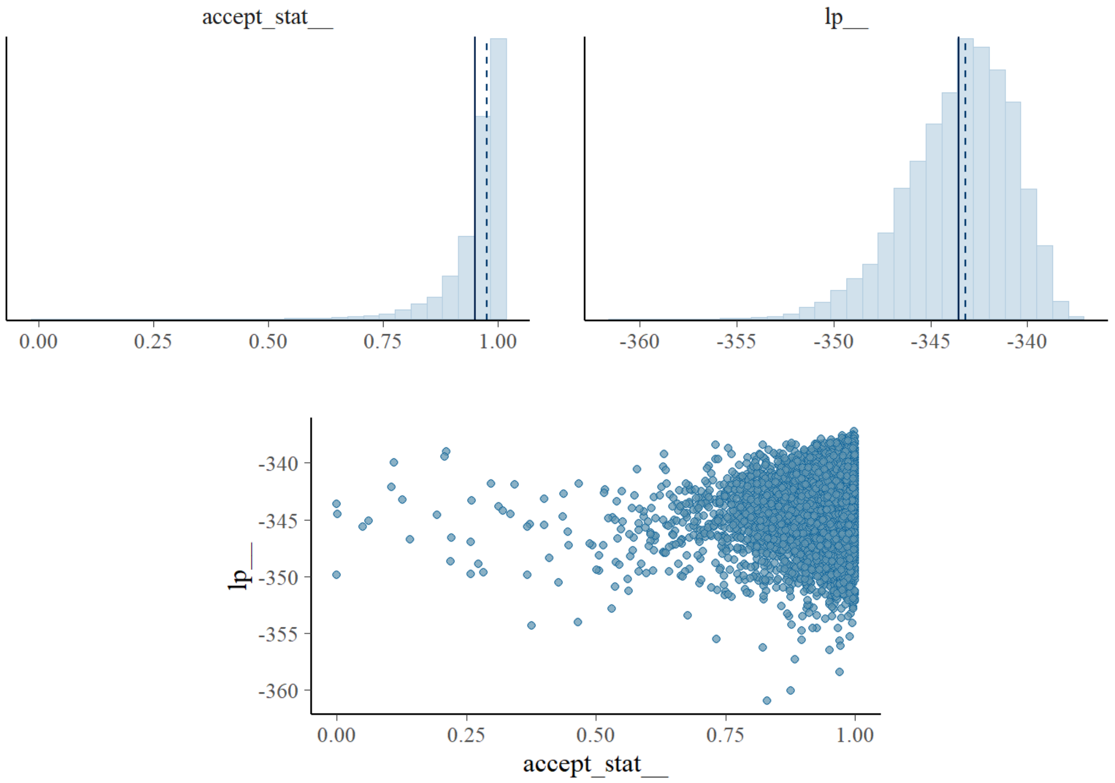
From the top panel we can see directly the log posterior distribution. It is worth noting that the divergent transitions are mostly in the medium probability regions, and not in the high, indicating that any divergence could be a false positive, i.e. divergent due to the numerical instability given the complexity of the entire framework. The location of the divergent transitions can provide information which parts of the target distribution is difficult to sample from, albeit comparing the two charts, we can conclude that the sampler did explore the difficult regions of the posterior. Turning to the bottom panel one might be cautious due to the high acceptance rate[9]. In general the intuition applies for the HMC as well that if the acceptance rate is very high it might be indicative of inefficient sampling[10]. To reject this possibility we plot the marginal posterior distributions and the scatter plot of

[9] The acceptance rate refers to the intermediate Metropolis step in the HMC Algorithm implemented in STAN.

[10] It should be noted that STAN allows to set the target Metropolis acceptance rate with a specific control option that adapts the jump size based on the sampling during the burn in phase.

the acceptance rate and the log-posterior on Figure 2.

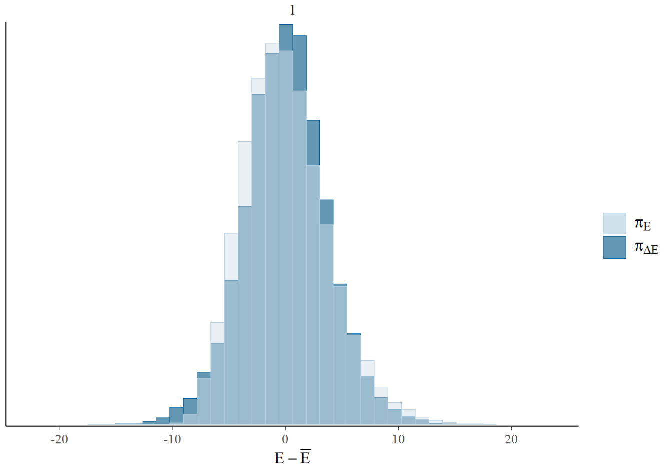
Figure 2: Small Scale DSGE Diagnostics: Acceptance Information



Notes: The figure plots the marginal posterior distribution of acceptance statistic (top leftpanel), marginal posterior distribution of the log-posterior (top right panel), and the scatter plot of acceptance statistic (x-axis bottom panel) against the log-posterior (y-axis bottom panel). The vertical lines indicate the mean (solid line) and median (dashed line). A bad plot would show a relationship between the acceptance statistic and the log-posterior.

The figure shows no relationship of the acceptance rate and the log-posterior, in fact it indicates that the posterior has been adequately explored. This leads us to the discussion of the energy distribution in order to assess robustness of the HMC algorithm, shown on Figure 3. It is desirable that the histograms are ”well-matched: [...] The closer $\pi_{\Delta E}$ is to π_E the faster the random walk explores the energies and the smaller the autocorrelations will be in the chain” (Gabry and Veen, 2020). Figure 3 shows the reason for the low autocorrelation, and thus the high efficiency of the Hamiltonian Monte Carlo algorithm, the energy levels, and with it the posterior-probability levels of the target distribution, are well explored.

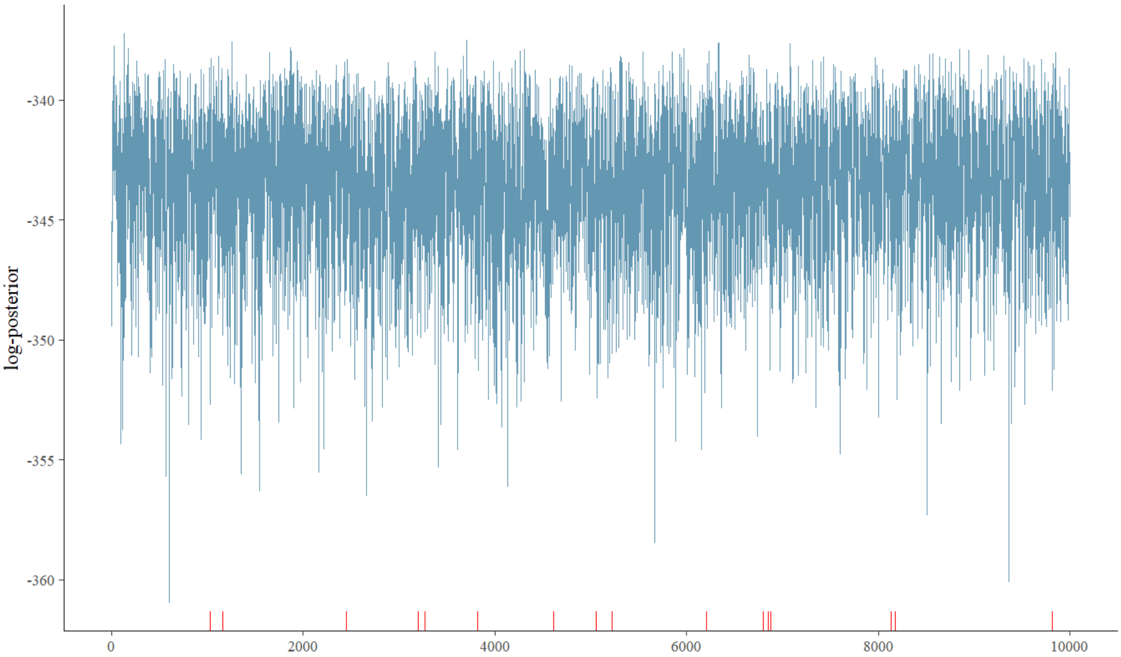
Figure 3: Small Scale DSGE Diagnostics: Energy Distribution



Notes: These are plots of the overlaid histograms of the marginal energy distribution (π_E) and the energy transition distribution ($\pi_{\Delta E}$). See Betancourt (2018) and Carpenter et al. (2017) for more details.

Lastly discussing the trace plot of the log-posterior we can visually inspect the sampling behaviour. Figure 4 shows that the chain explored the different parts of the parameter space. This applies to the other chains and structural parameters as well, all indicating a proper sampling.

Figure 4: Small Scale DSGE Diagnostics: Trace plot



Notes: The log-posterior of the draws from the Hamiltonian Monte Carlo are shown in blue. Divergent transitions are marked on the x-axis with red lines.

Turning to the structural parameter estimates one can verify that the posterior estimates from Hamiltonian Monte Carlo are the same as ones obtained with the Random Walk Metropolis Hastings algorithm.

This verifies the proper functioning of the algorithm, and tells that in small DSGE models with simple target densities, Random Walk Metropolis Hastings sampling works sufficiently well. To explore the properties of Hamiltonian Monte Carlo in a larger model the next section presents the estimation of the Smets-Wouters model.

Table 3: Posterior Estimates of the Small Scale DSGE Model

	Hamiltonian Monte Carlo		Random Walk Metropolis Hastings	
Parameter	Mean	[0.05, 0.95]	Mean	[0.05, 0.95]
τ	2.43	[1.62, 3.35]	2.37	[1.58, 3.82]
κ	0.85	[0.62, 0.99]	0.85	[0.62, 0.98]
ψ_1	1.95	[1.59, 2.34]	1.92	[1.55, 2.20]
ψ_2	0.61	[0.21, 1.15]	0.60	[0.20, 1.21]
$r^{(A)}$	0.42	[0.05, 0.90]	0.44	[0.05, 0.95]
$\pi^{(A)}$	3.41	[2.79, 4.03]	3.38	[2.76, 3.80]
$\gamma^{(Q)}$	0.60	[0.37, 0.83]	0.60	[0.37, 0.74]
ρ_r	0.81	[0.76, 0.85]	0.77	[0.71, 0.82]
ρ_g	0.98	[0.95, 1.00]	0.98	[0.95, 1.00]
ρ_z	0.93	[0.90, 0.97]	0.92	[0.88, 0.92]
$100\sigma_r$	0.19	[0.16, 0.20]	0.22	[0.18, 0.26]
$100\sigma_g$	0.67	[0.59, 0.78]	0.65	[0.57, 0.84]
$100\sigma_z$	0.19	[0.16, 0.23]	0.20	[0.16, 0.36]

Notes: The Table shows the posterior mean and the 5 and 95 percentile of the posterior from the HMC and the RWMH estimation respectively. The results for the HMC are based on $N = 10{,}000$ draws from the posterior, with 10 parallel chains and a burn in of 500 draws for each.
The results for the Random Walk Metropolis Hastings algorithm are based on the authors' replication of the table reported in Herbst and Schorfheide (2015) using the original code available with 100,000 draws. Please note the slight difference in the posterior estimates and the different notation for the scaling of the shock variances compared to Herbst and Schorfheide (2015).
We attribute the former to the inherent random nature of the sampling.

5.2 Smets-Wouters Model

The Smets and Wouters (2007) model is a medium-scale closed economy DSGE model. It became the standard workhorse model for economic policy analysis and served as a basis for newer generations of DSGE models that followed. It is estimated for the US with Random Walk Metropolis Hastings algorithm for the sample of 1960:1–2004:4 using seven key macroeconomic variables: real GDP, real consumption, real investment, the GDP deflator, real wages, employment and the nominal short-term interest rate.[11] The model features a deterministic growth rate driven by labor-augmenting technology progress. The model is subject to nominal and real frictions. The former affecting the labour and goods markets as Calvo-type nominal rigidities similar to Christiano et al. (2005). Both wages and intermediate product markets are subject to partial indexation to lagged inflation. The real frictions manifest themselves as investment adjustment and capital utilization costs. Monetary policy follows a Taylor type rule, with interest rate smoothing and the reaction to inflation- and output gap, the former defined as the deviation from the estimated steady state inflation, the latter as the distance to the flex price economy.

Exogenous variation of the model is driven by seven exogenous shock processes: the standard total factor productivity, monetary policy, investment specific technology, exogenous spending, the model features a risk premium shock and wage and price markup shocks with a MA structure. The latter property introduces anticipated, news shocks for both the regular and the wage Phillips curve. All exogenous shocks are iid-normal with zero mean and estimated variance. The model is log-linearized around the steady state and net of deterministic growth rate. Variables are expressed in terms of percentage deviations from steady state. In order to introduce anticipated news shocks we augment the model with auxiliary state variables, similar to Dynare, so the Binder-Pesaran algorithm can be easily applied.

We estimate the Smets-Wouters model with HMC and present the sampling diagnostics in the Appendix. Once again the efficiency of the HMC algorithm is apparent. Even though we estimate the model with 1000 draws only, it results in an effective sample size of 287 for the log-posterior. We document zero divergent transitions and a well behaved sampling behaviour that explored the target density well. Comparing the posterior results presented in Table 4 and 5 we can conclude that both estimations deliver similar results. The only exception refers to the steady state of inflation, that is estimated to be slightly lower with HMC. This is a result already documented in the forecasting literature, that the Smets-Wouters model has been inappropriate on long run for inflation.

The results obtained and the HMC diagnostics together confirm that the target

[11] Both real consumption and investments are deflated using the GDP deflator. The hours variable is defined as average weekly hours of all persons in the non-farm business sector times total civilian employment.

Table 4: Posterior Estimates of the Smets-Wouters Structural Parameters

	Hamiltonian Monte Carlo		Random Walk Metropolis Hastings	
Parameter	Mean	[0.05, 0.95]	Mean	[0.05, 0.95]
φ	5.88	[4.28, 7.47]	5.93	[4.26, 7.64]
σ_c	1.41	[1.21, 1.65]	1.42	[1.19, 7.64]
h	0.65	[0.65, 0.79]	0.73	[0.66, 0.80]
ξ_w	0.75	[0.66, 0.84]	0.75	[0.66, 0.84]
σ_l	2.09	[1.11, 3.07]	2.06	[1.11, 2.93]
σ_l	2.09	[1.11, 3.07]	2.06	[1.11, 2.93]
ξ_p	0.64	[0.56, 0.73]	0.64	[0.56, 0.73]
ι_w	0.56	[0.35, 0.78]	0.57	[0.37, 0.78]
ι_p	0.24	[0.11, 0.38]	0.23	[0.09, 0.37]
ψ	0.47	[0.30, 0.67]	0.47	[0.30, 0.64]
Φ	1.64	[1.51, 1.76]	1.63	[1.50, 1.76]
r_{pi}	2.05	[1.79, 2.33]	2.05	[1.78, 2.32]
ρ	0.82	[0.77, 0.85]	0.82	[0.78, 0.86]
r_y	0.10	[0.07, 0.14]	0.10	[0.06, 0.14]
r_{dy}	0.21	[0.16, 0.25]	0.21	[0.17, 0.25]
$\bar{\pi}$	0.68	[0.51, 0.86]	0.77	[0.59, 0.94]
$100(\beta^{-1} - 1)$	0.14	[0.07, 0.22]	0.15	[0.06, 0.23]
$\bar{l}$	0.83	[-0.75, 2.37]	0.73	[-1.00, 2.45]
$\bar{\gamma}$	0.46	[0.43, 0.49]	0.47	[0.44, 0.49]
α	0.21	[0.18, 0.24]	0.20	[0.17, 0.23]

Notes: The Table shows the posterior mean and the 5 and 95 percentile of the posterior from the Hamiltonian Monte Carlo and the Random Walk Metropolis Hastings estimation respectively. The results for the Hamiltonian Monte Carlo are based on $N = 1000$ draws from the posterior and a burn in of 500 draws.
The results for the Random Walk Metropolis Hastings algorithm are based on the authors' replication of the model using Johannes Pfeiffer's replication files written in Dynare with an acceptance rate of 30.42%, two chains of 500,000 draws and a burn in of 100,000. Thus the resulting number of draws is 800,000.

density of the Smets-Wouters model is well behaved. Thus the application of the RWMH algorithm is warranted as long as tight priors are assumed. Future research may also use HMC diagnostics to facilitate the selection of appropriate priors.

Table 5: Posterior Estimates of the Smets-Wouters Model's Shock Processes

	Hamiltonian Monte Carlo		Random Walk Metropolis Hastings	
Parameter	Mean	[0.05, 0.95]	Mean	[0.05, 0.95]
σ_a	0.48	[0.43, 0.52]	0.47	[0.42, 0.51]
σ_b	0.24	[0.19, 0.28]	0.23	[0.19, 0.28]
σ_g	0.52	[0.48, 0.57]	0.51	[0.46, 0.56]
σ_I	0.46	[0.39, 0.54]	0.45	[0.37, 0.53]
σ_r	0.24	[0.21, 0.25]	0.23	[0.21, 0.26]
σ_p	0.13	[0.10, 0.16]	0.13	[0.11, 0.16]
σ_w	0.25	[0.22, 0.28]	0.24	[0.21, 0.28]
ρ_a	0.98	[0.97, 0.99]	0.98	[0.97, 0.99]
ρ_b	0.27	[0.11, 0.47]	0.28	[0.10, 0.46]
ρ_g	0.97	[0.96, 0.99]	0.97	[0.96, 0.99]
ρ_I	0.69	[0.60, 0.78]	0.69	[0.60, 0.79]
ρ_r	0.17	[0.07, 0.28]	0.17	[0.06, 0.28]
ρ_p	0.96	[0.93, 0.99]	0.96	[0.92, 0.99]
ρ_w	0.97	[0.94, 0.99]	0.97	[0.95, 0.99]
μ_p	0.80	[0.67, 0.90]	0.80	[0.69, 0.91]
μ_w	0.89	[0.82, 0.94]	0.89	[0.82, 0.95]
μ_w	0.89	[0.82, 0.94]	0.89	[0.82, 0.95]
ρ_{ga}	0.57	[0.44, 0.69]	0.54	[0.41, 0.68]

Notes: The Table shows the posterior mean and the 5 and 95 percentile of the posterior from the Hamiltonian Monte Carlo and the Random Walk Metropolis Hastings estimation respectively. The results for the Hamiltonian Monte Carlo are based on $N = 1000$ draws from the posterior and a burn in of 500.
The results for the Random Walk Metropolis Hastings algorithm are based on the authors' replication of the model using Johannes Pfeiffer's replication files using Dynare with an acceptance rate of 30.42%, two chains of 500,000 draws and a burn in of 100,000. Thus the resulting number of draws is 800,000.

6 Extension: Sequential Hamiltonian Monte Carlo

One of the main disadvantages of the HMC algorithm is that it fails to explore multimodal posterior distributions which is documented in existing literature, see e.g. Shiwei et al. (2014). An interesting experiment which also addresses critics by researchers with respect to the estimation setup of the original Smets-Wouters model was carried out in Herbst and Schorfheide (2014). In particular, in the latter work the authors unrestrict the Bayesian model by using uninformative priors for a number of parameters instead of setting tight priors as in Smets and Wouters (2007). Hence, they allow for the information to obtain a larger weight when estimating the model. Herbst and Schorfheide (2014) reports a bimodal shape of the marginal posterior density for a handful parameters once uninformative priors are applied in which case widely used MCMC based samplers as the RWMH algorithm do not mix properly. Instead, commonly used samplers get stuck in one of the modes, depending on the starting point of the chain. To remedy the issue of multimodality several algorithms have already been proposed in the literature, e.g. Neal (2001), Liu and Chen (1998), Gilks and Berzuini (2002) and Moral et al. (2006) where the latter works mainly combine three different algorithms: importance sampling and resampling, rejection sampling, and Markov chain iterations. Chopin (2004) derives a central limit theorem for a large class of SMC sampling methods. Herbst and Schorfheide (2014) carried out pioneer work by introducing the SMC algorithm to DSGE models to remedy issues with multimodality. The proposed SMC framework in Herbst and Schorfheide (2014) fits also into the scheme described by Chopin (2004) and is in principle a sequential importance sampler. In each step the posterior density $p(Y|\theta)^{\beta_n}p(\theta)$ at stage n, where the likelihood is weighted by $0 \leq \beta_n \leq 1$ $\forall n$, serves as a proposal density for the density to be sampled from at the next stage $p(Y|\theta)^{\beta_{n+1}}p(\theta)$ with $\beta_{n+1} > \beta_n$. This framework is also commonly referred to as likelihood tempering in existing literature. Alternatively one can also carry out data tempering by increasing the number of observations included to calculate the likelihood function at each stage. Without going too deeply into details, at each stage the importance weights for all draws $\{\theta_j^{(n)}\}_{j=1}^J$ at stage n, that is the fraction of the posterior densities at stage $n+1$ and n equaling to $p(Y|\theta^{(n)})^{\beta_{n+1}-\beta_n}$, is calculated and serve as the weights for the importance sampling. The swarm of parameter draws and weights $\{\theta_j^{(n)}, w_j^{(n)}\}_{j=1}^J$ together at each stage are commonly referred to as *particles*. Once the variance of the weights becomes large the draws are resampled using the actual weights and the weights are reset to unity. Finally at each stage the draws are mutated or moved applying a Metropolis-Hastings step which is alternatively also referred to as the 'rejuvenation' step.

A main drawback of using the RWMH sampler to rejuvenate the parameter draws at each stage is again that the MH-proposal θ' is either too often rejected or the distance $||\theta - \theta'||$ between the proposal and the current parameter draw is relatively small. In case one targets an acceptance rate of 25 percent each particle will be updated only at each fourth stage on average. The intuition behind likelihood tempering is also that decreasing β_n, the weight of the likelihood function in the posterior density, reduces the energy barrier between distant separated modes which enables also to commonly applied MCMC samplers to move between modes. However this feature can be only exploited when the step size is large enough. For the reason that the HMC algorithm is capable of proposing updates θ' to the current draws θ which are distant and in theory always accepted, it can also exploit this potential when β_n is relatively small. A key question in this context is how large is the probability that the true parameter vector θ lies in the region of the posterior density surrounding a particular mode. This probability is measured by the volumes under the posterior density around a particular mode $\frac{1}{Z}\int_{\theta \in \Theta_i} p(Y|\theta)p(\theta)d\theta$. A potential issue if using the RWMH algorithm in the rejuvenation step is that particles will tend to get stuck in the same region around the typical set where they started from at stage zero and could potentially bias the estimation. With the number of particles going to infinity this bias will have to disappear even if particles were not rejuvenated at all, when using e.g. annealed importance sampling by Neal (2001), as convergence of these algorithms is warranted. However, with increasing amount of parameters the number of particles necessary will increase also exponentially such that a guided approach to rejuvenate the actual parameter draw might be of an advantage. The Sequential Hamiltonian Monte Carlo algorithm has already been applied by Daviet (2018) to logit discrete choice models and reports better convergence properties than the simple SMC method if a leave-one-out approximation of the observed distribution of the particles is used in the correction step. In our work we will use the SMC framework also used in Herbst and Schorfheide (2014) with both multinomial and stratified resampling. The next algorithm summarizes the main steps:

Algorithm 5: Sequential Hamiltonian Monte Carlo

1. Search for the different modes by starting the HMC algorithm from different parameter settings.

2. Specify a sequence $\{\beta_n\}_{n=0}^{N}$ such that $1 = \beta_N > ... > \beta_{n+1} > \beta_n > ... > \beta_0 \geq 0$

3. Tune the HMC sampler for each target density $p(Y|\theta^{(n)})^{\beta_n} p(\theta^{(n)})$ separately, depending also on the current position a given particle $\theta_j^{(n)}$ to be rejuvenated,

if necessary.

4. Run the SMC algorithm by applying the HMC algorithm to execute the rejuvenation step and use always the pretuned sampler at each stage for the target distribution $p(Y|\theta^{(n)})^{\beta_n} p(\theta^{(n)})$ depending also on the current position of the actual draw $\theta_j^{(n)}$.

This algorithm fits also into the scheme proposed by Chopin (2004), as already pointed out by Daviet (2018). Therefore, under common regularity conditions and assuming that the multinomial resampling is used, almost sure convergence will hold:

$$\frac{1}{J}\sum_{j=1}^{J} h(\theta_j^{(n)}) \overset{a.s.}{\rightarrow} \mathbb{E}_{\tilde{\pi}_n}(h) \tag{34}$$

$$\frac{\sum_{j=1}^{J} w_j^{(n)} h(\theta_j^{(n)})}{\sum_{j=1}^{J} w_j^{(n)}} \overset{a.s.}{\rightarrow} \mathbb{E}_{\pi_t}(h) \tag{35}$$

$$\frac{1}{J}\sum_{j=1}^{J} h(\hat{\theta}_j^{(n)}) \overset{a.s.}{\rightarrow} \mathbb{E}_{\pi_n}(h) \tag{36}$$

where $\tilde{\pi}_n(\cdot) := \int \pi_{n-1}(\theta^{(n-1)}) k^{(n)}(\hat{\theta}^{(n-1)}, \cdot) d\theta^{(n-1)}$ with $k^{(n)}$ being the stochastic kernel density function implied by the HMC algorithm. Furthermore $\pi_n(\theta^{(n)}) = \dfrac{1}{Z_n} p(Y|\theta^{(n)})^{\beta_n} p(\theta^{(n)})$, $w_j^{(n)} \propto \nu_j^{(n)} = \pi_n(\theta_j^{(n-1)})/\tilde{\pi}_n(\theta_j^{(n-1)})$ and $\hat{\theta}_j^{(n)}$ the particle positions after resampling. As HMC leaves π_{n-1} invariant, it follows that $\tilde{\pi}_{(n)} = \pi_{n-1}$, hence $w_j^{(n)} = p(Y|\theta_j^{(n-1)})^{\beta_n - \beta_{n-1}}$.

Furthermore, the limit distribution equals to:

$$J^{1/2}\left\{\frac{1}{J}\sum_{j=1}^{J} h(\hat{\theta}_j^{(n)}) - \mathbb{E}_{\pi_n}(h) \overset{D}{\rightarrow} \mathcal{N}(0, \hat{V}_n(h))\right\} \quad \forall n = 1, ..., N \tag{37}$$

with $\hat{V}_n(h)$ obtained recursively:

$$\tilde{V}_0(h) = \text{Var}_{\tilde{\pi}_{(0)}}(h) \tag{38}$$

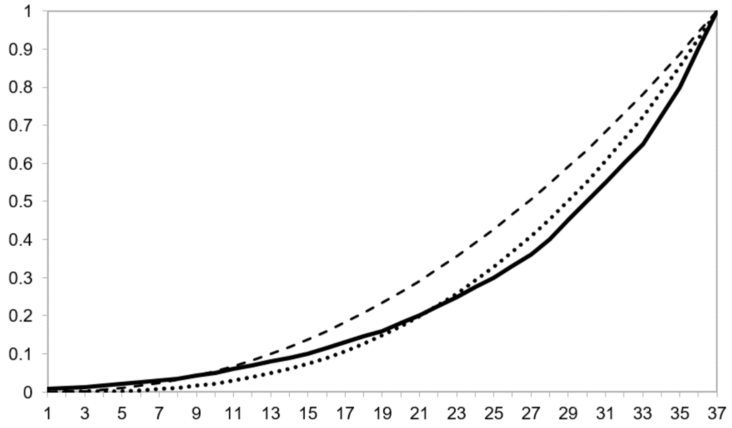
$$\tilde{V}_n(h) = \hat{V}_{n-1}(h)\left\{\mathbb{E}_{k_n}(h)\right\} + \mathbb{E}_{\pi_{n-1}}(h)\text{Var}_{k_n}(h) \quad \forall n = 1, ..., N \tag{39}$$

$$V_n(h) = \tilde{V}_n\left\{\nu_n \cdot (h - \mathbb{E}_{\pi_n}(h))\right\} \quad \forall n = 1, ..., N \tag{40}$$

$$\hat{V}_n(h) = V_n(h) + \text{Var}_{\pi_n}(h) \quad \forall n = 1, ..., N \tag{41}$$

To apply the algorithm we estimate again the Smets and Wouters (2007) model and release the priors in line with Herbst and Schorfheide (2014). We use also the same data set as we used for the estimation of the restricted model. Before executing the estimation code the sampler has to be tuned. In particular, we use $N = 37$ stages and $J = 256$ particles in order not to waste computational resources, which amount is rather low if compared SMC frameworks using RWMH for rejuvenation. The tempering schedule $\{\beta_n\}_{n=1}^{N}$ was calibrated in a way that $p(Y|\theta)^{\beta_n}p(\theta)$ serves always sufficiently well as proposal density for $p(Y|\theta)^{\beta_{n+1}}p(\theta)$, hence the bridge densities are never too different. Even with such a relatively small amount of stages and particles modes are not absorbed highlighting the power of the SHMC estimator in the sense that the rejuvenation step is guided. The following graph displays the shape of the tempering schedule:

Figure 5: Tempering Schedule

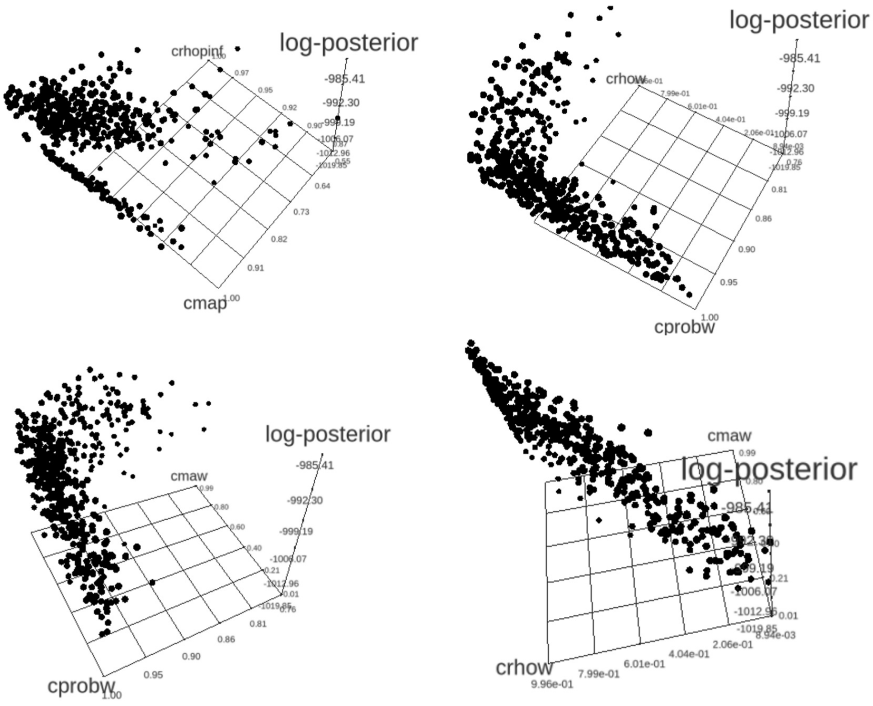


Notes: The solid line shows the tempering schedule used for the estimation. The dashed line shows the tempering schedule if $\beta_n = ((n-1)/(N-1))^\lambda$ with $\lambda = 2.1$ and the dotted line if $\lambda = 2.75$.

The solid line shows the tempering schedule while we also plotted the original schedule used in Herbst and Schorfheide (2014) with $\lambda = 2.1$ (dashed line). At the low end the tempering schedules correspond while after approximately one quarter the schedule used by Herbst and Schorfheide (2014) starts to increase more rapidly. As a comparison we also plotted the schedule from Herbst and Schorfheide (2014) with $\lambda = 2.75$ which provides a better approximation of the schedule used for our estimation framework. Using the HMC sampler there is no need to increase the tempering schedule as rapidly due to the better sampling properties at higher β_n values which allows the particle positions to remain at lower β_n-levels and to mix between the modes for a longer time. However, one should also notice that already at relatively low β_n levels mixing is far away from optimal. As β_n increases less information can be extracted from the density as regards the ratio of the volumes under the modes by moving the particles in the parameter space, yet one can obtain more information with respect to the exact shape of the modes. Another difference if compared with the tempering schedule used by Herbst and Schorfheide (2014) is that while the latter starts with a draw from the prior distribution our initial sampling stems from a slightly informed distribution with $\beta_n = 0.005$, where the HMC sampler is capable to mix between the modes. In our initial sample approximately 30 percent of the particles are from the region around the mode which seem be dominated and to encompass less volume. The latter setup is also in line with our prior beliefs based on existing literature, see also Lanne and Luoto (2018) which augments the SMC algorithm with a non-sequential importance sampling. We also applied the criterion used in Herbst and Schorfheide (2014) to decide whether to resample at a given stage n, yet we resample when the effective sample size (ESS) drops below 0.7 instead of 0.5. In practice, the algorithm resamples in most of the cases at each second stage. Alternatively, we could have also resampled deterministically at each second stage.

We performed the estimation both using multinomial and stratified resampling. Our estimation results suggest that the posterior density for a handful parameters is ill-behaved. In particular, we find in line with Herbst and Schorfheide (2014) and Lanne and Luoto (2018) that the joint kernel density estimates of the parameters ρ_p and μ_p, the ARMA(1,1) terms in the exogenous shock process of the Phillips-curve, is bimodal as illustrated below. In our estimations we obtain that approximately

Figure 6: Joint Posterior Density Estimates



Notes: The plot show the joint posterior densities of the following parameters: [ρ_p,μ_p] (upper left), [ξ_w,ρ_w] (upper right), [ξ_w,μ_w] (lower left) and [ρ_w,μ_w] (lower right). Sample size equals 512, where two sample draws of the size $J = 256$, respectively, were merged, the first obtained by applying multinomial resampling, the second one by stratified resampling. Divergence rate at the last stage $\beta_n = 1$ was approximately 2.3 percent and 1.5 percent, respectively, while the overall divergence rate throughout all $N = 37$ stages amounted to approximately 5.1 percent for both samples.

between 10 and 30 percent of the particles are concentrated in the area around the dominated mode which is higher than the probability of around 5 percent reported in Herbst and Schorfheide (2014). The parameters determining the wage Phillips curve, ξ_w and ρ_w, that is the wage rigidity and the AR(1) term of the mark-up shock process exhibit also a bimodal pattern, yet both modes are rather stretched out in length. The joint kernel density of ξ_w and the MA(1) coefficient of the wage mark-up shock, μ_w, is shaped similarly as the joint density of $[\xi_w, \rho_w]$. The reason for this feature is that ρ_w and μ_w are highly correlated. The joint kernel density exhibits a long ridge along the 45° line which suggests that the mark-up shock process is overparametrized as also suggested by Lanne and Luoto (2018) and restricting the model could result in an improved fit. In general we can conclude that by combining the HMC estimator with SMC we obtain a powerful tool which allows for the estimation of complex and ill-behaved posterior densities and delivers results line with existing literature.

7 Conclusion and Outlook

In this paper we review the benchmark DSGE estimation framework, the RWMH, and present an advanced alternative, the HMC sampler. Subsequently we implement the algorithm for DSGE models in STAN, a state-of-the-art, high-performance software package which has become a workhorse development environment for Bayesian estimation. We estimate a small scale three equation NK textbook model and the Smets-Wouters model using HMC. Our estimation results largely correspond to those from existing literature which underlines the accuracy of the estimation method and the implemented algorithm. In addition we present in detail the sampling diagnostics which enables to conclude that the target densities of the three equation textbook model and the Smets-Wouters model in its original setup exhibit a regular shape. We confirm that in such cases the RWMH algorithm operates adequately. We highlight that the advanced sampling diagnostics for HMC enables to identify parameters which are difficult to sample. In addition a further advantage of HMC is that it does not require any posterior mode search.

We also combine the HMC algorithm with the SMC method to address a shortcoming of the HMC algorithm that it fails to explore ill-behaved posterior densities. We apply this extended framework to estimate the Smets-Wouters model using less informative priors and obtain bimodal posterior densities which results are also inline with those in existing literature.

We are confident that HMC opens new avenues to revisit existing DSGE model estimation exercises in light of the improved sampling properties and the available diagnostics. However, we also acknowledge that further effort is needed to increase the speed of the algorithm.

References

Anderson, G., "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models," *Manuscript, Federal Reserve Board of Governors*, 2000.

Betancourt, M., "A Conceptual Introduction to Hamiltonian Monte Carlo," *arXiv preprint*, 2018, *1701.02434v2.*

Binder, M. and H. Pesaran, "Multivariate linear rational expectations models: characterization of the nature of the solutions and their fully recursive computation," *Econometric Theory*, 1997, *13* (6), 877–888.

Blanchard, O. J. and C. M. Kahn, "The solution of linear difference models under rational expectations," *Econometrica*, 1980, *48* (5), 1305–1312.

Calvo, G. A., "Staggered Prices in a Utility-Maximizing Framework," *Journal of Monetary Economics*, 1983, *12* (3), 383–398.

Carpenter, B., A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabalistic programming language," *Journal of Statistical Software*, 2017, *76* (1).

Chopin, N., "Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Iinference," *Journal of the American Statistical Association*, 2004, *93* (443), 1032–1044.

Christiano, L. J., "Solving dynamic equilibrium models by a methods of undetermined coefficients," *Computational Economics*, 2002, *20* (1-2), 21–55.

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

Clarida, R., J. Galí, and M. Gertler, "The Science of Monetary Policy: A New Keynesian Perspective," *Journal of Economic Literature*, 1999, *37*, 1661–1707.

Creal, D., "Sequential Monte Carlo Samplers for Bayesian DSGE Models," *Manusscript, University Chicago Booth*, 2007.

Daviet, R., "Inference with Hamiltonian Sequential Monte Carlo Simulators," *arXiv*, 2018, *1812.07978v1.*

Duane, S., A.D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters B*, 1987, *195* (2), 216–222.

Durmus, A., É. Moulines, and E. Saksman, "On the convergence of Hamiltonian Monte Carlo," *arXiv preprint*, 2019, *arXiv:1705.00166v2.*

Fernandez-Villaverde, J. and J. Rubio-Ramirez, "Estimating DSGE Models: Recent Advances and Future Challenges," *NBER Working Paper No. 27715*, 2020.

_ , _ , and F. Schorfheide, "Solution and Estimation Methods for DSGE Models," *In: H. Uhlig and J. Taylor (eds.): Handbook of Macroeconomics, Elsevier, New York*, 2016, *2*, 527–724.

Gabry, J. and D. Veen, "ShinyStan Version 3.0.0," *mc-stan.org*, 2020.

Gilks, W. R. and C. Berzuini, "Following a moving target - Monte Carlo inference for dynamic Bayesian Models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2002, *63* (7), 127–146.

Goodrich, Ben and Carlos Montes-Galdon, "Estimating DSGE Models with Stan," *https://pdfs.semanticscholar.org/0a9b/35f9f11ce0c489a911622d700d2a4e64f385.pdf*, retrieved on 25 August, 2020.

Hamilton, J. D., *Time Series Analysis*, Princeton, New Jersey: Princeton University Press, 1994.

Herbst, E. P. and F. Schorfheide, "Sequential Monte Carlo Sampling for DSGE Models," *Journal of Applied Econometrics*, 2014, *27* (7), 1073–1098.

_ and _ , *Bayesian Estimation of DSGE Models*, Princeton, New Jersey: Princeton University Press, 2015.

Hoffman, M. D. and A. Gelman, "The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo," *The Journal of Machine Learning Research*, 2014, *15* (1), 1593–1623.

Kim, J., "Constructing and estimating a realistic optimizing model of monetary policy," *Journalof Monetary Economics*, 2000, *45* (2), 329–359.

King, R. G. and M. W. Watson, "The solution of singluar linear difference systems under rational expectations," *International Economic Review*, 1998, *39* (4), 1015–1026.

Kydland and Prescott, "Time to Build and Aggregate Fluctuations," *Econometrica*, 1982, *50* (6), 1345–1370.

Lanne, M. and J. Luoto, "Data-Driven Identification Constraints for DSGE Models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2018, *80* (2), 236–258.

Liu, J. S. and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, 1998, *93* (443), 1032–1044.

Livingstone, S., M. Betancourt, S. Byrne, and M. Girolami, "On the geometric ergodicity of Hamiltonian Monte Carlo," *arXiv preprint*, 2018, *arXiv:1601.08057.*

Mackenzie, P. B., "An improved hybrid Monte Carlo method," *Physics Letters B*, 1989, *226*, 369–371.

Moral, P. Del, A. Doucet, and A. Jasra, "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006, *68* (3), 411–436.

Neal, R. M., "Annealed importance sampling," *Statistics and Computing*, 2001, *11*, 125–139.

_ , "MCMC Using Hamiltonian Dynamics," in G. L. Jones S. Brooks, A. Gelman and eds. X.-L. Meng, eds., *Handbook of Markov Chain Monte Carlo*, New York: CRC Press, 2011, pp. 30–61.

Otrok, C., "On measuring the welfare cost of business cycles," *Journal of Monetary Economics*, 2001, *47* (1), 61–92.

Roberts, G.O., A. Gelman, and W.R. Gilks, "Weak convergence and optimal scaling of random walk Metropolis algorithms," *Annals of Applied Probability*, 1997, *7*, 110–120.

Schorfheide, F., "Loss function-based evaluation of DSGE models," *Journal of Applied Econometrics*, 2000, *15* (6), 645–670.

Shiwei, L., J. Streets, and B. Shahbaba, "Wormhole Hamiltonian Monte Carlo," *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 1953–1959.

Sims, C. A., "Solving linear rational expectations models," *Computational Economics*, 2002, *20* (1-2), 1–20.

Smets, F. and R. Wouters, "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 2007, *97* (3), 586–606.

Uhlig, H., "A toolkit for analyzing non-linear dynamic stochastic models easily," in R. Marimón and eds. A. Scott, eds., *Computational Methods for the Study of Dynamic Economies.*, Oxford, UK: Oxford University Press, 1999, pp. 30–61.

Zhoua, B., J. Lamb, and G.-R. Duana, "On Smith-type iterative algorithms for the Stein matrix equation," *Applied Mathematics Letters*, 2009, *22*, 1038–1044.

Appendix

Smets-Wouters Model: Hamiltonian Monte Carlo Estimation Diagnostics

Warnings

```
[1] "None of the 1000 iterations ended with a divergent transition."
```

Numerical diagnostics

```
n_eff Rhat     mean se_mean   sd
log-posterior  286.97 1.00 -1191.64    0.27 4.63
crpi           541.79 1.01     2.05    0.01 0.16
crdy          1209.11 1.00     0.21    0.00 0.03
cry            547.47 1.01     0.10    0.00 0.02
crr            471.66 1.01     0.82    0.00 0.02
constelab     1044.53 1.00     0.83    0.03 0.96
constepinf    1039.36 1.00     0.68    0.00 0.10
ctrend         442.39 1.00     0.46    0.00 0.02
constebeta     955.50 1.00     0.14    0.00 0.05
cgy            976.79 1.00     0.57    0.00 0.07
cmaw           625.70 1.01     0.89    0.00 0.04
cmap           358.32 1.00     0.80    0.00 0.07
calfa          716.24 1.00     0.21    0.00 0.02
czcap          831.01 1.00     0.47    0.00 0.11
csadjcost      865.28 1.00     5.88    0.03 0.97
csigma         554.11 1.00     1.41    0.01 0.13
chabb          486.61 1.00     0.73    0.00 0.04
cfc            831.22 1.00     1.64    0.00 0.08
cindw         1003.05 1.00     0.56    0.00 0.13
```

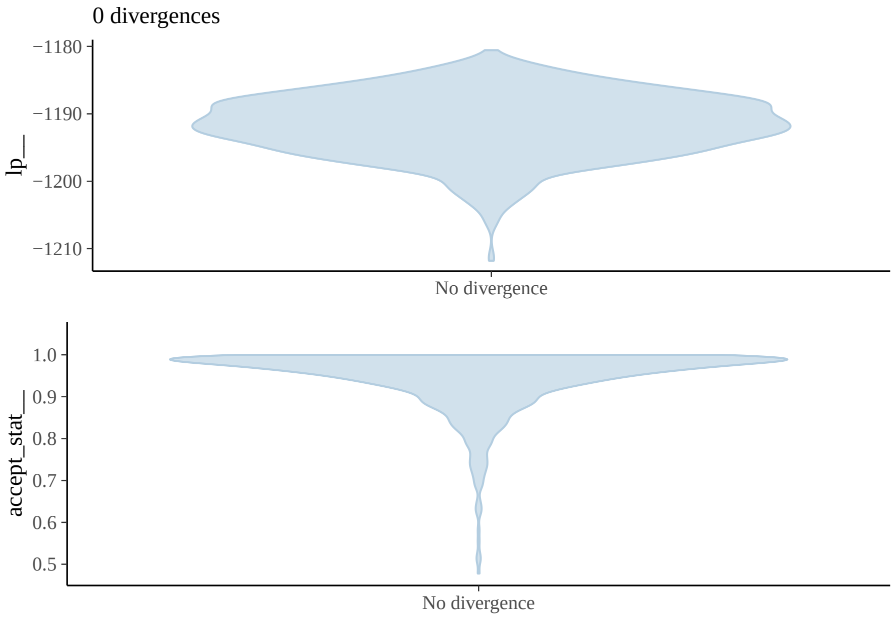
```
cprobw         588.38 1.00      0.75    0.00 0.05
cindp          736.73 1.00      0.24    0.00 0.08
cprobp         740.00 1.00      0.64    0.00 0.05
csigl          803.21 1.00      2.09    0.02 0.59
crhoa          950.71 1.00      0.98    0.00 0.01
crhob          476.22 1.00      0.27    0.01 0.11
crhog          573.45 1.00      0.97    0.00 0.01
crhoqs         746.01 1.00      0.69    0.00 0.05
crhoms         793.29 1.01      0.17    0.00 0.06
crhopinf       555.60 1.00      0.96    0.00 0.02
crhow          615.86 1.01      0.97    0.00 0.01
sigmaea        815.18 1.00      0.48    0.00 0.03
sigmaeb        533.36 1.00      0.24    0.00 0.03
sigmaeg       1355.67 1.00      0.52    0.00 0.03
sigmaeqs       859.94 1.00      0.46    0.00 0.04
sigmaem        882.47 1.01      0.23    0.00 0.01
sigmaepinf     518.16 1.00      0.13    0.00 0.02
sigmaew        865.71 1.00      0.25    0.00 0.02
ctou             0.50 1.00      0.03    0.00 0.00
cg               0.50 1.00      0.18    0.00 0.00
curvp             NaN  NaN     10.00     NaN 0.00
curvw             NaN  NaN     10.00     NaN 0.00
clandaw           NaN  NaN      1.50     NaN 0.00
LL             506.40 1.00   -961.42    0.25 5.54
cpie          1038.58 1.00      1.01    0.00 0.00
cgamma         441.30 1.00      1.00    0.00 0.00
cbeta          955.70 1.00      1.00    0.00 0.00
clandap        831.22 1.00      1.64    0.00 0.08
cbetabar       785.52 1.00      0.99    0.00 0.00
cr             938.46 1.00      1.01    0.00 0.00
crk            785.35 1.00      0.03    0.00 0.00
cw            1145.77 1.00      0.69    0.00 0.04
```

```
cikbar         442.35 1.00     0.03    0.00 0.00
cik            442.38 1.00     0.03    0.00 0.00
clk            981.86 1.00     0.19    0.00 0.03
cky            783.99 1.00     6.28    0.02 0.55
ciy            753.17 1.00     0.19    0.00 0.02
ccy            753.17 1.00     0.63    0.00 0.02
crkky          716.24 1.00     0.21    0.00 0.02
cwhlc          924.85 1.00     0.83    0.00 0.01
cwly           716.24 1.00     0.79    0.00 0.02
conster        938.96 1.00     1.48    0.00 0.13
```

Visual diagnostics

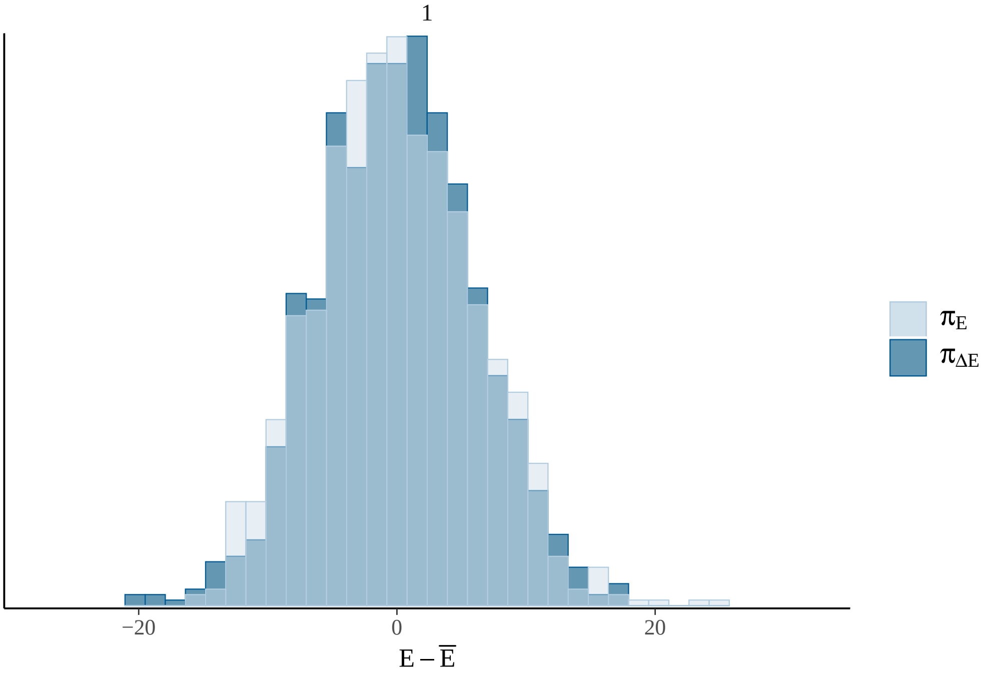
Divergence Information

These are plots of the *divergent transition status* (x-axis) against the *log-posterior* (y-axis top panel) and against the *acceptance statistic* (y-axis bottom panel) of the sampling algorithm for all chains. Divergent transitions can indicate problems for the validity of the results. A good plot would show no divergent transitions. If the divergent transitions show the same pattern as the non divergent transitions, this could indicate that the divergent transitions are false positives. A bad plot would shows systematic differences between the divergent transitions and non-divergent transitions. For more information see https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup.



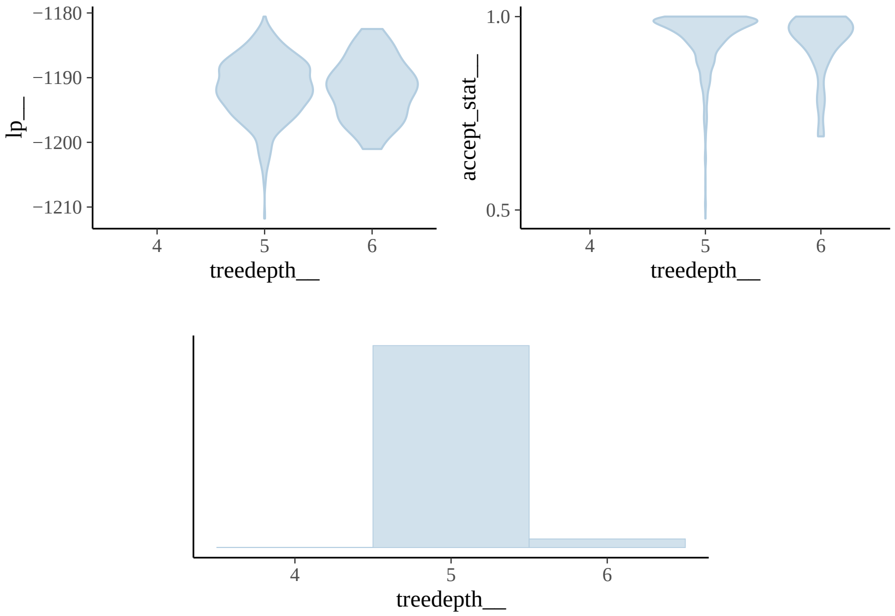
Energy

These are plots of the overlaid histograms of the marginal energy distribution (π_E) and the energy transition distribution ($\pi_{\Delta E}$) for all chains. A good plot shows histograms that look well-matched indicating that the Hamiltonian Monte Carlo should perform robustly. The closer $\pi_{\Delta E}$ is to π_E the faster the random walk explores the energies and the smaller the autocorrelations will be in the chain. If $\pi_{\Delta E}$ is narrower than π_E the random walk is less effective and autocorrelations will be larger. Additionally the chain may not be able to completely explore the tails of the target distribution. See Betancourt 'A conceptual introduction to Hamiltonian Monte Carlo' and Betancourt 'Diagnosing suboptimal cotangent disintegrations in Hamiltonian Monte Carlo' for the general theory behind the energy plots.



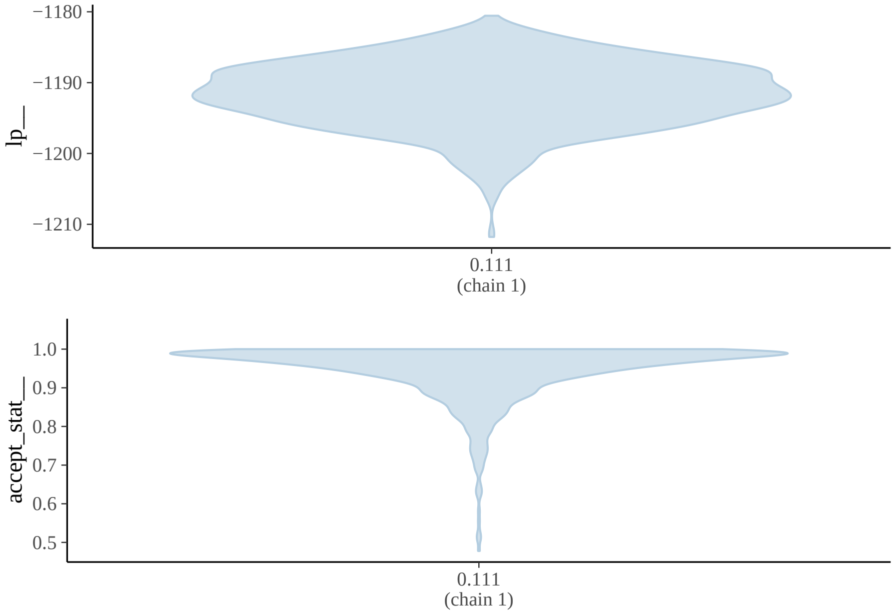
Treedepth Information

These are plots of the *treedepth* (x-axis) against the *log-posterior* (y-axis top left panel) and against the *acceptance statistic* (y-axis top right panel) of the sampling algorithm for all chains. In these plots information is given concerning the efficiency of the sampling algorithm. Zero treedepth can indicate extreme curvature and poorly-chosen step size. Treedepth equal to the maximum treedepth might be a sign of poor adaptation or of a difficult posterior from which to sample. The former can be resolved by increasing the warmup time, the latter might be mitigated by reparametrization. For more information see `https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded` or https://mc-stan.org/docs/reference-manual/hmc-algorithm-parameters.html.



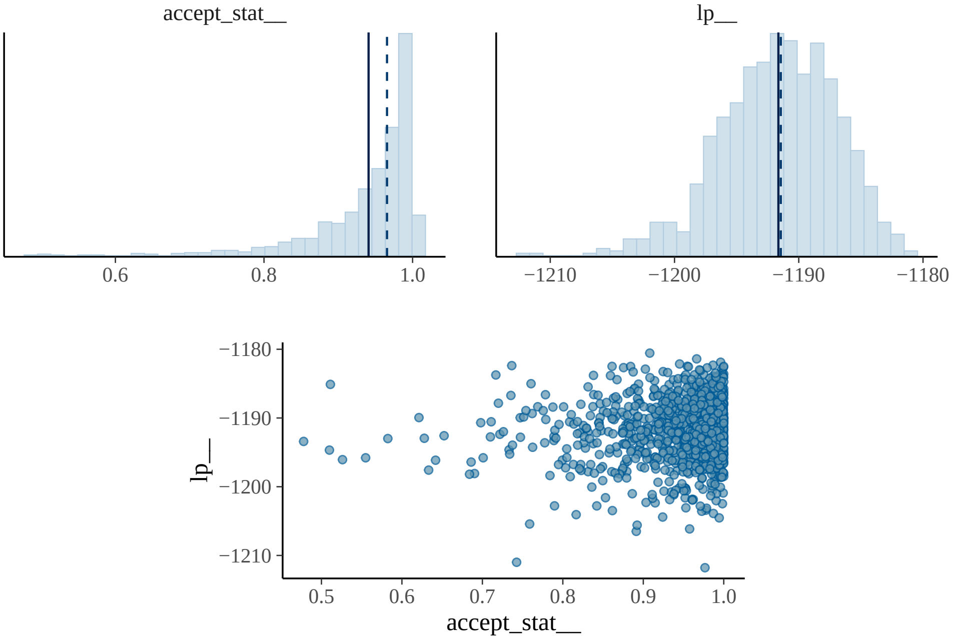
Step Size Information

These are plots of the *integrator step size per chain* (x-axis) against the *log-posterior* (y-axis top panel) and against the *acceptance statistic* (y-axis bottom panel) of the sampling algorithm. If the step size is too large, the integrator will be inaccurate and too many proposals will be rejected. If the step size is too small, the many small steps lead to long simulation times per interval. Thus the goal is to balance the acceptance rate between these extremes. Good plots will show full exploration of the log-posterior and moderate to high acceptance rates for all chains and step sizes. Bad plots might show incomplete exploration of the log-posterior and lower acceptance rates for larger step sizes.



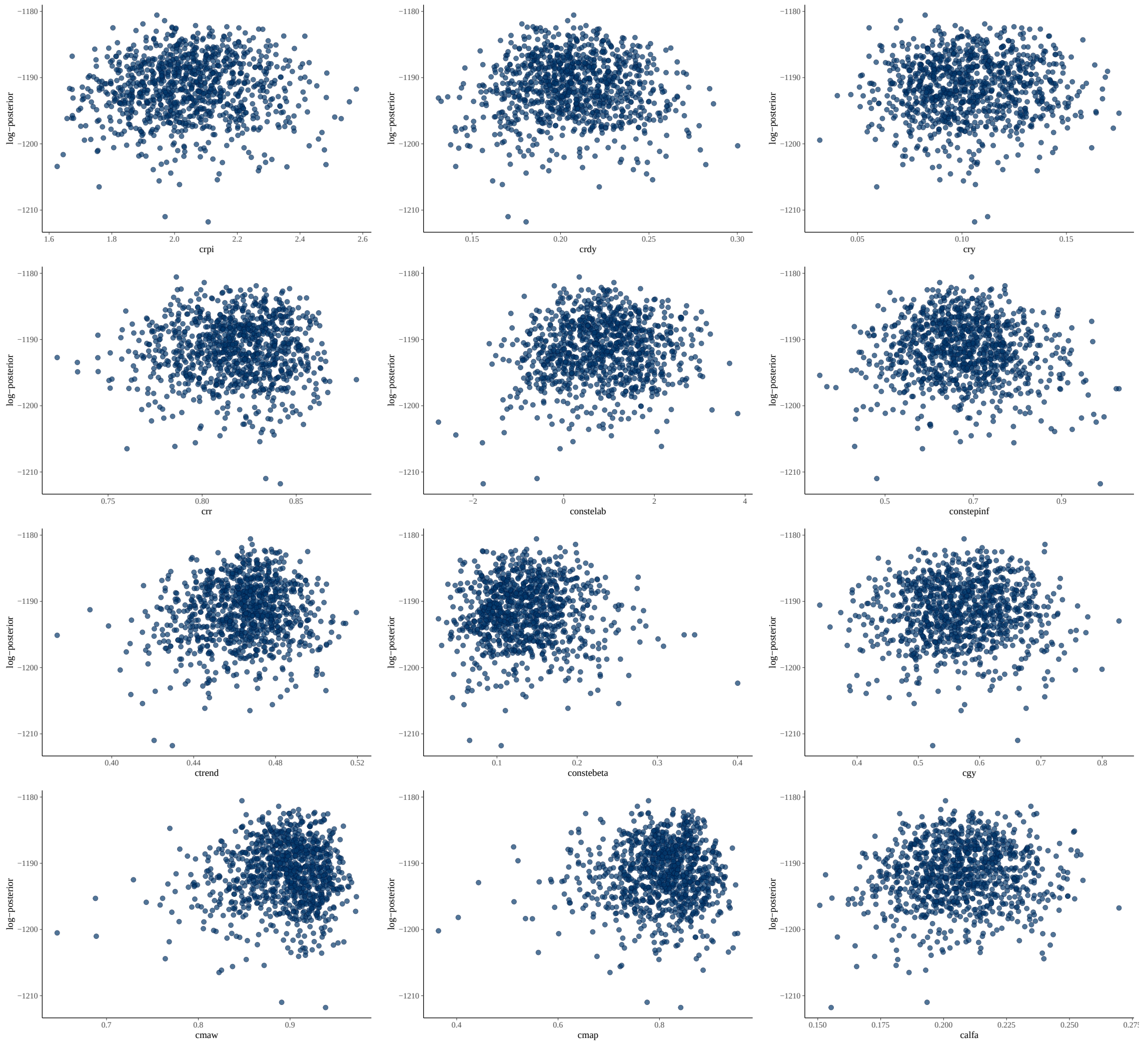
Acceptance Information

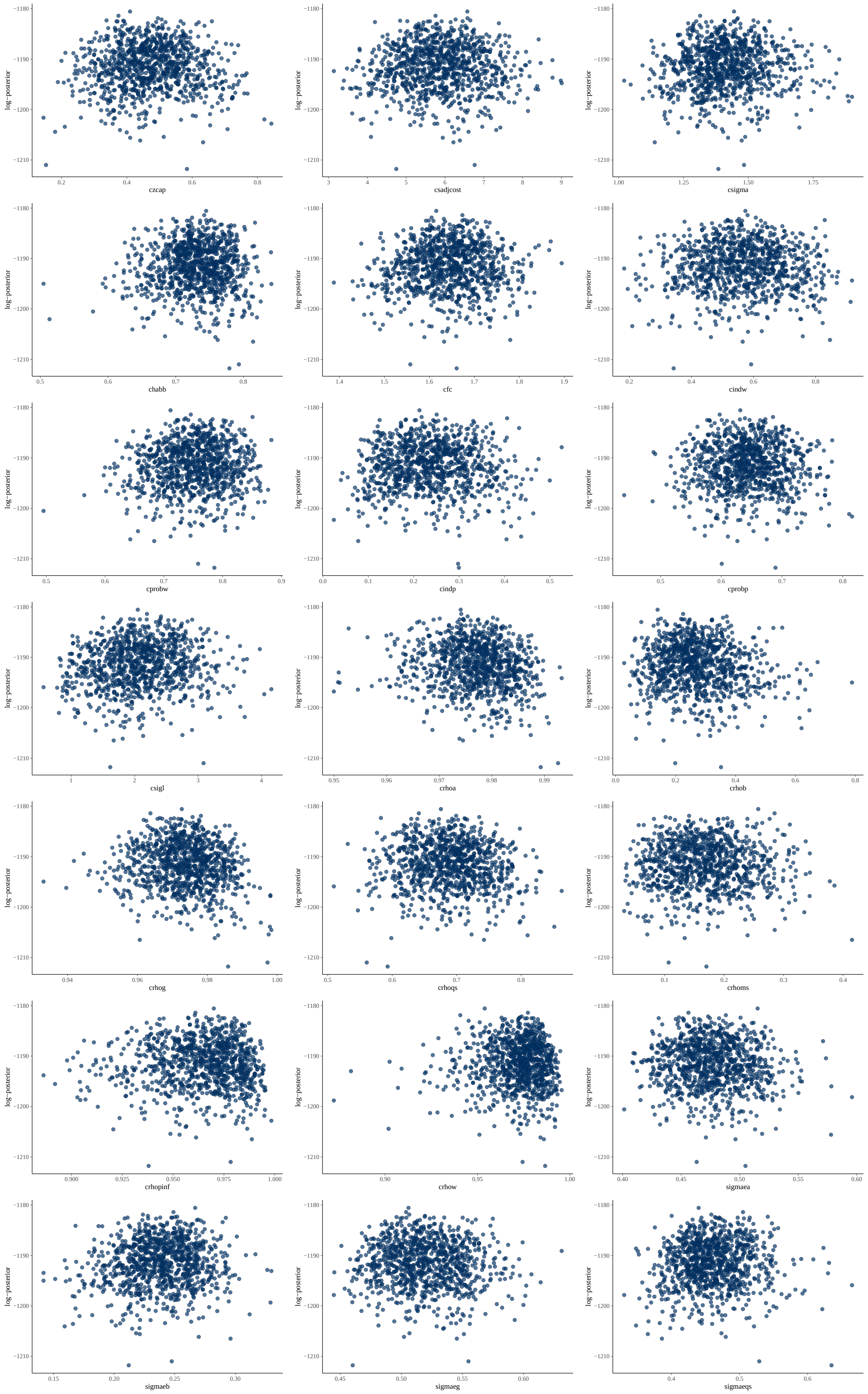
These are plots of the *acceptance statistic* (top leftpanel), the *log-posterior* (top right panel), and, the *acceptance statistic* (x-axis bottom panel) against the *log-posterior* (y-axis bottom panel) for all chains. The vertical lines indicate the mean (solid line) and median (dashed line). A bad plot would show a relationship between the acceptance statistic and the log-posterior. This might be indicative of poor exploration of parts of the posterior which might be might be mitigated by reparametrization or adaptation of the step size. If many proposals are rejected the integrator step size might be too large and the posterior might not be fully explored. If the acceptance rate is very high this might be indicative of inefficient sampling. The target Metropolis acceptance rate can be set with the `adapt_delta` control option. For more information see https://mc-stan.org/docs/reference-manual/hmc-algorithm-parameters.html.

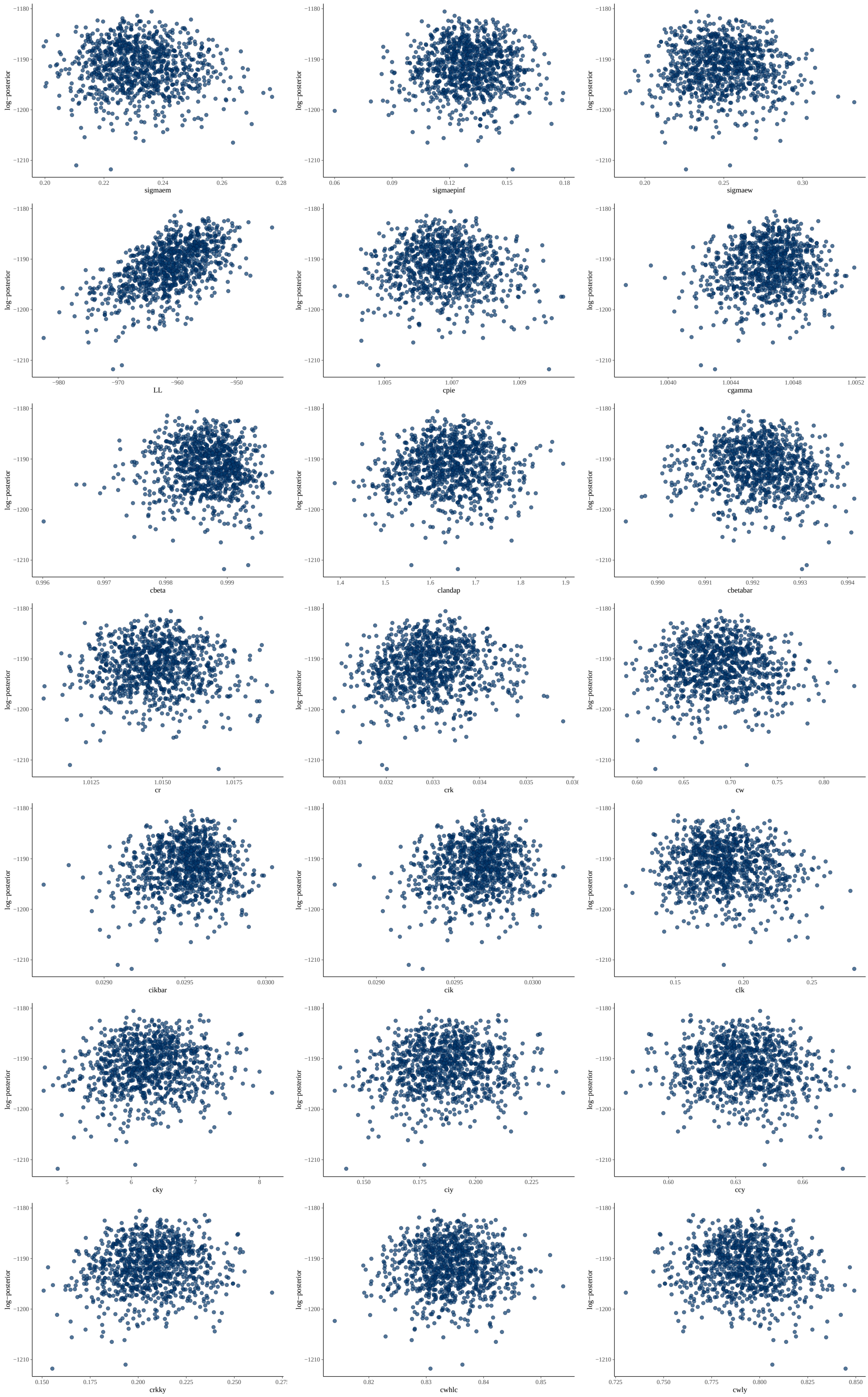


Scatter plots

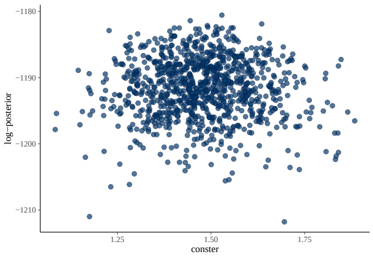
These are scatter plots of crpi, crdy, cry, crr, constelab, constepinf, ctrend, constebeta, cgy, cmaw, cmap, calfa, czcap, csadjcost, csigma, chabb, cfc, cindw, cprobw, cindp, cprobp, csigl, crhoa, crhob, crhog, crhoqs, crhoms, crhopinf, crhow, sigmaea, sigmaeb, sigmaeg, sigmaeqs, sigmaem, sigmaepinf, sigmaew, ctou, cg, curvp, curvw, clandaw, LL, cpie, cgamma, cbeta, clandap, cbetabar, cr, crk, cw, cikbar, cik, clk, cky, ciy, ccy, crkky, cwhlc, cwly, conster, Iter, plotted against `log-posterior`. The red dots, if present, indicate divergent transitions. Divergent transitions can indicate problems for the validity of the results. A good plot would show no divergent transitions. A bad plot would show divergent transitions in a systematic pattern. For more information see https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup.




log−posterior
−1180
−1190
−1200
−1210
czcap 0.2 0.4 0.6 0.8
csadjcost 3 4 5 6 7 8 9
csigma 1.00 1.25 1.50 1.75
chabb 0.5 0.6 0.7 0.8
cfc 1.4 1.5 1.6 1.7 1.8 1.9
cindw 0.2 0.4 0.6 0.8
cprobw 0.5 0.6 0.7 0.8 0.9
cindp 0.0 0.1 0.2 0.3 0.4 0.5
cprobp 0.5 0.6 0.7 0.8
csigl 1 2 3 4
crhoa 0.95 0.96 0.97 0.98 0.99
crhob 0.0 0.2 0.4 0.6 0.8
crhog 0.94 0.96 0.98 1.00
crhoqs 0.5 0.6 0.7 0.8
crhoms 0.1 0.2 0.3 0.4
crhopinf 0.900 0.925 0.950 0.975 1.000
crhow 0.90 0.95 1.00
sigmaea 0.40 0.45 0.50 0.55 0.60
sigmaeb 0.15 0.20 0.25 0.30
sigmaeg 0.45 0.50 0.55 0.60
sigmaeqs 0.4 0.5 0.6



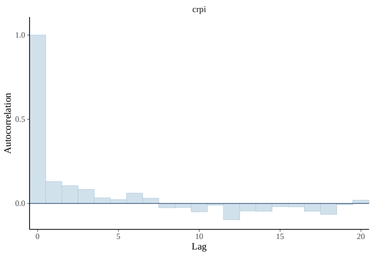
log−posterior
−1180
−1190
−1200
−1210
0.20
0.22
0.24
0.26
0.28
sigmaem
0.06
0.09
0.12
0.15
0.18
sigmaepinf
0.20
0.25
0.30
sigmaew
−980
−970
−960
−950
LL
1.005
1.007
1.009
cpie
1.0040
1.0044
1.0048
1.0052
cgamma
0.996
0.997
0.998
0.999
cbeta
1.4
1.5
1.6
1.7
1.8
1.9
clandap
0.990
0.991
0.992
0.993
0.994
cbetabar
1.0125
1.0150
1.0175
cr
0.031
0.032
0.033
0.034
0.035
0.03€
crk
0.60
0.65
0.70
0.75
0.80
cw
0.0290
0.0295
0.0300
cikbar
0.0290
0.0295
0.0300
cik
0.15
0.20
0.25
clk
5
6
7
8
cky
0.150
0.175
0.200
0.225
ciy
0.60
0.63
0.66
ccy
0.150
0.175
0.200
0.225
0.250
0.27£
crkky
0.82
0.83
0.84
0.85
cwhlc
0.725
0.750
0.775
0.800
0.825
0.850
cwly

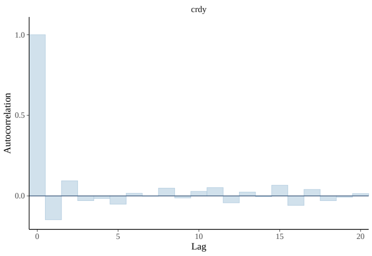


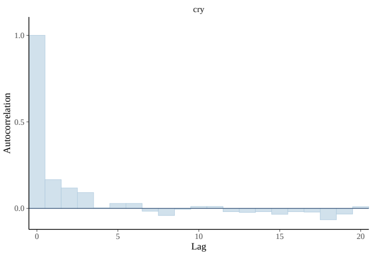
−1180
−1190
−1200
−1210
log−posterior
1.25
1.50
1.75
conster

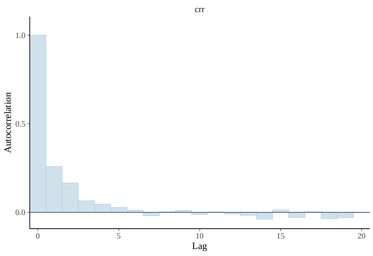
Autocorrelation

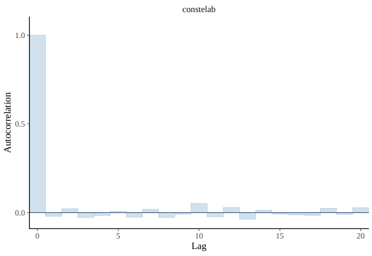
These are autocorrelation plots of crpi, crdy, cry, crr, constelab, constepinf, ctrend, constebeta, cgy, cmaw, cmap, calfa, czcap, csadjcost, csigma, chabb, cfc, cindw, cprobw, cindp, cprobp, csigl, crhoa, crhob, crhog, crhoqs, crhoms, crhopinf, crhow, sigmaea, sigmaeb, sigmaeg, sigmaeqs, sigmaem, sigmaepinf, sigmaew, ctou, cg, curvp, curvw, clandaw, LL, cpie, cgamma, cbeta, clandap, cbetabar, cr, crk, cw, cikbar, cik, clk, cky, ciy, ccy, crkky, cwhlc, cwly, conster, Iter. The autocorrelation expresses the dependence between the samples of a Monte Carlo simulation. With higher dependence between the draws, more samples are needed to obtain the same effective sample size. High autocorrelation can sometimes be remedied by reparametrization of the model.

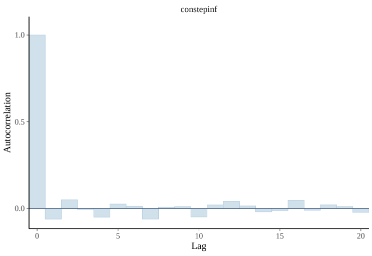


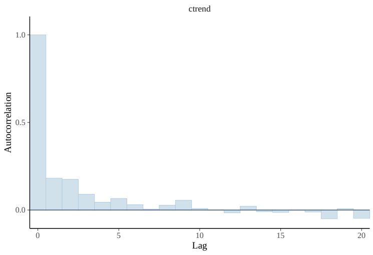


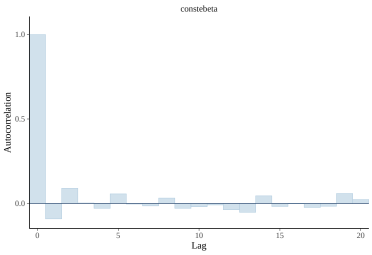


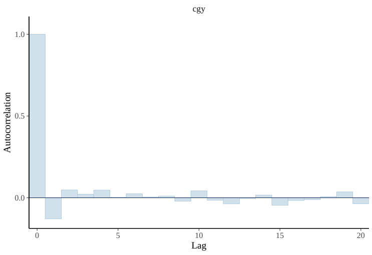


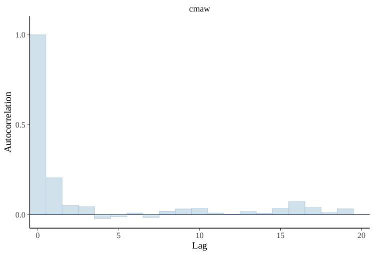


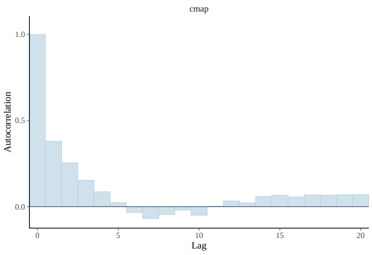


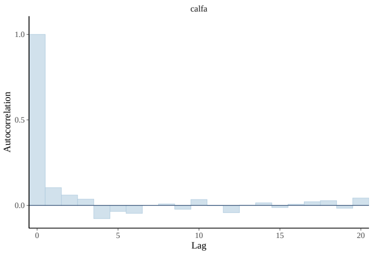


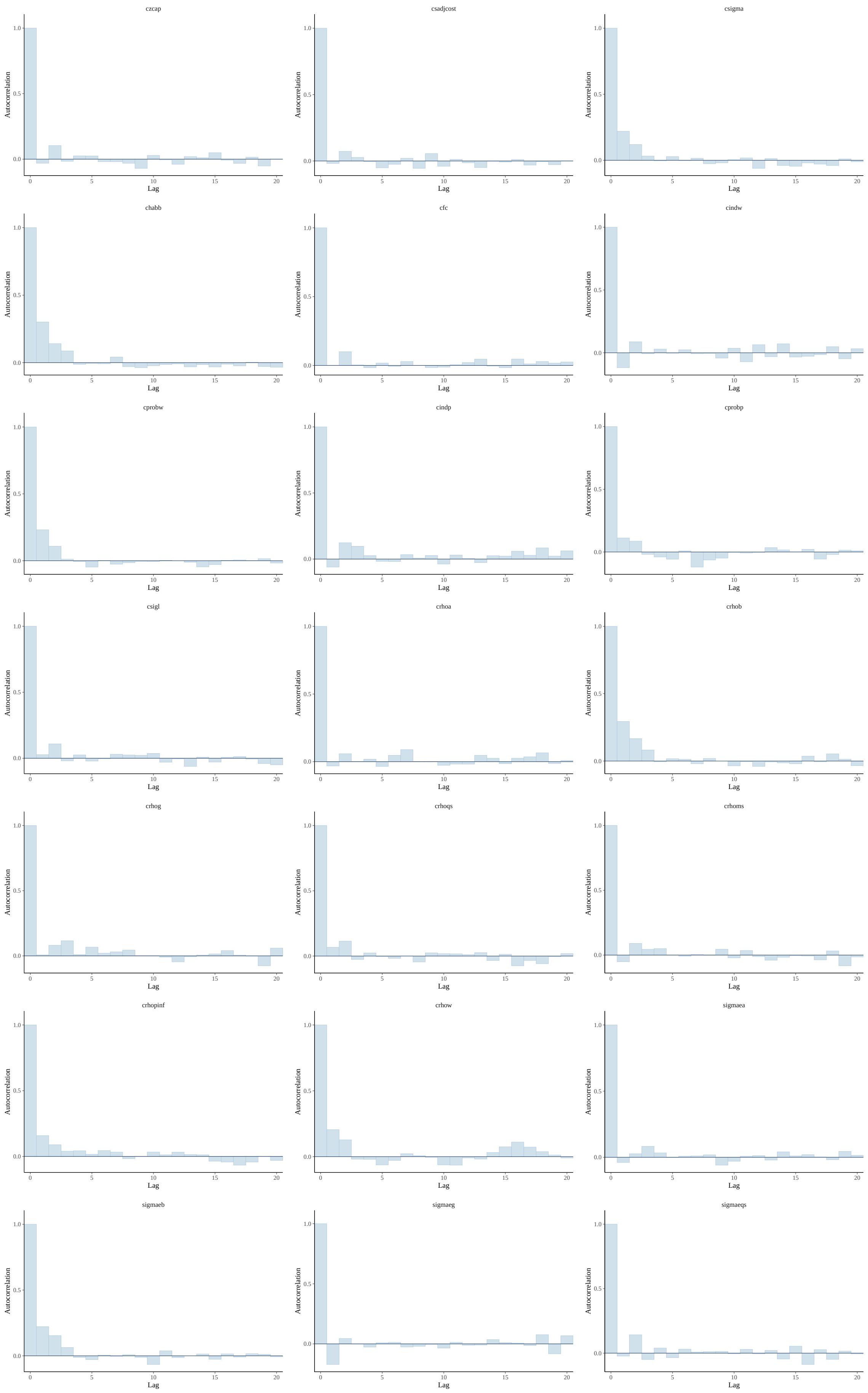


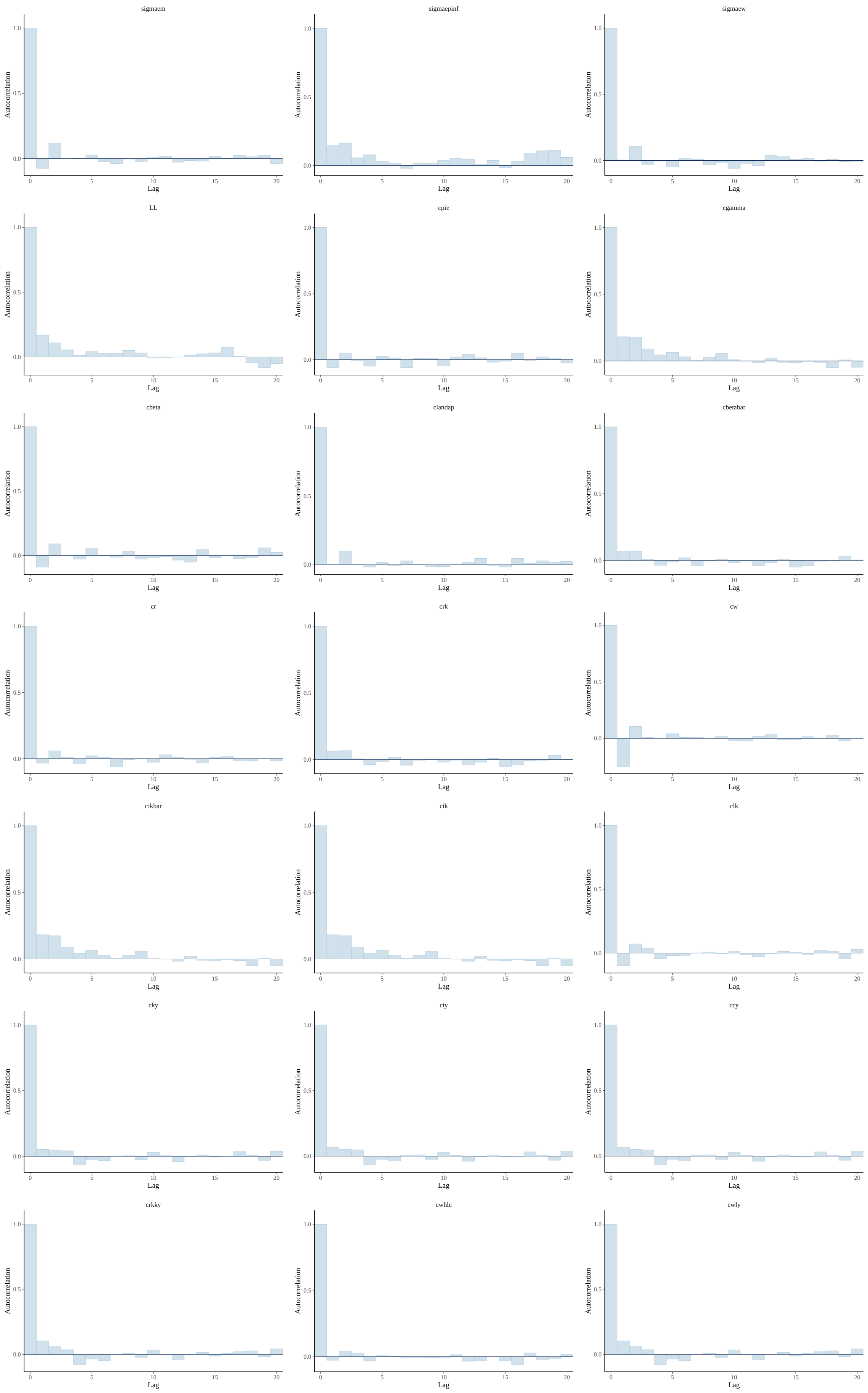


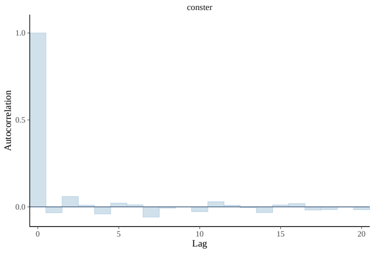






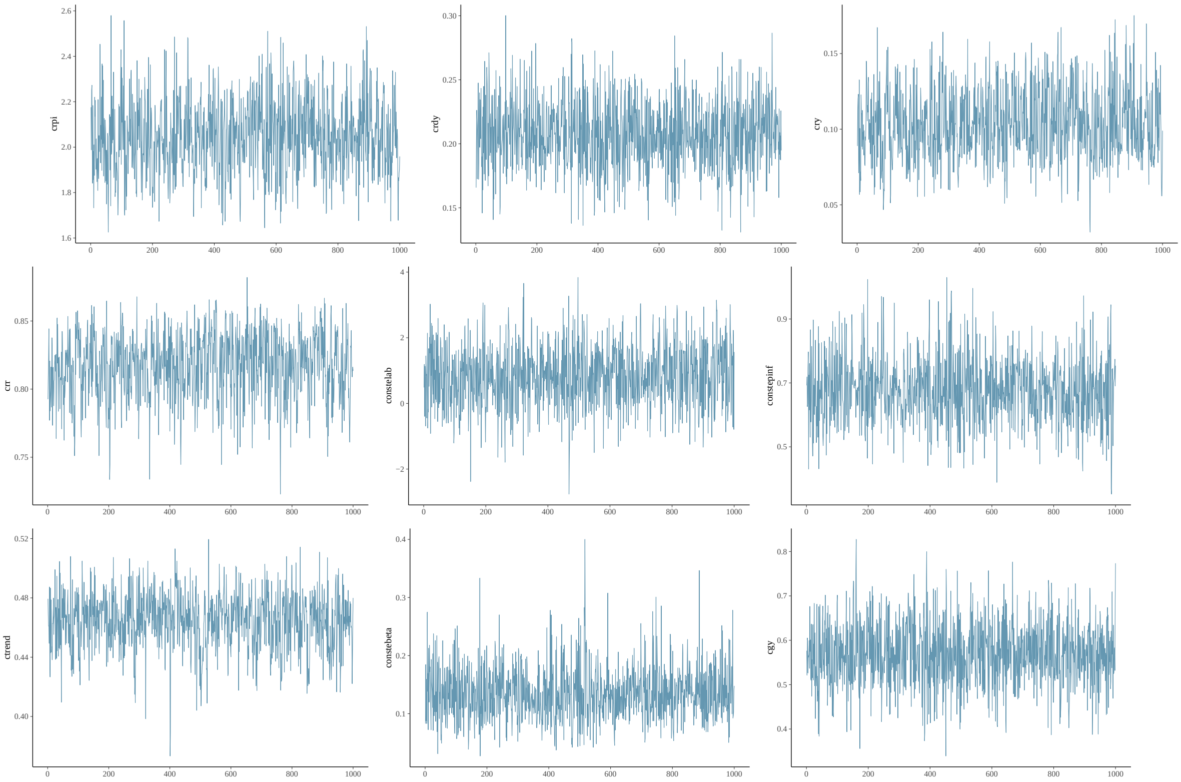

czcap
csadjcost
csigma
chabb
cfc
cindw
cprobw
cindp
cprobp
csigl
crhoa
crhob
crhog
crhoqs
crhoms
crhopinf
crhow
sigmaea
sigmaeb
sigmaeg
sigmaeqs
Autocorrelation
1.0
0.5
0.0
Lag
0
5
10
15
20

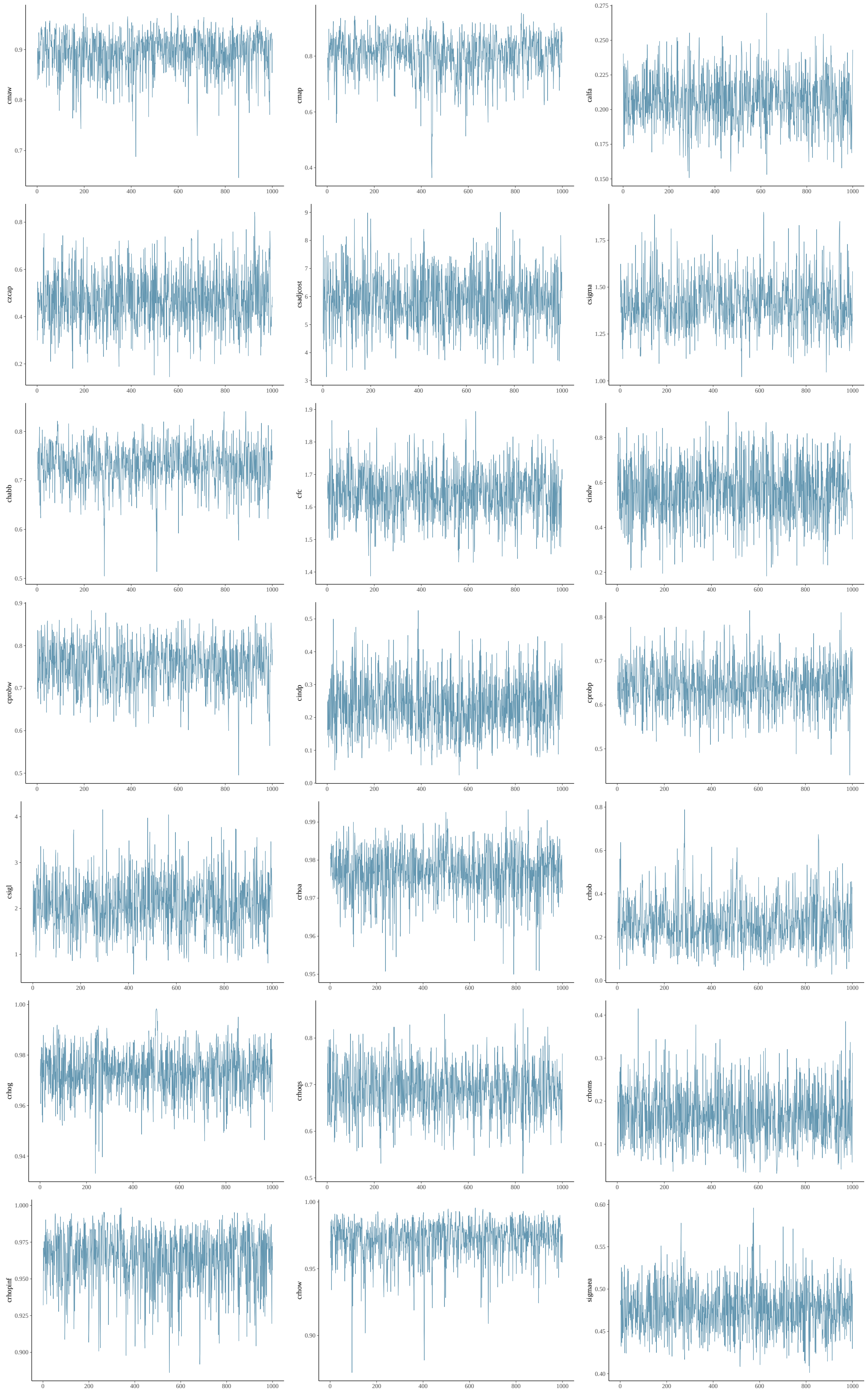

sigmaem
sigmaepinf
sigmaew
LL
cpie
cgamma
cbeta
clandap
cbetabar
cr
crk
cw
cikbar
cik
clk
cky
ciy
ccy
crkky
cwhlc
cwly
Autocorrelation
Lag
0.0
0.5
1.0
0
5
10
15
20


conster
1.0
0.5
0.0
Autocorrelation
0
5
10
15
20
Lag

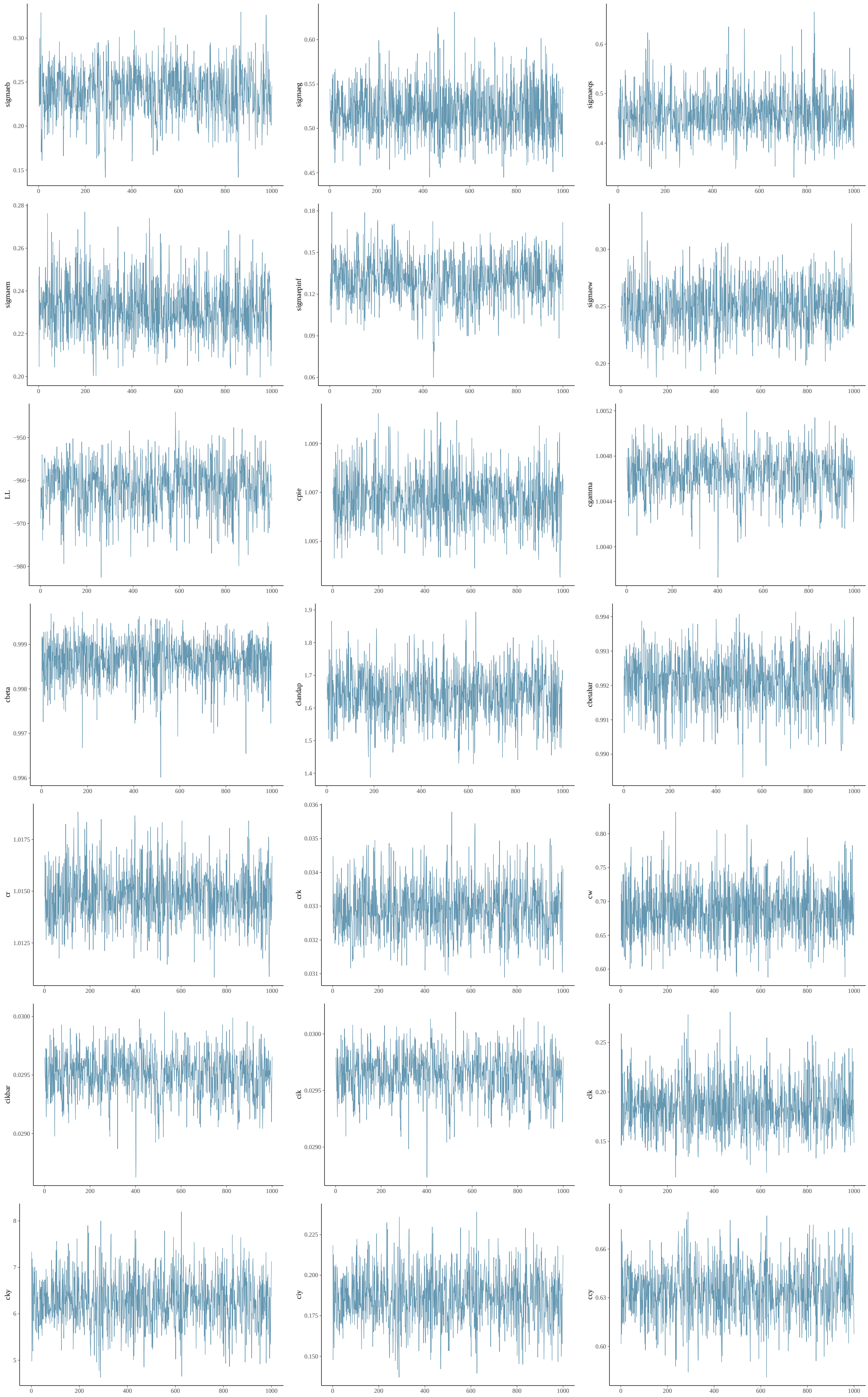
Trace Plots

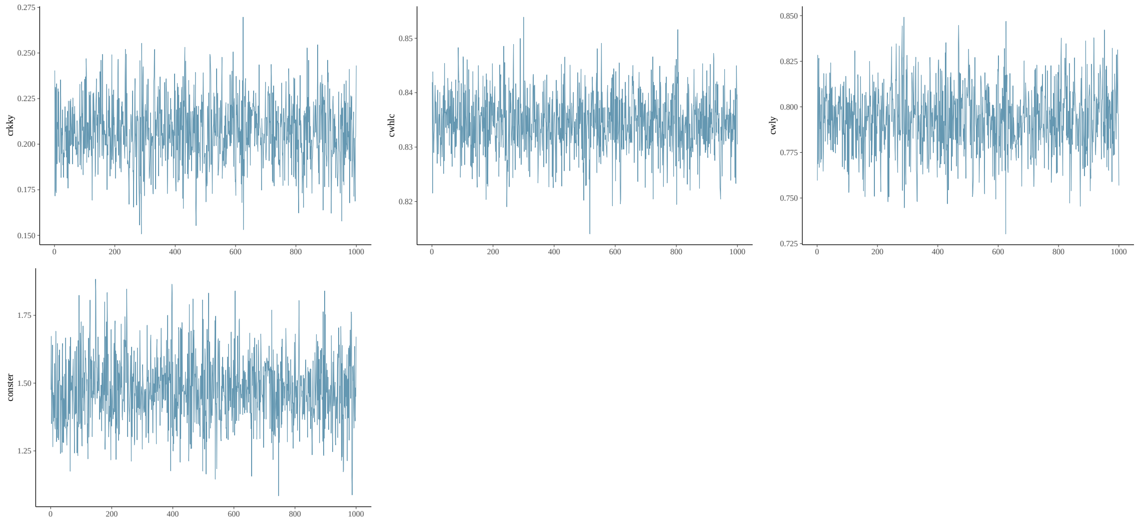
These are trace plots of crpi, crdy, cry, crr, constelab, constepinf, ctrend, constebeta, cgy, cmaw, cmap, calfa, czcap, csadjcost, csigma, chabb, cfc, cindw, cprobw, cindp, cprobp, csigl, crhoa, crhob, crhog, crhoqs, crhoms, crhopinf, crhow, sigmaea, sigmaeb, sigmaeg, sigmaeqs, sigmaem, sigmaepinf, sigmaew, ctou, cg, curvp, curvw, clandaw, LL, cpie, cgamma, cbeta, clandap, cbetabar, cr, crk, cw, cikbar, cik, clk, cky, ciy, ccy, crkky, cwhlc, cwly, conster, Iter, for all chains. Trace plots provide a visual way to inspect sampling behavior and assess mixing across chains. The iteration number (x-axis) is plotted against the parameter value at that iteration (y-axis). Divergent transitions are marked on the x-axis. A good plot shows chains that move swiftly through the parameter space and all chains that explore the same parameter space without any divergent transitions. A bad plot shows chains exploring different parts of the parameter space, this is a sign of non-convergence. If there are divergent transitions, looking at the parameter value related to these iterations might provide information about the part of the parameter space that is difficult to sample from. Slowly moving chains are indicative of high autocorrelation or small integrator step size, both of which relate to ineffective sampling and lower effective sample sizes for the parameter.





cmaw
0.7
0.8
0.9
cmap
0.4
0.6
0.8
calfa
0.150
0.175
0.200
0.225
0.250
0.275
czcap
0.2
0.4
0.6
0.8
csadjcost
3
4
5
6
7
8
9
csigma
1.00
1.25
1.50
1.75
chabb
0.5
0.6
0.7
0.8
cfc
1.4
1.5
1.6
1.7
1.8
1.9
cindw
0.2
0.4
0.6
0.8
cprobw
0.5
0.6
0.7
0.8
0.9
cindp
0.0
0.1
0.2
0.3
0.4
0.5
cprobp
0.5
0.6
0.7
0.8
csigl
1
2
3
4
crhoa
0.95
0.96
0.97
0.98
0.99
crhob
0.0
0.2
0.4
0.6
0.8
crhog
0.94
0.96
0.98
1.00
crhoqs
0.5
0.6
0.7
0.8
crhoms
0.1
0.2
0.3
0.4
crhopinf
0.900
0.925
0.950
0.975
1.000
crhow
0.90
0.95
1.00
sigmaea
0.40
0.45
0.50
0.55
0.60
0
200
400
600
800
1000


sigmaeb
0.30
0.25
0.20
0.15
sigmaeg
0.60
0.55
0.50
0.45
sigmaeqs
0.6
0.5
0.4
sigmaem
0.28
0.26
0.24
0.22
0.20
sigmaepinf
0.18
0.15
0.12
0.09
0.06
sigmaew
0.30
0.25
0.20
LL
−950
−960
−970
−980
cpie
1.009
1.007
1.005
cgamma
1.0052
1.0048
1.0044
1.0040
cbeta
0.999
0.998
0.997
0.996
clandap
1.9
1.8
1.7
1.6
1.5
1.4
cbetabar
0.994
0.993
0.992
0.991
0.990
cr
1.0175
1.0150
1.0125
crk
0.036
0.035
0.034
0.033
0.032
0.031
cw
0.80
0.75
0.70
0.65
0.60
cikbar
0.0300
0.0295
0.0290
cik
0.0300
0.0295
0.0290
clk
0.25
0.20
0.15
cky
8
7
6
5
ciy
0.225
0.200
0.175
0.150
ccy
0.66
0.63
0.60
0
200
400
600
800
1000


crkky
0.275
0.250
0.225
0.200
0.175
0.150
0
200
400
600
800
1000
cwhlc
0.85
0.84
0.83
0.82
0
200
400
600
800
1000
cwly
0.850
0.825
0.800
0.775
0.750
0.725
0
200
400
600
800
1000
conster
1.75
1.50
1.25
0
200
400
600
800
1000

IMFS Working Paper Series

Recent Issues

143 / 2020	Gregor Boehl Felix Strobel	U.S. Business Cycle Dynamics at the Zero Lower Bound
142 / 2020	Gregor Boehl Gavin Goy Felix Strobel	A Structural Investigation of Quantitative Easing
141 / 2020	Karl-Heinz Tödter	Ein SIRD-Modell zur Infektionsdynamik mit endogener Behandlungskapazität und Lehren für Corona-Statistiken
140 / 2020	Helmut Siekmann Volker Wieland	The Ruling of the Federal Constitutional Court concerning the Public Sector Purchase Program: A Practical Way Forward
139 / 2020	Volker Wieland	Verfahren zum Anleihekaufprogramm der EZB
138 / 2020	Francisco Gomes Michael Haliassos Tarun Ramadorai	Household Finance
137 / 2019	Martin Kliem Alexander Meyer-Gohde	(Un)expected Monetary Policy Shocks and Term Premia
136 / 2019	Luc Arrondel Hector Calvo-Pardo Chryssi Giannitsarou Michael Haliassos	Informative Social Interactions
135 / 2019	Tiziana Assenza Alberto Cardaci Domenico Delli Gatti	Perceived wealth, cognitive sophistication and behavioral inattention
134 / 2019	Helmut Siekmann	The Asset Purchase Programmes of the ESCB – an interdisciplinary evaluation
133 / 2019	Josefine Quast Maik Wolters	Reliable Real-time Output Gap Estimates Based on a Modified Hamilton Filter
132 / 2019	Galina Potjagailo Maik Wolters	Global Financial Cycles since 1880
131 / 2019	Philipp Lieberknecht Volker Wieland	On the Macroeconomic and Fiscal Effects of the Tax Cuts and Jobs Act
130 / 2019	Eduard Hofert	Regulating Virtual Currencies

129 / 2018	Olga Goldfayn-Frank Johannes Wohlfart	How Do Consumers Adapt to a New Environment in their Economic Forecasting? Evidence from the German Reunification
128 / 2018	Christopher Roth Johannes Wohlfart	How Do Expectations About the Macroeconomy Affect Personal Expectations and Behavior?
127 / 2018	Michael Haliassos Thomas Jansson Yigitcan Karabulut	Financial Literacy Externalities
126 / 2018	Felix Strobel	The Government Spending Multiplier, Fiscal Stress and the Zero Lower Bound
125 / 2018	Alexander Meyer-Gohde Daniel Neuhoff	Generalized Exogenous Processes in DSGE: A Bayesian Approach
124 / 2018	Athanasios Orphanides	The Boundaries of Central Bank Independence: Lessons from Unconventional Times
123 / 2018	Karl-Heinz Tödter Gerhard Ziebarth	Zinsen, Effektivpreise und Lebenskosten – Ein Beitrag zur Konstruktion eines intertemporalen Preisindex
122 / 2018	Helmut Siekmann	Legal Tender in the Euro Area
121 / 2018	Maik H. Wolters	How the Baby Boomers' Retirement Wave Distorts Model-Based Output Gap Estimates
120 / 2017	Helmut Siekmann	Die Einstandspflicht der Bundesrepublik Deutschland für die Deutsche Bundesbank und die Europäische Zentralbank
119 / 2017	Gregor Boehl	Monetary Policy and Speculative Stock Markets
118 / 2017	Gregor Boehl Thomas Fischer	Can Taxation Predict US Top-Wealth Share Dynamics?
117 / 2017	Tobias H. Tröger	Why MREL Won’t Help Much
116 / 2017	Tobias H. Tröger	Too Complex to Work – A Critical Assessment of the Bail-in Tool under the European Bank Recovery and Resolution Regime
115 / 2017	Guenter W. Beck Volker Wieland	How to Normalize Monetary Policy in the Euro Area
114 / 2017	Michael Binder Jorge Quintana Philipp Lieberknecht Volker Wieland	Model Uncertainty in Macroeconomics: On the Implications of Financial Frictions