Bayesian Estimation of DSGE Models with Hamiltonian Monte Carlo*

Mátyás Farkas† European Central Bank

Balint Tatar‡ Goethe-University Frankfurt

First version: August 31, 2020
This version: June 2, 2023

Abstract

In this paper we adapt the Hamiltonian Monte Carlo (HMC) algorithm to Dynamic Stochastic General Equilibrium (DSGE) models, a method currently applied in various fields owing to its superior sampling and diagnostic properties. We implement it into a freely available state-of-the-art, high-performance software (Stan). We estimate a small-scale textbook New Keynesian model and the Smets-Wouters model using US data. We find that in particular cases the HMC algorithm is faster by over an order of magnitude compared to the standard sampling method. Our results and sampling diagnostics confirm the parameter estimates available in the existing literature. In addition, we find bimodality in the Smets-Wouters model even if we estimate it by using the original tight priors. Finally, we combine the HMC framework with the Sequential Monte Carlo (SMC) algorithm to create a powerful tool which enables us to estimate DSGE models with ill-behaved posterior densities.

Keywords: Bayesian Analysis, DSGE Estimation, Hamiltonian Monte Carlo, Sequential Hamiltonian Monte Carlo

JEL-Codes: C11,C15,E10

*We would like to express our special gratitude to Nils Bertschinger (formerly FIAS) and Alexander Meyer-Gohde from IMFS for all the helpful discussions and suggestions. We also thank to Michael Betancourt for sharing resources and codes and for the related discussions from which this paper benefited. Furthermore, we thank to Michael Binder and Volker Wieland from IMFS and Harald Uhlig. We also thank Michele Lenza, Luca Dedola, Péter Karádi and participants of RCC5 seminar 11th July 2019 in the ECB for the very helpful comments and discussions, the participants of the 2021 Annual Conference of the International Association for Applied Econometrics (IAAE), participants of the IAAE Webinar Series and participants of other seminars. 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@googlemail.com

1 Introduction

Dynamic Stochastic General Equilibrium (DSGE) models have been shaping modern macroeconomic theory since the seminal contribution made by Kydland and Prescott (1982). In the past two decades DSGE models have been extended along a number of lines and have become the workhorse framework for analysing economic fluctuations and forecasting. Likelihood-based estimation has increasingly gained attraction[1], and a sufficiently proper fit of the empirical data has been achieved compared with other model classes.

The aim of this paper is to contribute to the literature on DSGE estimation by implementing the Hamiltonian Monte Carlo (HMC) algorithm for DSGE models. The HMC is widely used in numerous fields of academic research owing to its superior features such as high efficiency and enhanced diagnostics compared with other Markov Chain Monte Carlo (MCMC) methods. As an extension, we merge the HMC with the Sequential Monte Carlo (SMC) algorithm to estimate ill-behaved, bimodal posterior densities in an established DSGE model.

The pioneering work on Bayesian DSGE model estimation dates back to Schorfheide (2000) and Otrok (2001) and serves as the core of other MCMC-based estimation methods. Herbst and Schorfheide (2015) provides an excellent summary on the latter algorithms. To address the challenge of increasing modeling complexity, sequential sampling based estimation methods were developed as well, e.g. particle filters, see Fernandez-Villaverde and Rubio-Ramirez (2007) and Herbst and Schorfheide (2019) or the SMC algorithm. The SMC method was first applied to DSGE models by Creal (2007), then formalized by

[1]Initially, simple DSGE models were calibrated to match only selected moments of the data due to their restrictive nature. A complete review of the methodology and the transition from small scale calibrated DSGE models to the state-of-the-art estimation of medium to large scale models would be beyond the scope of this paper, therefore we refer to the excellent work of Fernandez-Villaverde et al. (2016).

Herbst and Schorfheide (2014) and constitutes also the core of Cai et al. (2021).

Although the standard MCMC-type estimation framework is widely applied in the DSGE literature and is readily available in standard software, it suffers from several weaknesses which have mostly been addressed either inadequately or not at all. Firstly, the simulated sample draws often suffer from considerably high autocorrelation, which results in a very small effective sample size. A common approach to addressing this shortcoming is to run longer and multiple chains and to consider only each n-th draw to obtain uncorrelated samples. Yet, thinning the Markov chain can render sampling inefficient as it easily becomes time-consuming, particularly when the dimension of the parameter space and the model itself is high. Secondly, the standard random-walk MCMC algorithm will explore the typical set only slowly in higher dimensional spaces. Large transitions from one point to another in the typical set will not be possible because the number of directions to move the chain increases exponentially in line with the dimension. Consequently, "we need a better way of exploring the typical set" and "to better exploit the geometry of the typical set itself", as suggested by Betancourt (2018, p.16.). Thirdly, although in theory the MCMC algorithm converges asymptotically to the target distribution under certain regularity conditions, in practice this convergence might occur very slowly. One of the key questions – namely whether the Markov chain has already converged to the target distribution – lacks a clear answer for the time being. Unfortunately, there is no one single way of addressing the latter issue. 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", as also pointed out by Brooks and Gelman (1998, p.445.). Even if standard diagnostics suggests that convergence has not yet been reached, it will be challenging to identify the reason for non-convergence, as

available diagnostics does not provide any information on irregular regions of the posterior distribution. Fourthly – although not completely unaddressed – it is common practice to conduct a mode search to specify the starting point for MCMC-type samplers. This practice is closely followed in the DSGE literature, and several algorithms for posterior mode search are readily available in standard software for DSGE estimation. However, mode search algorithms often fail to provide a suitable starting point or, alternatively, their execution might be tedious and time-consuming. Furthermore, Betancourt (2018) argues that, in general, the mode may not be representative of the typical set as the latter can be increasingly distant from the mode in higher dimensions.

An advanced algorithm leveraging the information in the geometry of the typical set and thereby addressing the above issues is the HMC algorithm. Similarly to the Kalman filter and MCMC, it has its roots in physics and dates back to Duane et al. (1987).[2] It is considered to be the new standard in high-dimensional numerical simulation, where the gradient of the target density can be evaluated. Given the accessibility of the advanced software package for Bayesian estimations (Stan) implementing the HMC algorithm, this methodology is currently being applied by many researchers in various fields. The HMC has been shown to have significantly better sampling properties than the baseline algorithm – the Random-Walk Metropolis-Hastings (RWMH), see e.g. Neal (2011) – where the latter algorithm is the workhorse estimation framework for DSGE models. Herbst and Schorfheide (2015) also acknowledge the advantages of the HMC and state that progress in this direction in the field of DSGE model estimation would be preferable.

To the best of our knowledge, this is the first available paper to apply the HMC

[2]In the original work of Duane et al. (1987) it was called 'Hybrid Monte Carlo' and designed for the numerical simulation of lattice field theory simulations of quantum chromodynamics.

algorithm to DSGE models and to present any results.[3] We have implemented the HMC in Stan because it uses C++, a low-level, high-performance programming language.[4] This tool is suitable for dealing with complex models and symbolic differentiation, which makes their accurate implementation feasible. There is therefore no need to rely on approximations of the gradient of the posterior likelihood function. It also comes complete with a set of powerful diagnostics and a visualisation toolkit, which are readily available and unique to HMC and provide further evidence of whether the typical set has been explored appropriately.[5]

In contrast to the RWMH sampler, the HMC is more efficient because in the optimum its draws are uncorrelated. This implies that the HMC is more likely to explore the typical set properly even in higher-dimensional problems. A further advantage of the HMC is that time-consuming and potentially erroneous mode estimation can be abandoned because its gradient-based sampling provides guidance on how to find and explore the typical set appropriately; see Betancourt (2018). Moreover, the diagnostics is capable of revealing inefficient or improper sampling caused by fat tails, funnels or high curvature and may also serve as an indication of difficult-to-sample parameters, which are potentially attributable to weak identification.[6]

To demonstrate the features of the HMC algorithm we first present stylised examples and then estimate the textbook small-scale New Keynesian model from Herbst and

[3] In contemporaneous work Fernandez-Villaverde and Guerron-Quintana (2021) propose the application of HMC for DSGE estimation and work is in progress to apply it to differentiable state space models.

[4] C++ can be considered as a superset of the C language with features added as e.g. object-oriented programming, exception handling, a rich C++ library and improved memory management.

[5] For a recent review of the work-flow with Stan we refer to Gabry et al. (2019).

[6] In particular, if the autocorrelations of the sample draws generated by the HMC algorithm does not decay to zero after the first lag, it indicates that a specific parameter is difficult to sample implying that the posterior is either oddly shaped, or the parameters suffer from weak identification. In contrast, the high autocorrelation of the Metropolis-Hastings algorithm is an inherent feature and thus cannot be used to identify problematic parameters.

Schorfheide (2015) and the Smets and Wouters (2007) model (hereafter SW model). We find that in particular cases the HMC algorithm provides a twenty-fold improvement in sampling speed over the RWMH algorithm. The results we obtain with the HMC algorithm are very similar to those in the existing literature. Given the HMC diagnostics, we wish to point out that there are no severe local irregularities such as high curvature or funnels which could frustrate the HMC sampler. This can be stated based on a relatively small number of sample draws. In the SW model, however, certain parameters are difficult to sample from. These new insights suggest that abandoning the mode search and applying gradient-based guided HMC sampling is beneficial owing to its unique features that are not available to RWMH sampling around the mode. Furthermore, we also find that a handful of parameters in the SW model feature bimodal posterior distributions even if tight priors are set and the entire data sample is used for the estimation.

Although implementing 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 shares one drawback with RWMH: it fails to deal with multimodal target densities. If modes are separated by large energy barriers, e.g. the posterior likelihood function has no support between them, the HMC will get stuck in one mode and will not be able to escape within a reasonable time. To address this shortcoming we merge the HMC with the SMC algorithm presented in Herbst and Schorfheide (2014) and explore ill-shaped, bimodal posterior densities in the SW model when less restrictive priors are set.

The remaining part of this paper is organised as follows. In Section 2 we present the HMC algorithm and describe our approach to adapting it to the estimation of DSGE models. Section 3 contains stylised examples that visualise unique features of the HMC algorithm. It also presents the estimation results of the textbook small-scale New Key-

nesian DSGE model and the SW model. Section 4 extends the paper by combining the HMC and the SMC to estimate ill-behaved posterior densities. Section 5 concludes the paper.

2 The Hamiltonian Monte Carlo Method

This section starts with an overview of the general Bayesian DSGE estimation framework and provides then an intuitive description of the operation of the HMC algorithm in general. Afterwards we explain how to adapt the HMC algorithm to estimate linearized DSGE models. For a more extensive treatment we refer to the Technical Appendix.

To estimate a Bayesian model, the first step is to specify the joint distribution of the data and the model parameters, $P(Y, \theta)$, represented by its corresponding density function, $p(Y, \theta)$.[7] To obtain the posterior density, $p(\theta|Y)$, one can apply Bayes' rule:

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

where $p(Y|\theta)$ is referred to as the likelihood function and $p(\theta)$ is the density function of the prior distribution. To specify a likelihood function conditioned on the parameters and to 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 state variables which can be carried out by a variety of DSGE solution algorithms.[8] In case the DSGE model has a unique stable solution it can be represented in the following VAR-form:

$$s_t = G_0(\theta)s_{t-1} + G_1(\theta)\epsilon_t \tag{2}$$

where s_t is the vector of model variables at time t, ϵ_t is the error term vector and $G_0(\theta)$

[7] Throughout this paper, distributions will be represented by their corresponding density functions.

[8] See e.g. Blanchard and Kahn (1980), Binder and Pesaran (1997), King and Watson (1998), Uhlig (1999), Klein (2000), Kim (2000), Christiano (2002), Sims (2002) and Anderson (2010).

and $G_1(\theta)$ are matrices of appropriate dimensions depending on the parameter vector θ. The above VAR-form is linked to the data by means of the measurement equation:

$$y_t = H_0(\theta) + H_1(\theta)t + H_2(\theta)s_t + u_t \tag{3}$$

where y_t represents the observed data vector, u_t the measurement error, H_0, H_1 and H_2 are again appropriate matrices. The state space representation permits the expression of the joint density function for the observed data and the DSGE model variables where the latter are generally unobserved:

$$p(Y_{1:T}, S_{1:T}|\theta) = \prod_{t=1}^{T} p(y_t, s_t|Y_{1:t-1}, S_{1:t-1}, \theta) = \prod_{t=1}^{T} p(y_t|s_t, \theta)p(s_t|s_{t-1}, \theta) \tag{4}$$

$p(y_t|s_t, \theta)$ and $p(s_t|s_{t-1}, \theta)$ are the conditioned probability density functions of the observables and the states given the parameters and the present and lagged values of the states. To obtain the likelihood function the unobserved states, s_t, have to be integrated out. For linearized DSGE models with Gaussian error terms one can apply the Kalman filter for the latter purpose and to obtain the log-likelihood function, $p(Y_{1:T}|\theta)$. Once the prior density of the parameters, $p(\theta)$, is specified one can set up the RWMH algorithm to sample from the posterior density by applying Algorithm 1.[9] 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.[10] The scaling parameter, c_0, should be chosen in a way that the acceptance ratio equals to 23.4 percent, that was proven to be the optimal acceptance ratio under specific assumptions, see Roberts et al. (1997). In practice, this parameter is chosen in a way that the acceptance ratio lies in the range of 20-40 percent.

There exist several other modified versions of the MH-algorithm: the Block-MH algorithm, applied e.g. by Cúrdia and Reis (2010), the Metropolis-Adjusted Langevin

[9] This algorithm summarizes the main steps extensively described in Herbst and Schorfheide (2015).
[10] As the density function of the normal distribution is symmetric, the proposal densities cancel.

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 includes the solution of the DSGE model for θ and the evaluation of $p(Y|\theta)$ and $p(\theta)$.
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 $N(\tilde{\theta}, c_0^2\tilde{\Sigma})$), solve the DSGE model for $\theta^{(0)}$ and evaluate $p(Y|\theta^{(0)})$ and $p(\theta^{(0)})$.
4. For $n = 1, ..., N$
   (a) Draw θ' from the proposal distribution $\mathcal{N}(\theta^{(n-1)}, c_0^2\tilde{\Sigma})$.
   (b) Solve the DSGE model for θ' and evaluate $p(Y|\theta')$ and $p(\theta')$.
   (c) Accept θ', i.e. $(\theta^{(n)} = \theta')$, with probability $\min\{1, f(\theta^{(n-1)}, \theta'|Y)\}$ and reject $(\theta^{(n)} = \theta^{(n-1)})$ otherwise, where
$$f(\theta^{(n-1)}, \theta'|Y) = \frac{p(Y|\theta')p(\theta')q(\theta'|\theta^{(n-1)})}{p(Y|\theta^{(n-1)})p(\theta^{(n-1)})q(\theta^{(n-1)}|\theta')}.$$
5. Estimate the posterior expected value of the function $h(\theta)$ by $\frac{1}{N}\sum_{i=1}^{N} h(\theta^{(i)})$.

(MALA) algorithm, originally proposed by Besag (1994) and the MH-Newton algorithm, see Qi and Minka (2002).[11] In addition to the very popular strand of first-order linearized DSGE models there exist more elaborated ones with non-linear state spaces. To evaluate the likelihood function in this more complex environment, particle filters were proposed in the literature, e.g. in Fernandez-Villaverde and Rubio-Ramirez (2007). At the same time, particle filters such as the SMC algorithm in Herbst and Schorfheide (2014) are also applied if the posterior likelihood is irregularly shaped which may even occur when standard models are estimated using first order linear approximations. Our extension of the HMC algorithm with SMC falls into the latter category of applications.

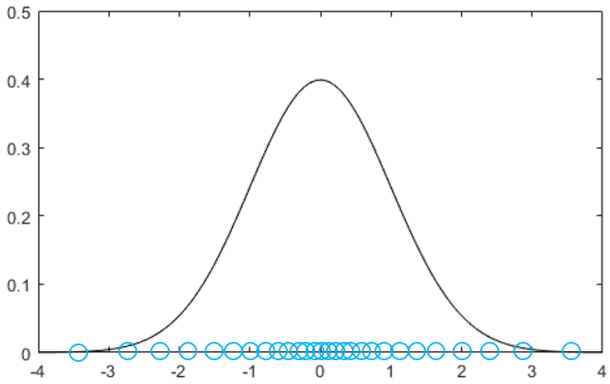
Similarly to advanced MCMC methods from above, the HMC algorithm builds on the information provided by the gradient of the log-posterior density function. It uses the information in the geometry of the target distribution, that is, its shape and the equations

[11] For further details we refer to the Technical Appendix and an excellent assessment can be found in Herbst and Schorfheide (2015).

characterizing it. Its main advantage is that by means of the Hamiltonian equations, a concept borrowed from physics, the algorithm enables to propose a new parameter draw θ' which is distant from the current one, θ, while it maintains very high acceptance rates.

To illustrate the concept of HMC let us apply it to a simple example.[12] We assume that one intends to sample from a one dimensional standard normal distribution with density function $f(q) = 1/(2\pi)^{1/2}\exp\{q^2/2\}$. The aim is to generate samples for q from each location of the domain of q, the real line, in proportion to the value of the density function, $f(q)$, at that location of the domain. This case is illustrated in Figure 1.

Figure 1: Sample Draws from a Normal Distribution



Notes: The blue circles show desired sample draws from a standard normal target density, displayed in black color.

To understand the idea of the HMC algorithm let us make a small excursion to physics. In classical mechanics the dynamics of a mechanical system over time given a particle's position and momentum is modeled by functions measuring the potential and kinetic energy of the particle. Classical examples of a mechanical system are a bouncing ball, a pendulum or an oscillating spring. The evolution of such a system can be perfectly described by the Hamiltonian equation, measuring the total energy which is the sum of

[12] Our explanation draws on the example in the excellent book of Lambert (2018).

potential energy, $U(q)$, and kinetic energy, $K(p)$:

$$H(p,q) = U(q) + K(p) \tag{5}$$

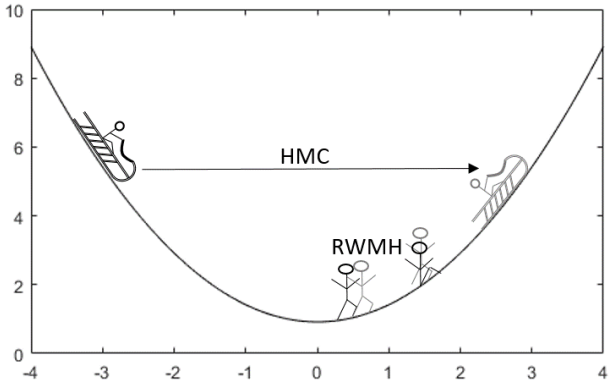
The change in the position q and in the momentum p, both of dimension d, respectively, over time is characterized by the solution to the following system of partial differential equations derived from the Hamiltonian equation:

$$\frac{dq_i}{dt} = \frac{\partial H(q,p)}{\partial p_i} \quad \forall i = 1, ..., d$$
$$\frac{dp_i}{dt} = -\frac{\partial H(q,p)}{\partial q_i} \quad \forall i = 1, ..., d$$

In an ideal, frictionless mechanical system the total energy always remains constant which turns out to be a handy property when applying the Hamiltonian in a probabilistic setting.

To apply this concept to our simple sampling problem from above let us consider instead of $f(q)$ the negative logarithm of the density function $g(q) := -\log f(q)$. Thereby we flip over $f(q)$ and obtain a valley-shaped function $g(q)$, being the new target density. Next, we imagine a person using a sleigh and trying to explore the valley covered in snow, i.e. the new target density defined by $g(q)$. We assume that our explorer starts out at some point in the valley, sitting on a sleigh, somewhere on the surface generated by the function $g(q)$ and is initially pushed randomly with some impulse, either uphill or downhill, see Figure 2. In contrast, think of the RWMH algorithm as a person exploring the valley by foot and proceeding in equally distant steps in random directions. Assuming that our explorer on the sleigh started her journey downhill, after passing the trough of the valley she will continue sliding uphill. After getting gradually slower she will stop at some point and will start sliding again downhill into the opposite direction, and so on.

Figure 2: Hamoltionan Monte Carlo vs. Random Walk Metropolis Hastings



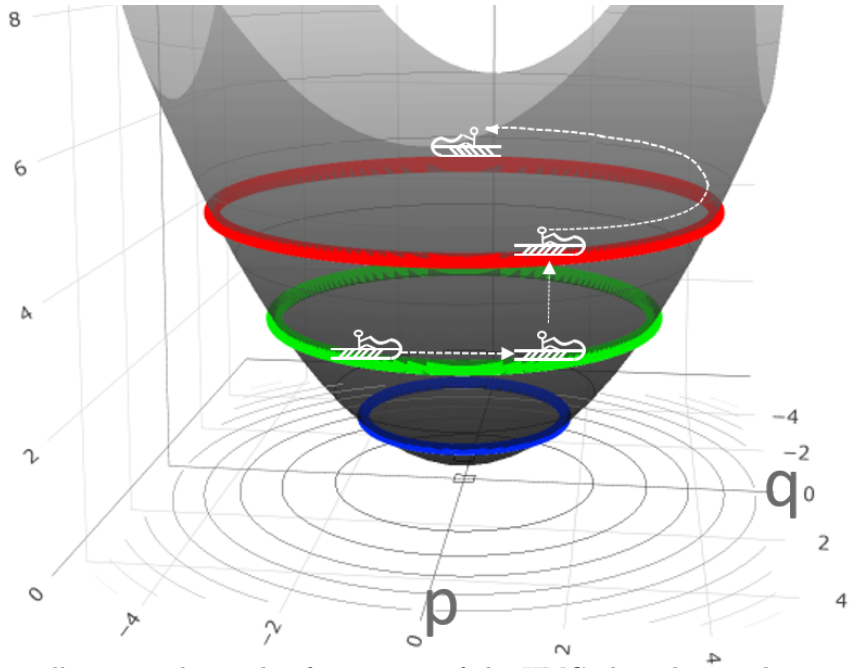
Notes: The figure shows an illustrative comparison between the mode of operation of the RWMH and the HMC algorithm if sampling from a one dimensional normal density.

Assuming also that the snow-covered surface of the valley is frictionless, this motion will continue forever as the total energy remains constant. The motion of the sleigh on a frictionless surface can be perfectly described by the Hamiltonian equation in analogue to the mechanical systems mentioned before. The total energy of the sleigh can be described by its potential energy function $U(q) := g(q)$ and its kinetic energy $K(p) := |p|^2/(2m)$, depending on its momentum p and m corresponds to the mass of the sleigh with the person. By moving up or down the valley the explorer on the sleigh exchanges potential energy $U(q)$ for kinetic energy $K(p)$. As the sleigh slides down (up) the hill, its potential energy will decrease (increase) and its kinetic energy increase (decrease). After some time $t = T$ we record the position q of our sleighing explorer. Given the shape of the terrain the person with the sleigh will be more often in lower regions of the valley than in higher regions. In contrast, a RWMH explorer would find it easy to walk downhill to the trough of the valley, but would struggle to make steps upwards. In other words, under RWMH sampling a proposal is always accepted if the probability of the new proposal q' is larger than the probability of the current draw q, steps downhill, and only accepted randomly depending on the proportion of the probabilities of q' and q, steps uphill. Therefore, by applying RWMH sampling, the explorer would walk down to the valley, and explore the

region in the bottom, i.e. the target density around the mode, but always struggle to make steps uphill and proceed only very slowly.

A straightforward question to ask is: Why is the explorer on the sleigh able to effortlessly slide and to always arrive safely at the other side of the valley? The core of the concept is that by tracking the momentum p and the kinetic energy $K(p)$ the parameter space is extended by the same dimension to measure total energy. This extended space is called the phase space and is fundamental to Hamiltonian dynamics. Visually, extending the parameter space and tracking both position and momentum at the same time allows the sleigh moving from one side to the other side of the valley just by sliding around on the same (energy) level, hence moving only horizontally in this simple case, see Figure 3.

Figure 3: Visualization of the Hamiltionan Monte Carlo Algorithm



Notes: The figure illustrates the mode of operation of the HMC algorithm in the extended parameter space if used to sample from a normal density. When the sleigh moves horizontally on the same height the Hamiltonian equations are used to calculate the path and keep the sleigh on the same energy level. The vertical move of the sleigh demonstrates a new impulse draw after the sleigh was stopped and the sample draw was recorded. The new impulse moves the sleigh to a different energy level on which it again continues exploring the extended parameter space horizontally along the new path defined by the Hamiltonian equations. Source: author's illustration.

The reason why the sleigh remains at the same energy level is that in a frictionless mechanical system the total energy always remains constant. Therefore, in the extended

space the application of the Hamiltonian equation ensures that along the Hamiltonian path the acceptance probability is always unity and the new proposal is always accepted, as long as the system of differential equations can be solved accurately. In this extended space, stopping and then pushing the sleigh with a different impulse is equivalent to 'moving' it onto a different height, hence energy level, sliding horizontally again along the Hamiltonian path, i.e. the contour lines of the extended space. When stopping the sleigh, the explorer 'brakes', throws away kinetic energy $K(p)$, e.g. by wasting it while braking, and records only the current position q. By repeating the process of pushing the sleigh with a random impulse, letting it slide and stopping it sufficiently often, the valley will be explored appropriately. The sequence of the recorded positions at each stop equals then the sequence of sample draws and the kinetic energy is simply ignored. This allows us to define Algorithm 2 for the proposal of the new sample draw and the update step in a typical Metropolis-Hastings algorithm instead of using random proposals.

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 and the momentum vector (q', p') by applying the Hamiltonian equations deterministically starting from $q = q^{(n)}$ and p.
3. Metropolis-Hastings step: accept the new proposal and set $q^{(n+1)} = q'$ with probability $\min[1, \exp(-(U(q') - U(q) + K(p') - K(p)))]$.

This update scheme can be applied to any target density function by considering the negative logarithm of the target density instead, in analogue to the simple case from above. However, the more complex sampling algorithm requires significantly larger computational resources. The main bottleneck of the algorithm is to calculate the partial derivatives of the Hamiltonian equation, in particular those of the likelihood function. Furthermore, one needs to solve the resulting system of differential equations.

To implement the algorithm for DSGE models we used Stan, a state-of-the-art probabilistic programming language for Bayesian inference written in C++ language. It allows users to set up hierarchical Bayesian models and provides an easy to apply interface to the HMC algorithm for complex models. One of the key challenges lies in the accurate solution of the Hamiltonian equations. A dedicated class of symplectic 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, hence a long period of time. Stan uses a simple implementation referred to as 'leapfrogging' to solve for the discrete-time approximation of the Hamiltonian equations which is summarized by Algorithm 3.

Algorithm 3 Leapfrogging

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

Although the algorithm is easy to implement at first glance, it generates a further challenge, especially when applied in the context of DSGE estimation. It requires the evaluation of the gradient of the log-posterior which might be 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 which requires only a limited number of differentiation rules. The gradient is constructed via the chain rule by creating an expression tree backwards starting with the

last expression in the likelihood function. Stan is capable of differentiating any iterative algorithm which is particularly useful when implementing the estimation of DSGE models. A straightforward fully iterative DSGE solution method is the Binder and Pesaran (1997) algorithm which allows for a direct implementation in Stan. A further built-in feature of Stan is that it automatically optimizes the number of steps by means of the No U-Turn Sampling (NUTS) algorithm, see Hoffman and Gelman (2014). It avoids thereby that the sampler turns back into the direction of the actual sample draw to be updated.[13]

3 Applications

In the first part of this section we present stylized numerical examples to demonstrate the unique features of the HMC algorithm along with its novel diagnostic capabilities. In the second and third part we show the results obtained by applying the HMC algorithm to a textbook small scale New Keynesian model and subsequently to the medium scale SW model serving as the core for a wide range of applied policy models.

3.1 Stylized Numerical Examples

The HMC algorithm makes use of the information in the gradient of the posterior likelihood function and allows for traversing large distances in the parameter space while it maintains high acceptance probabilities along the Hamiltonian path. In the optimal case, when the posterior density is not ill-behaved, thus there are no irregularities in its surface, HMC is capable of sampling effortlessly and produces virtually uncorrelated sample draws. In contrast, an ill-behaved posterior density could render it difficult to be explored appropriately by a given sampler which may lead to inaccuracies of the estimation results.

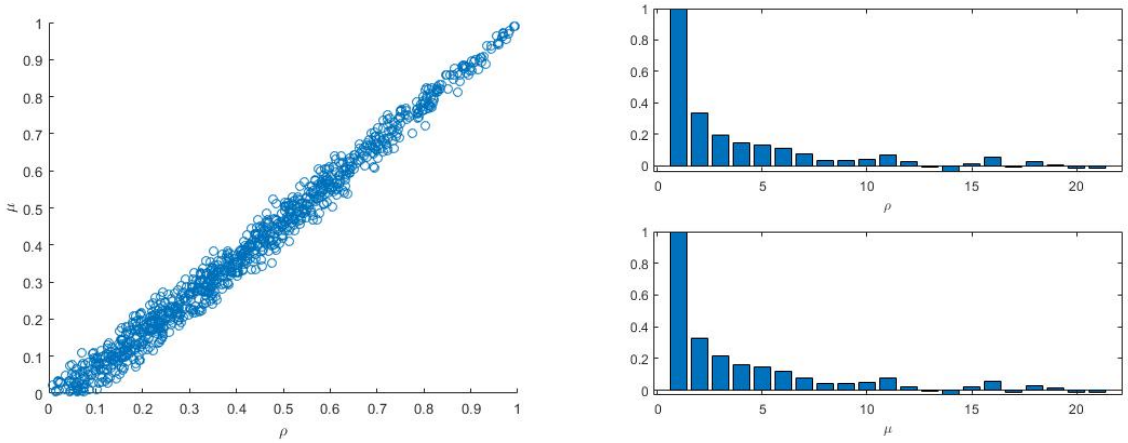
[13]For a more formal treatment of HMC and for further details on the implementation using Stan we refer to the Technical Appendix.

In the latter case, the RWMH sampler could for example get stuck. The new proposal would be typically rejected when trying to enter ill-behaved parts of the posterior density so that certain parts could remain less well or even completely unexplored. In practice, the RWMH algorithm is tuned to accept between 20-40 percent of the proposed updates, therefore rejections or high autocorrelation in the sample draws would per se not indicate an irregularity of the posterior distribution and possible sampling issues. In the presence of irregularities, the HMC algorithm would, however, struggle with producing uncorrelated sample draws or even entirely drift off from the Hamiltonian path and thus indicate sampling issues. In addition, due to its setup and diagnostic features HMC is capable of providing information on the source of the problem. The following stylized examples aim at visualizing these unique features and diagnostic capabilities of the HMC algorithm.

Autocorrelated sample draws

ARMA(p, q) processes are typically applied in a DSGE framework to model exogenous processes with anticipated shocks, often referred to as *news shocks* or *foresight* in the economic literature, see e.g. the price and wage markup shock processes in the SW model. Although in certain cases it might be well-justified to specify ARMA(p, q) shock processes, it may also lead to estimation issues. For example, if the data generating process is white noise, the model would become overidentified and parameters weakly identified, which could result in inefficient sampling or wrong conclusions and interpretation of the parameters. Furthermore, the posterior distribution could degenerate completely which would lead to severe sampling issues. In case the posterior is not completely degenerate, due to its construction, HMC may deliver information on problems with model specification. To visualize this, we generate a white noise process $u_t \overset{iid}{\sim} N(0, 0.1)$ with 1,000 observations and fit an ARMA(1,1) process $u_t = \rho u_{t-1} + \epsilon_t - \mu\epsilon_{t-1}$ with uniform priors for $\rho, \mu \in [0, 1]$.

Figure 4: Sample Draws and Autocorrelations – ARMA(1, 1)

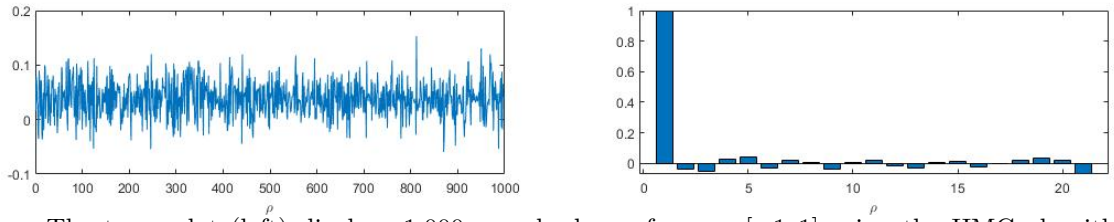


Notes: The scatter plot (left) displays 1,000 sample draws for $\rho, \mu \in [0, 1]$ using the HMC algorithm if estimating an ARMA(1,1) process $u_t = \rho u_{t-1} + \epsilon_t - \mu \epsilon_{t-1}$ whereas the data generating process is $u_t \overset{iid}{\sim} N(0, 0.1)$. The bar charts (right) show the estimated correlograms for ρ and μ.

For the standard deviation of the disturbance, σ, we postulate an inverse gamma prior distribution $\sigma \sim IG(0.1, 2)$. The scatter plot in Figure 4 shows 1,000 sample draws for ρ and μ obtained by applying the HMC algorithm.[14] A visual inspection of the sample draws from the realized posterior distribution suggests that the joint distribution of ρ and μ corresponds to a ridge along the main diagonal of $[0, 1] \times [0, 1]$ as any combination of $\rho = \mu$ exactly offsets the mutual effect of the parameter pair. For the generated data series ρ and μ are estimated at approximately 0.42 and 0.38, respectively, being significantly different from zero. While observationally all combinations of ρ and μ along the ridge should generate very similar time series, their interpretation is different though. As shown in Figure 4 the sample draws for ρ and μ exhibit autocorrelation which does not appear to be severe at first sight as it does not render the effective sample size chronically low. However, if the model is not overidentified, hence we fit a pure AR(1) process with $\rho \in (-1, 1)$, HMC samples effortlessly and one obtains uncorrelated sample draws. ρ is estimated then at 0.04 which corresponds approximately to the true value. In both cases σ is estimated at very close to its true value of 0.1. This feature of the HMC algorithm

[14] For the tuning and warm-up phase 1,000 sample draws were used, respectively.

Figure 5: Sample Draws and Autocorrelations – AR(1)



Notes: The trace plot (left) displays 1,000 sample draws for $\rho \in [-1, 1]$ using the HMC algorithm if estimating an AR(1) process $u_t = \rho u_{t-1} + \epsilon_t$ whereas the data generating process is $u_t \overset{iid}{\sim} N(0, 0.1)$. The bar chart (right) shows the estimated correlogram for ρ.

also translates into the estimation of a DSGE framework. We generated a data sample of 200 observations for the textbook small scale New Keynesian model proposed in Herbst and Schorfheide (2015) by using the parameter estimates reported therein. However, we fitted for the *iid* disturbances of both the technology and the government spending process an ARMA(1,1) process. HMC delivered again significantly autocorrelated sample draws for the parameters of the above processes as the latter overspecification resulted in irregularities and low effective sample sizes.[15] Overall, it appears that in the presence of irregularities in the posterior distribution, possibly also related to identification issues, the HMC algorithm is not capable of sampling effortlessly and produces autocorrelated sample draws. Even if autocorrelation is not as severe that it would result in a chronically low effective sample size it might be an indication of improper model specification. Hence, the unique feature of the HMC algorithm to produce uncorrelated sample draws even in higher dimensions may also contribute to uncover general issues with the model setup.

Divergent transitions

In practice we always deal with approximative models and finite data sets – in macroeconomics the number of observations available is often relatively limited – so the geometry

[15] A summary of the effective sample sizes is provided in the Appendix. A detailed estimation of the small scale model proposed in Herbst and Schorfheide (2015) with original data is carried out in the following subsection.

of the realized likelihood function can degenerate arbitrarily severely. The more complex models are and the higher their dimension is, the more prone is the realized likelihood function to manifest strong degeneracies along the dimension of particularly weakly informed parameters. Severe degeneracies may obscure latent parts of interest in the model and hence prevent an accurate quantification of the posterior density function leading to inaccurate inferences. A further advantageous feature of the HMC algorithm is that by resorting to the information in the gradient of the posterior likelihood function and featuring a guided diffusion, it is capable of proactively diagnosing severe degeneracies which geometrically impede accurate sampling and the computation of appropriate statistics.

A typical and well documented example for a severe degeneracy is e.g. Neal's (2003) funnel, a distribution with heavily narrowing ends. In a DSGE estimation framework a model is typically solved and transformed into a VAR(1) form by applying a numerical algorithm which renders it impossible to obtain a closed form solution of the entries of the VAR(1) coefficient matrix depending on the parameters to be estimated. There will be plenty of possibilities for sums and products of parameters characterizing the entries of the VAR(1) coefficient matrix and shaping thereby the realized posterior likelihood function to induce a severely degenerated distribution along certain dimensions. To demonstrate the diagnostic capabilities of HMC we modify the stylized state space model from chapter 4.3 in Herbst and Schorfheide (2015) and borrow an example from Betancourt (2020).

$$y_t = [0\ 1]\, s_t, \qquad s_t = \begin{bmatrix} \phi_1 & 0 \\ \phi_3 & \phi_2 \end{bmatrix} s_{t-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \epsilon_t \qquad \epsilon_t \overset{iid}{\sim} N(0, 0.75^2) \tag{6}$$

The first state equation, $s_{1,t}$, resembles an exogenous shock process in a DSGE model while the second one, $s_{2,t}$ is similar to an endogenous state variable which depends on its own lag

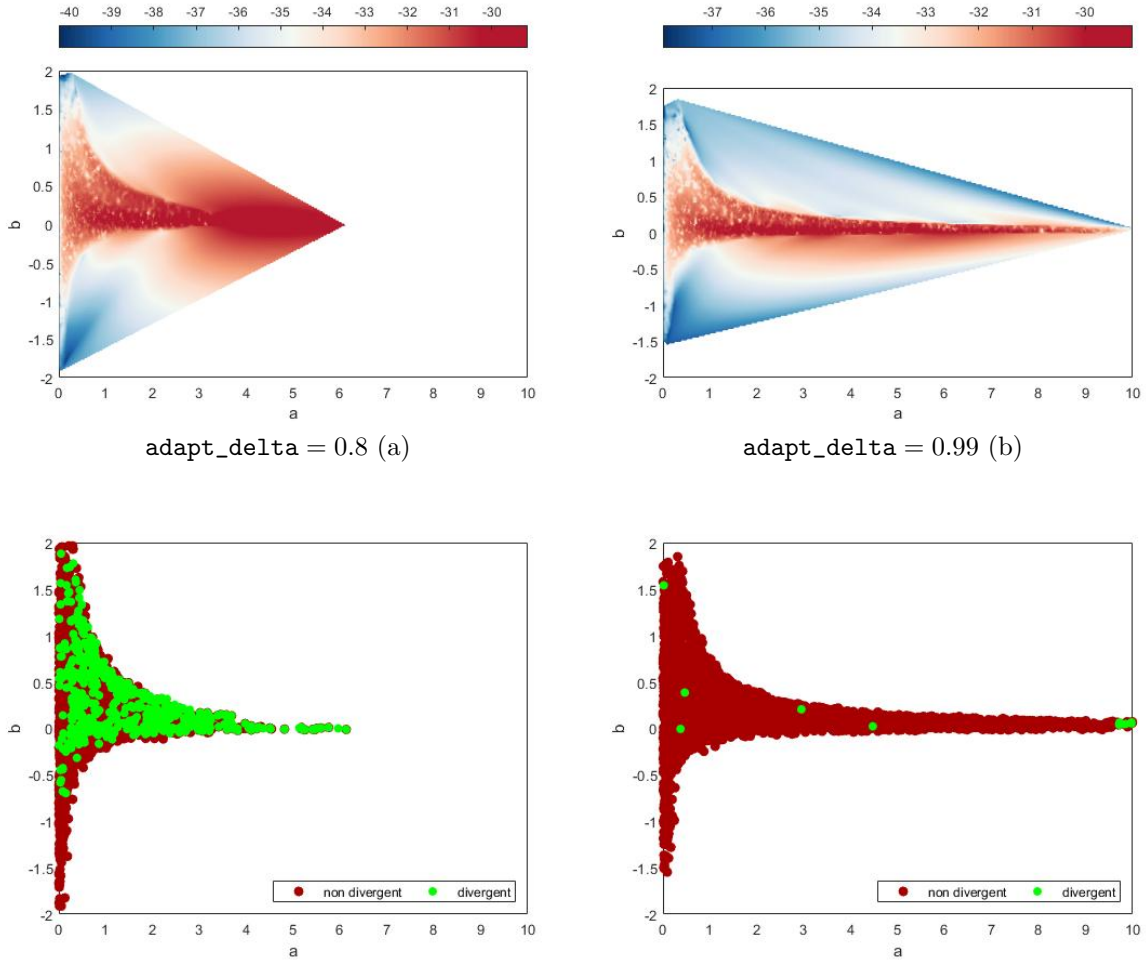
and that of the exogenous process. For visualizational purposes we create a pathological posterior distribution by setting $\phi_1 = 0.2$, $\phi_2 = ab$ with $a = 1$ and $b = 0.5$ and $\phi_3 = 1$ to generate a data sample of 100 observations. We use a uniform prior for ϕ_1 on [0,1] and also assume that $a \sim N(0, 10)$, $b \sim N(0, 1)$ with $a \in [0, 10]$ and $b \in [-2, 2]$. Furthermore, we observe only the endogenous state variable. In this setup the parameters a and b are not well-identified which creates a funnel in the posterior likelihood distribution.[16]

We sampled from the posterior distribution using the HMC algorithm with standard options as set up in Betancourt (2020). Hence, we estimate four parallel chains with 1,000 warm-up and 1,000 sample draws for each chain and let Stan tune the step size and the mass matrix. Based on our sample draws, the HMC proprietary diagnostics looks unpleasant. The diagnostics warns that approximately 20 percent of the sample draws are divergent.[17] This suggests, that the automatically calibrated step size for the leapfrog integrator during the warm-up phase was inappropriate. When moving from the present sample draw to the new proposal draw along the Hamiltonian path we drifted off the energy level defined by the approximated Hamiltonian path using the leapfrog integrator. The built in diagnostics of Stan identifies this as a divergent transition. The reason for this can be easily revealed by plotting the sample draws, see Figure 6. Our model framework generates a severely degenerated posterior density with a funnel which is an obstacle to accurate sampling. It is narrowing excessively much for the sampler and prevents it from reaching areas with high probability. To prove this, we manually adjusted the sampling options by increasing '`adapt_delta`' from the default value, 0.8, to 0.99 and force thereby the sampler to reduce the step size for the leapfrog integrator. This renders the sampler to

[16] As pointed out by Herbst and Schorfheide (2015), works of Iskrev (2010) and Komunjer and Ng (2011) provide criteria to assess whether parameters of a DSGE model are locally identifiable, however these are *ex ante* tests.

[17] In some of the simulations the divergence rate was even significantly higher.

Figure 6: Degenerated Densities – Samples and Heat Maps for Parameters a and b



`adapt_delta` = 0.8 (a)

`adapt_delta` = 0.99 (b)

`adapt_delta` = 0.8 (c)

`adapt_delta` = 0.99 (d)

Notes: Figure (a) and (b) show the interpolated heat maps, measured in terms of the value of the log posterior density, based on four chains with 1,000 warm-up and 1,000 sample draws in both cases, obtained however with different step sizes setting `adapt_delta` to 0.8 and to 0.99, respectively. Figure (c) and (d) plot the non-divergent and divergent sample draws from the same estimations with `adapt_delta` set to 0.8 and to 0.99, respectively.

expand deeper into the funnel. Although we refined the HMC sampler to the maximum as recommended by its built in diagnostics, we still remain with a low percentage of divergent transitions, possibly also attributable to high local curvature which frustrates the sampler too. In general, in presence of severe divergence of the HMC algorithm modelers should be advised to review the model framework as a whole and either reparameterize it to obtain a better behaved density or change it to resolve possible issues with identification resulting in ill-behaved densities.

We also estimated the same model using RWMH implemented in Dynare. We sampled two chains with 10 million draws each and discarded the first half. Depending on the scaling parameter configured and the resulting acceptance rate we observed that in some cases the build-in convergence statistics did not indicate severe sampling issues. However, in many cases it showed signs of possible non-convergence, whereas in general it might be also difficult to judge whether the Brooks and Gelman (1998) statistics has already settled due to the lack of a hard criterion. Furthermore, to conclude non-convergence based on the Brooks and Gelman (1998) statistics one needs to draw a very large sample due to the lack of a clear quantitative criterion which may be time consuming. We also reduced the scaling parameter of the RWMH sampler in the hope to explore the funnel better, yet sometimes it got completely stuck and the acceptance rate fell to zero.

In general, we can conclude that due to its characteristics the HMC algorithm is capable to provide researchers additional diagnostics based on the geometry of the posterior likelihood function which may facilitate to uncover the sources of pathological densities.

3.2 A Small Scale New Keynesian Model

A basic DSGE model estimated in Herbst and Schorfheide (2015) is a slightly altered version of the three equation textbook New Keynesian model (see e.g. in Clarida et al. (1999)), including a government sector.[18] It consists of a dynamic IS curve, a New Keynesian Phillips curve and a Taylor-type monetary policy rule with interest rate smoothing. Both the technology shock and government consumption is AR(1), the monetary policy shock is *iid*. For further details we refer to the Appendix.

To estimate the model, three observables are used: GDP growth, inflation and the

[18]The model is similar to the one estimated in An and Schorfheide (2007) which differs from the standard three equation New Keynesian model by assuming quadratic price adjustment à la Rotemberg (1983) instead of the Calvo (1983) scheme and by adding a government sector to the model.

nominal interest rate.[19] The model has 13 structural parameters to be estimated. The priors assumed are similar to those in Herbst and Schorfheide (2015) and are summarized in the Appendix. For the estimation, we used 10 parallel chains with each 1,000 draws. Due to the efficiency of HMC a burn-in of 500 draws is sufficient to ensure that the sampler finds regions of high probability which is also confirmed by the diagnostics.

Table 1: 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.

Table 1 shows the statistics describing the sampling efficiency of the HMC estimator 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 total energy, i.e. by the random variation of the momentum. The latter is warranted by the smart application of the gradient to set the trajectory in the phase space along the Hamiltonian, i.e. the Hamiltonian equations ensure that all draws, after initial convergence, are from the target distribution.

This improvement in efficiency is one of the main reasons why we consider HMC as

[19] In this setup, we do not allow for any measurement error.

a significant improvement for DSGE estimation. Herbst and Schorfheide (2015) report the inefficiency factor, the inverse of N_{eff}/N, for the relative risk aversion parameter (τ) for the different RWMH algorithms. They point out that the RWMH algorithm suffers from high inefficiency due to its high autocorrelation. To grasp the leap in efficiency, the naive identity matrix based Metropolis proposal is as inefficient that "100,000 draws [...] is about as accurate as an approximation obtained from 5.5 *iid* draws" (Herbst and Schorfheide, 2015, p.91.), while the standard, benchmark RWMH algorithm described in Chapter 2 has an inefficiency that increases the effective sample size to 1,137.[20] The 3-Block RWMH algorithm results in an equivalent of 2,440 *iid* draws. In comparison, the effective sample size (N_{eff}) for 100,000 draws with HMC is 89,370 for the risk aversion parameter (τ) which represents a 78.6 fold efficiency improvement over the standard, 1-Block, and a 36.6 fold over the 3-Block RWMH 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[21] renders HMC also for larger models feasible.[22]

Lastly, and probably most importantly, we highlight that due to the faster convergence of the draws to the typical set we can abandon the practice of a mode-estimation before

[20] Herbst and Schorfheide (2015) report the inefficiency factor of 88 for the 1-Block RWMH algorithm for the parameter τ. In terms of inefficiency, the HMC has a 1.12 inefficiency factor.

[21] See Technical Appendix.

[22] We also tried to run the estimation using a stock GPU, yet we did not experience any improvement, in fact it was significantly slower than using a CPU. However, as regards GPU computing in Stan work is in progress, see Ciglaric et al. (2020). By using high performance, specialized GPU hardware and due to the propagation of higher CPU core counts we expect significant improvement in the computational speed in the coming years, further advancing the applicability of HMC to estimate DSGE models.

sampling. Potentially this improves the reliability of our estimation method in higher dimensional models considerably, as discussed by Betancourt (2018).[23]

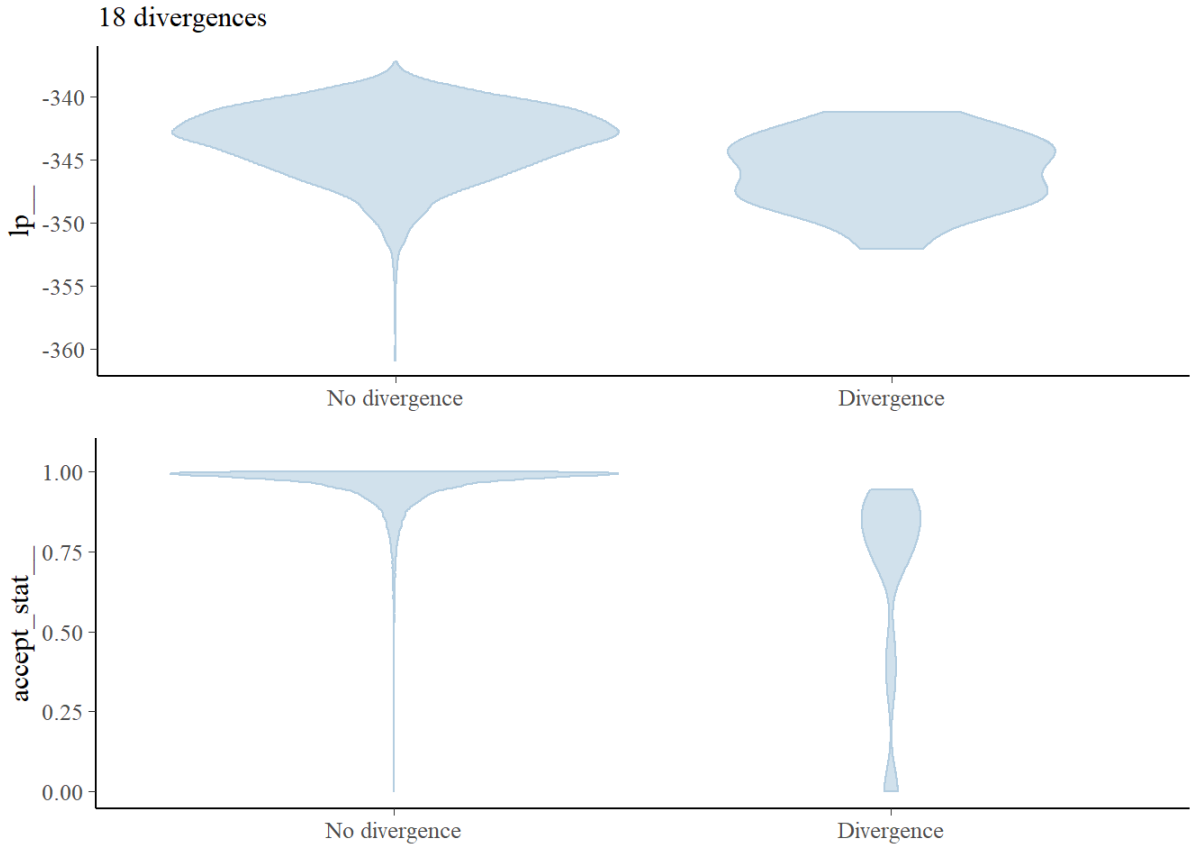
The second column of Table 1 reports the ratio of the Monte Carlo standard error of the mean (MCSE) to the posterior standard deviation (SD). The former is related to the accuracy of the simulation, the smaller the standard error, the closer the estimated parameter value is to the true value. The latter measures total uncertainty around the structural parameter. The ratio is considered to be small if it is below 5 percent, thus the values around 1 percent are indicative of a highly efficient sampling.

Turning to the diagnostics of the sampling, we start with the number and properties of divergent transitions.[24] As discussed, divergent transitions occur when the approximated path calculated by the leapfrog integrator drifts completely away from the original Hamiltonian path. Possible reasons could be a way too high curvature for the calibrated step size or other irregularities in the posterior likelihood as cliffs or funnels. The existence of divergent transitions is a warning sign, however, the rejected transitions might be also false positive. If they do not display a common pattern and are of a low proportion, they can be neglected. From 10,000 draws we observed 18 divergent iterations, approximately 0.2 percent. Due to their very low share and the lack of a systematic pattern we can state that the results are to be trusted. Figure 7 plots the frequency of divergent transitions against the log-posterior ($lp\_\_$) in the top panel, and the acceptance statistics ($accept\_stat\_\_$) in the bottom panel. 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

[23]We are confident that future research will highlight the advantages of HMC in large DSGE models with irregularly shaped posteriors.

[24]To visualize the diagnostics of the HMC method we used ShinyStan Version 3.0 (Gabry and Veen, 2020).

Figure 7: Small Scale DSGE Diagnostics: Divergence Information



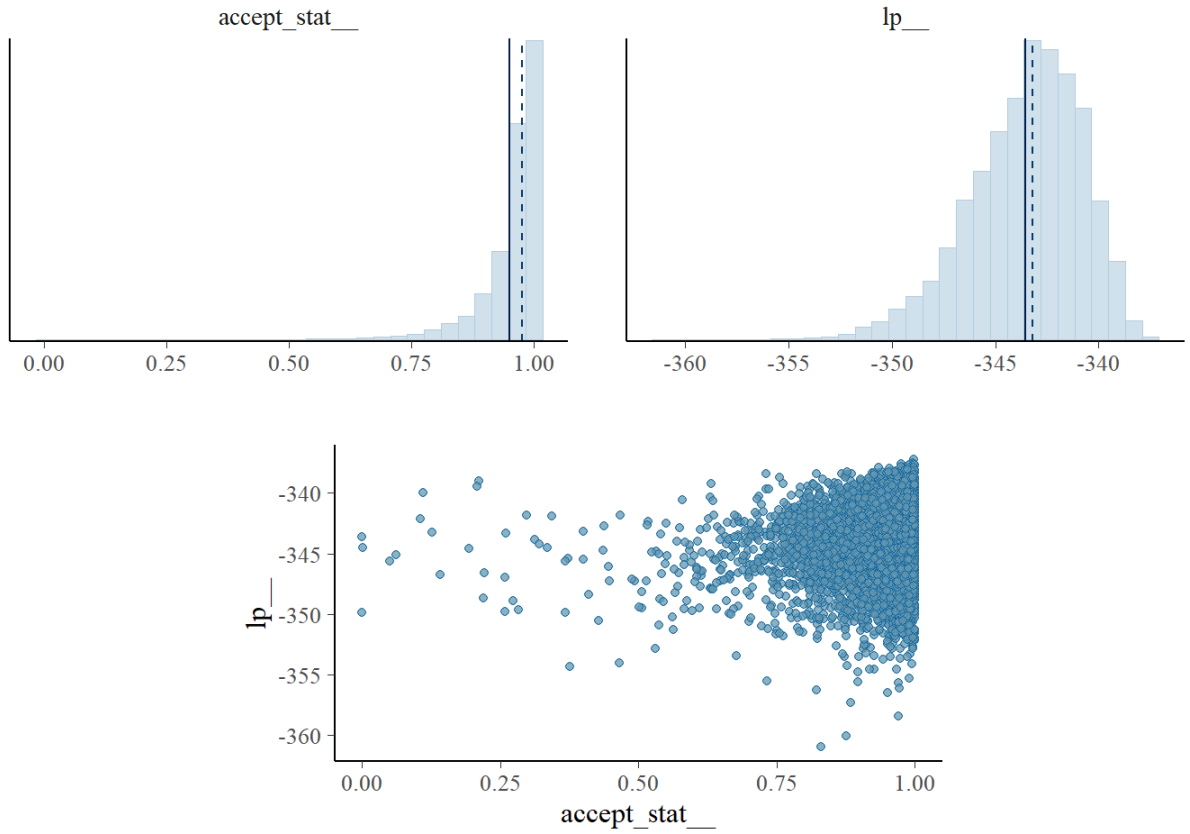
Notes: Plots of the divergent transitions (x-axis) against the log-posterior (y-axis top panel) and against the acceptance statistic (y-axis bottom panel) of the HMC sampling algorithm.

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 about which parts of the target distribution are difficult to sample from, albeit comparing the two charts, we can conclude that the sampler explored the difficult regions of the posterior. Turning to the bottom panel one might be cautious due to the high acceptance rate[25]. In general the intuition applies for HMC as well. If the acceptance rate is very high it might be indicative of inefficient sampling[26]. To reject this possibility we plot the marginal posterior distributions and the scatter plot of the acceptance rate and

[25]The acceptance rate refers to the intermediate Metropolis step in the HMC algorithm implemented in Stan.

[26]As discussed in the previous subsection Stan allows to refine the sampler and set the target Metropolis acceptance rate with a specific control option that adapts the step size based on the sampling during the burn in phase.

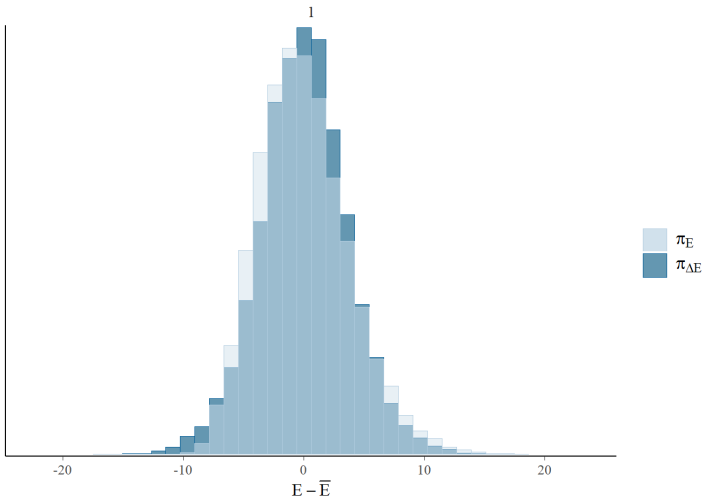
Figure 8: Small Scale DSGE Diagnostics: Acceptance Information



Notes: The figure plots the marginal posterior distribution of acceptance statistic (top left panel), 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 log-posterior in Figure 8. It shows no relationship between the acceptance rate and the log-posterior. In fact, it indicates that the posterior was adequately explored. This leads us to the discussion of the energy distribution to assess the robustness of the HMC algorithm, shown in Figure 9. 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 9 shows the reason for the low autocorrelation, and thus the high efficiency of the HMC algorithm, the energy levels, and with it the posterior-probability levels of the target distribution, being well explored.

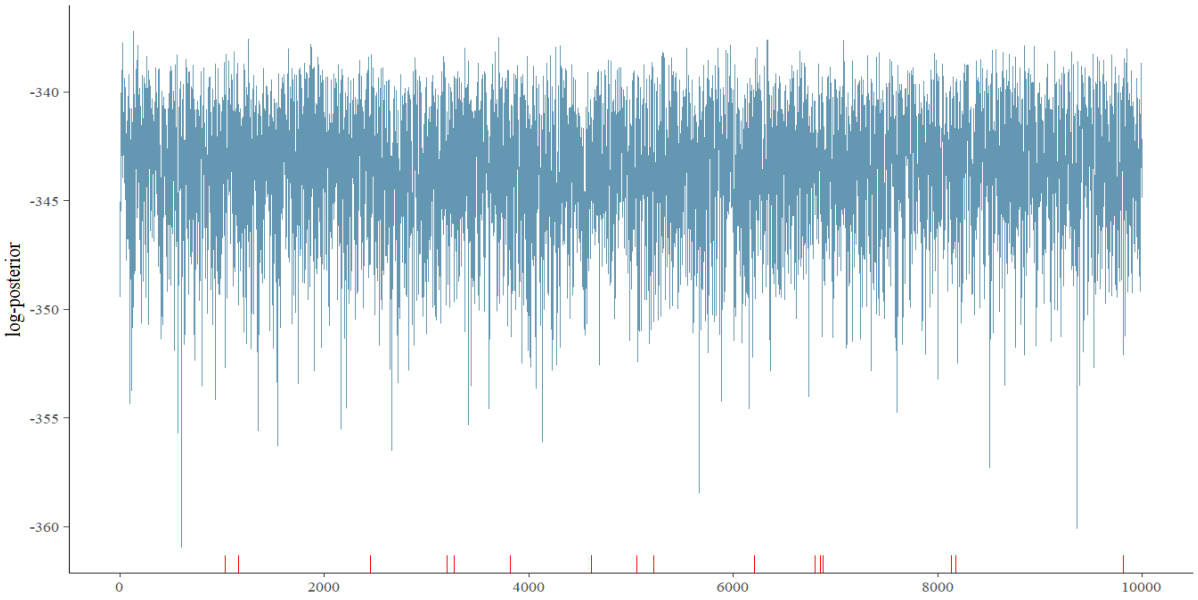
Figure 9: Small Scale DSGE Diagnostics: Energy Distribution



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 behavior. Figure 10 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. Hence, the chains are mixing sufficiently well.

Figure 10: 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 HMC are identical to those obtained with the RWMH algorithm.[27] This verifies the proper functioning of the algorithm and shows that in small DSGE models with simple target densities, RWMH sampling works sufficiently well.

Table 2: 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 HMC are based collectively on $N = 10{,}000$ draws from the posterior, obtained with 10 parallel chains, with a burn in of 500 draws and 1,000 sample draws for each. The results for the RWMH 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.

Now we turn to the detailed runtime analysis. As already pointed out, to ensure high-quality sampling provided by the HMC algorithm, the gradient has to be evaluated which requires additional computational time. To measure the overall gain in terms of effective sample size, we benchmark the algorithm against the readily available, commonly used software, Dynare. The standard mode search algorithm in Dynare, (`mode_compute=4`), did not manage to find an appropriate starting point for the RWMH algorithm. Therefore, we had to revert to the Monte Carlo based optimization routine (`mode_compute=6`). We draw

[27] 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).

again 100,000 samples from the posterior and omit the first 20,000 draws. More than half of the complete runtime was used for the mode search. We also run the HMC algorithm for a little less than the amount of time needed to execute the RWMH algorithm, that is, slightly less than 2 minutes 30 seconds. This resulted in 24,000 sample draws after a warm-up of 1,000 draws.[28] The advantage of our algorithm is that the verification whether the actual sets of parameters along the Hamiltonian path imply a unique stable solution can be switched off, if intended. In this case, the verification is not needed at all, as constraining the parameter space in a smart way – e.g. one does not allowed that the AR(1)-coefficients of the exogenous shocks become explosive and sets also the lower bound for the response to inflation in the monetary policy rule to above unity – renders the algorithm less complex and improves the runtime. For this small model the verification deteriorates the runtime by approximately 30 percent on the given hardware. However, there are indications that this loss might be in the ballpark of 10 percent for larger models or longer data series. The efficient samples sizes are provided in the Table 3. To ensure a coherent calculation of the effective sample size we stop considering autocorrelations at the point when the sum of the autocorrelations of two subsequent lags becomes negative to avoid noise. Table 3 shows that the effective sample sizes are far higher for the HMC algorithm than for the RWMH, for many of the parameters it is way beyond one order of magnitude, for some of them it is over 20 times higher. We acknowledge, however, that with increasing sample sizes the difference will shrink as for this particular case the mode search made up approximately 60 percent of the runtime. But even in that case the difference would be substantial.

We also benchmark the algorithm on simulated data based on the parameters obtained

[28]We ran both algorithms on the same desktop computer with an Intel 13900k CPU (24 cores/32 threads). We run both algorithms in their optimal environment, that is, Dynare in Windows and HMC in Linux.

Table 3: Posterior Estimates and Effective Sample Sizes – Small Scale DSGE Model

	HMC		RWMH	
Param.	Mean	ESS	Mean	ESS
τ	2.43	22112.2	2.39	1546.6
κ	0.85	14632.1	0.85	1090.9
ψ_1	1.95	11751.2	1.95	1198.6
ψ_2	0.61	18283.0	0.60	986.2
$r^{(A)}$	0.42	16115.0	0.40	1229.8
$\pi^{(A)}$	3.39	16748.1	3.41	1087.7
$\gamma^{(Q)}$	0.59	15237.4	0.60	1169.4
ρ_r	0.81	16692.5	0.80	1121.1
ρ_g	0.98	22235.8	0.98	1051.5
ρ_z	0.93	10949.4	0.93	1285.2
$100\sigma_r$	0.19	21306.4	0.20	1148.7
$100\sigma_g$	0.68	21857.7	0.68	1010.2
$100\sigma_z$	0.19	20616.7	0.19	1221.5

Notes: The table shows the posterior mean and the effective sample sizes for the HMC and the RWMH estimation, respectively. The results for HMC are based on 24,000 draws from the posterior after a burn in of 1,000 sample draws. The results for the RWMH algorithm are based on the authors own code in Dynare with 100,000 draws of which the first 20,000 draws were omitted. The runtime of the HMC algorithm was 141 seconds and that of the RWMH algorithm 153 seconds, where from the latter one 5 seconds were already deduced to account for the calculation of the results.

above with 200 observations. For this purpose, we change the prior of the parameter κ – which resulted in issues with the mode search in the above case – to a $\sim Beta(1.5, 1.5)$ distribution with a moderate curvature. We draw 100,000 samples with a burn-in of 20,000 draws with the RWMH while in slightly less time with HMC we drew 5,500 samples after a warm-up of 1,000 draws. The results are displayed in Table 4.

In this case HMC still samples far more efficiently than RWMH, however the difference is smaller, as expected, because the standard mode search algorithm implemented for the RWMH algorithm could be applied which takes only a few seconds to run. The HMC algorithm can still generate a larger effective sample size up to a factor of over 6. The results obtained from both algorithms correspond to each other, the differences are marginal. Compared with the parameter values used for the generation of the data the results obtained with both algorithms differ to a certain extent, however given that the

Table 4: Posterior Estimates and Effective Sample Sizes with Simulated Data – Small Scale DSGE Model

	HMC		RWMH	
Param.	Mean	ESS	Mean	ESS
τ	2.53	6882.5	2.53	1302.5
κ	0.78	3602.2	0.78	2063.6
ψ_1	1.60	3914.0	1.59	1239.0
ψ_2	0.55	4813.7	0.54	1107.1
$r^{(A)}$	0.35	2774.9	0.35	1840.9
$\pi^{(A)}$	3.25	2750.5	3.25	1471.1
$\gamma^{(Q)}$	0.50	2762.2	0.50	1355.0
ρ_r	0.76	6234.1	0.75	1288.3
ρ_g	0.97	4186.4	0.97	1413.3
ρ_z	0.89	4051.7	0.89	1287.1
$100\sigma_r$	0.20	6675.6	0.20	1080.4
$100\sigma_g$	0.69	6684.5	0.70	1277.1
$100\sigma_z$	0.22	6238.1	0.22	1291.5

Notes: The table shows the posterior mean and the effective sample sizes for the HMC and the RWMH estimation, respectively. The results for HMC are based on 5,500 sample draws from the posterior after a burn in of 1,000 draws. The results for the RWMH algorithm are based on the authors own code in Dynare with 100,000 draws of which the first 20,000 draws were omitted. The runtime of the HMC algorithm was 108 seconds and that of the RWMH algorithm 112 seconds, where from the latter one 5 seconds were already deduced to account for the calculation of the results.

data series consists only of 200 observations, this is little surprising.

3.3 Smets-Wouters Model

To explore the properties of HMC in a larger model we proceed with the estimation of the SW model which is a medium-scale closed economy DSGE model. It has become the standard workhorse model for economic policy analysis and served as a basis for newer generations of DSGE models. We estimated the model with US data for the period 1960Q1–2004Q4 using seven key macroeconomic variables: real GDP, real consumption, real investment, the GDP deflator, real wages, employment and the nominal short-term interest rate.[29] The model features a deterministic growth rate driven by labour-augmenting

[29]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. Originally in Smets and Wouters (2007) the model was estimated for the sample 1966Q1-

technology progress and is subject to nominal and real frictions. Nominal rigidities affect the labour and goods markets via Calvo pricing 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 a reaction to the inflation gap and the output gap.[30]

The exogenous variation of the model is driven by seven mostly AR(1) shock processes with estimated conditional variances: standard total factor productivity, monetary policy, investment specific technology, exogenous spending and risk premium and a wage and a price markup shock with an additional MA(1) structure.[31] The latter setup introduces anticipated news shocks for both the regular and the wage Phillips curve. The model is log-linearized around the steady state and net of deterministic growth rate.

We present the complete sampling diagnostics pertaining to the estimation of the SW model using HMC in the Appendix. The efficiency of the HMC algorithm is apparent. Even though we estimate the model with 1,000 draws only, it results in an effective sample size of 418.92 for the log-posterior. We compared the estimates with those obtained by the RWMH algorithm and present them in Table 5 and 6. We can conclude that both algorithms deliver similar results.[32] We failed to find any divergent transitions and the results together with the HMC diagnostics seem to indicate that the target density of the

2004Q4, however e.g. Cai et al. (2021) also estimated the model starting from 1960Q1. Johannes Pfeiffer's replication package written in Dynare contains data from 1947.

[30]The inflation gap is defined as the deviation from the steady state inflation and the output gap is the difference of output and its flexible price counterpart.

[31]The monetary policy shock is *iid*. 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.

[32]The runtime for the HMC algorithm with $N = 1{,}000$ draws from the posterior and a burn in of 500 draws on a AMD Ryzen 3950x (16 cores/32 threads) CPU is approx. 1.5 to 2 hours (including a tuning and warm-up time of 55-75 minutes). The runtime for the RWMH algorithm using Johannes Pfeiffer's replication files written in Dynare for two chains of 500,000 draws with a burn in of 100,000 draws on the same computer is approx. 45 minutes per chain.

Table 5: Posterior Estimates of the Smets-Wouters Model - Structural Parameters

	HMC		RWMH	
Parameter	Mean	[0.05, 0.95]	Mean	[0.05, 0.95]
φ	5.93	[4.38, 7.68]	5.90	[4.21, 7.56]
σ_c	1.41	[1.20, 1.66]	1.41	[1.16, 1.63]
h	0.73	[0.65, 0.80]	0.73	[0.66, 0.80]
ξ_w	0.75	[0.67, 0.84]	0.75	[0.67, 0.84]
σ_l	2.10	[1.21, 3.02]	2.12	[1.17, 3.03]
ξ_p	0.65	[0.56, 0.73]	0.65	[0.56, 0.73]
ι_w	0.56	[0.35, 0.77]	0.57	[0.36, 0.77]
ι_p	0.24	[0.11, 0.38]	0.23	[0.09, 0.36]
ψ	0.46	[0.30, 0.65]	0.46	[0.29, 0.64]
Φ	1.65	[1.53, 1.79]	1.64	[1.51, 1.77]
r_π	2.04	[1.77, 2.33]	2.04	[1.77, 2.30]
ρ	0.82	[0.77, 0.85]	0.82	[0.78, 0.86]
r_y	0.10	[0.07, 0.14]	0.10	[0.07, 0.14]
r_{dy}	0.21	[0.17, 0.25]	0.21	[0.16, 0.25]
$\bar{\pi}$	0.67	[0.50, 0.86]	0.67	[0.51, 0.83]
$100(\beta^{-1}-1)$	0.13	[0.07, 0.22]	0.14	[0.06, 0.21]
$\bar{l}$	0.88	[-0.68, 2.45]	0.85	[-0.69, 2.37]
$\bar{\gamma}$	0.47	[0.43, 0.49]	0.47	[0.43, 0.50]
α	0.21	[0.18, 0.23]	0.21	[0.18, 0.24]

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 HMC are based on $N = 1{,}000$ draws from the posterior and a burn in of 500 draws. The results for the RWMH algorithm are based on the authors' replication of the model using Johannes Pfeiffer's replication files written in Dynare with an acceptance rate of approximately 30.5%, two chains of 500,000 draws and a burn in of 100,000 draws.

SW model is well behaved. The application of the RWMH algorithm should be warranted as long as tight priors are assumed. However, a brief glance at the autocorrelation function of certain parameters suggests that the sample suffers from slight autocorrelation. As discussed, HMC is able to travel large distances in the parameter space and in the optimum one should obtain an uncorrelated sample for each parameter. Slight autocorrelation per se does not invalidate sampling, yet it is an indication for inefficient sampling for which feature the reason is not apparent at first sight.

An evident idea to identify the reason for inefficient sampling is to start the automatic tuning procedure from a different starting point than before. The gradient based

Table 6: Posterior Estimates of the Smets-Wouters Model - Shock Processes

	HMC		RWMH	
Parameter	Mean	[0.05, 0.95]	Mean	[0.05, 0.95]
σ_a	0.48	[0.43, 0.52]	0.48	[0.43, 0.52]
σ_b	0.24	[0.19, 0.28]	0.24	[0.19, 0.29]
σ_g	0.52	[0.47, 0.57]	0.52	[0.47, 0.56]
σ_I	0.46	[0.38, 0.54]	0.46	[0.38, 0.53]
σ_r	0.23	[0.21, 0.25]	0.23	[0.21, 0.25]
σ_p	0.13	[0.10, 0.16]	0.13	[0.10, 0.16]
σ_w	0.25	[0.21, 0.28]	0.25	[0.22, 0.28]
ρ_a	0.98	[0.97, 0.99]	0.98	[0.97, 0.99]
ρ_b	0.27	[0.12, 0.48]	0.28	[0.10, 0.46]
ρ_g	0.97	[0.96, 0.99]	0.97	[0.96, 0.99]
ρ_I	0.69	[0.59, 0.78]	0.69	[0.60, 0.79]
ρ_r	0.17	[0.07, 0.28]	0.17	[0.06, 0.27]
ρ_p	0.96	[0.92, 0.99]	0.96	[0.94, 0.99]
ρ_w	0.97	[0.94, 0.99]	0.97	[0.95, 0.99]
μ_p	0.80	[0.66, 0.90]	0.80	[0.70, 0.92]
μ_w	0.89	[0.82, 0.94]	0.89	[0.83, 0.95]
ρ_{ga}	0.57	[0.44, 0.70]	0.57	[0.44, 0.70]

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 HMC are based on $N = 1{,}000$ draws from the posterior and a burn in of 500 draws. The results for the RWMH algorithm are based on the authors' replication of the model using Johannes Pfeiffer's replication files written in Dynare with an acceptance rate of approximately 30.5%, two chains of 500,000 draws and a burn in of 100,000 draws.

approach is designed to navigate the chain to find the typical set. After starting the chain from a different spot the HMC algorithm indeed found a further mode in spite of setting tight priors and using the data sample 1960Q1-2004Q4. Cai et al. (2021) have already documented that even if tight priors, identical to Smets and Wouters (2007), are set, yet a shorter sample from 1960Q1 to 1991Q3 is used, it will lead to identification problems as the data does not contain sufficient information to pin down properly a handful of parameters. While Cai et al. (2021) document that the parameters h (habit persistence in consumption), ι_p (degree of price indexation), ρ_p (persistence of price markup shock) and ρ_{ga} (loading of government spending on technology shock innovations) exhibit a multimodal pattern, our results in Table 7 suggest that several other parameters differ as

Table 7: Posterior Estimates of the Smets-Wouters Model - Alternative Mode vs. Original Mode

	Alternative Mode	Original Mode		Alternative Mode	Original Mode
	Mean [0.05, 0.95]	Mean [0.05, 0.95]		Mean [0.05, 0.95]	Mean [0.05, 0.95]
φ	5.43 [3.76, 7.17]	5.93 [4.38, 7.68]	α	0.21 [0.18, 0.24]	0.21 [0.18, 0.23]
σ_c	1.41 [1.18, 1.64]	1.41 [1.20, 1.66]	σ_a	0.48 [0.44, 0.53]	0.48 [0.43, 0.52]
h	0.69 [0.58, 0.77]	0.73 [0.65, 0.80]	σ_b	0.21 [0.15, 0.26]	0.24 [0.19, 0.28]
ξ_w	0.80 [0.73, 0.87]	0.75 [0.67, 0.84]	σ_g	0.52 [0.47, 0.56]	0.52 [0.47, 0.57]
σ_l	2.13 [1.19, 3.12]	2.10 [1.21, 3.02]	σ_I	0.45 [0.37, 0.53]	0.46 [0.38, 0.54]
ξ_p	0.80 [0.75, 0.85]	0.65 [0.56, 0.73]	σ_r	0.23 [0.21, 0.25]	0.23 [0.21, 0.25]
ι_w	0.52 [0.33, 0.72]	0.56 [0.35, 0.77]	σ_p	0.21 [0.19, 0.24]	0.13 [0.10, 0.26]
ι_p	0.31 [0.17, 0.48]	0.24 [0.11, 0.38]	σ_w	0.23 [0.20, 0.27]	0.25 [0.21, 0.28]
ψ	0.40 [0.24, 0.56]	0.46 [0.30, 0.65]	ρ_a	0.97 [0.96, 0.98]	0.98 [0.97, 0.99]
Φ	1.63 [1.50, 1.77]	1.65 [1.53, 1.79]	ρ_b	0.41 [0.19, 0.68]	0.27 [0.12, 0.48]
r_π	1.97 [1.68, 2.26]	2.04 [1.77, 2.33]	ρ_g	0.97 [0.96, 0.99]	0.97 [0.96, 0.99]
ρ	0.85 [0.82, 0.88]	0.82 [0.77, 0.85]	ρ_I	0.71 [0.61, 0.80]	0.69 [0.59, 0.78]
r_y	0.13 [0.08, 0.17]	0.10 [0.07, 0.14]	ρ_r	0.13 [0.05, 0.22]	0.17 [0.07, 0.28]
r_{dy}	0.22 [0.18, 0.27]	0.21 [0.17, 0.25]	ρ_p	0.93 [0.89, 0.96]	0.96 [0.92, 0.99]
$\bar{\pi}$	0.67 [0.51, 0.83]	0.67 [0.50, 0.86]	ρ_w	0.97 [0.93, 0.99]	0.97 [0.94, 0.99]
$\bar{\beta}$	0.14 [0.07, 0.24]	0.13 [0.07, 0.22]	μ_p	0.98 [0.97, 0.99]	0.80 [0.66, 0.90]
$\bar{l}$	0.61 [-0.92, 2.05]	0.88 [-0.68, 2.45]	μ_w	0.91 [0.85, 0.95]	0.89 [0.82, 0.94]
$\bar{\gamma}$	0.46 [0.43, 0.49]	0.47 [0.43, 0.49]	ρ_{ga}	0.57 [0.45, 0.70]	0.57 [0.44, 0.70]

Notes: The table shows the posterior mean and the 5 and 95 percentile of the posterior from the HMC estimation. The results for HMC are based on $N = 1{,}000$ draws from the posterior and a burn in of 500 draws. $\bar{\beta}$ is defined as $100(\beta^{-1} - 1)$, similarly to the baseline estimation.

well from those documented in the baseline estimation. In addition to further parameters related to price and wage setting, e.g. ξ_p, ξ_w, σ_p, μ_p, there seems to be a significant difference in the estimates for φ (investment adjustment costs), ψ (capacity utilization costs) and ρ_b (persistence of risk premium shock), at least if compared with the differences in the estimates for the rest of the parameters.[33] We also double checked our results obtained with HMC by estimating the model using RWMH in Dynare and starting the estimation from the mean of the parameter estimates from the alternative mode obtained with HMC. The results obtained are fairly similar, see Appendix. Finally, in some cases we observed

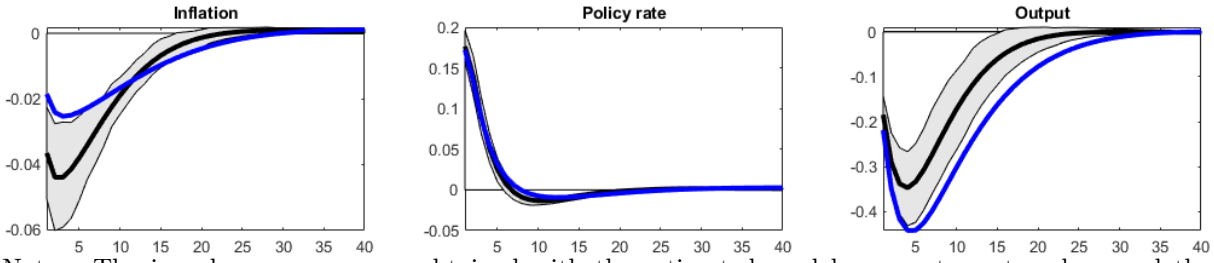
[33]The results also hold up if we estimate the model on the original sample 1966Q1-2004Q4 used in Smets and Wouters (2007).

that using RWMH and running two parallel chains one of the chains swaps the mode and remains there for the rest of the time. However, estimates obtained by averaging over the entire sample from both chains are likely to be biased and therefore more sophisticated methods are required. In general, it is noteworthy though that improved HMC diagnostics could contribute to uncover irregularities in the posterior likelihood.

Policy implications

In this section we use the estimated modes to analyze the impulse responses to structural shocks. We start by answering the question whether the two modes deliver different monetary policy transmission as this model serves as the core of policy models widely applied by central banks. Figure 11 plots the impulse responses of inflation, the policy rate and output to a positive unanticipated monetary policy shock. Starting with the original mode, we recover the result in Smets and Wouters (2007), that monetary policy leads to a hump-shaped fall in inflation and output, both peaking at business cycle frequency, 5 quarters, and dying out after 20 quarters. In contrast, the impulse response functions based on the estimates at the alternative mode show that the same shock has half the impact on inflation and a third more on output. Furthermore, we wish to highlight that the impulse responses of inflation and output feature higher persistence and lie outside the 90 percent credible interval around the original mode. Hence, they are distinct in line with the bimodality documented before. Figure 12 plots the impulse responses of the same key variables to a positive price markup shock. It shows that a markup shock at the alternative mode, dominated by its higher persistence and more pronounced anticipatory character, turns deflationary after the initial impact, to which monetary policy responds with an easing. The impulse response functions illustrate the different character of the SW model at the two modes and highlight that the original results could have overstated

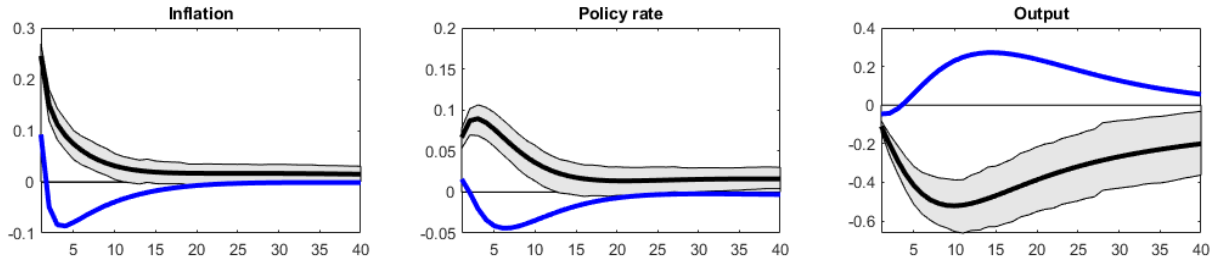
Figure 11: Monetary Policy Shock



Notes: The impulse responses are obtained with the estimated model parameters at and around the original mode and at the alternative, second mode. The figures plot the impulse response functions to a monetary policy shock at the original mode in black together with the 5th and 95th percentiles of the credible interval (CI) around it, based on the implicit assumption that only one mode exists. The impulse response functions at the alternative mode are shown in blue where the magnitude of the standard deviation of the shock was taken over from that in the original mode.

the impact of monetary policy. Figure 13 summarizes the historical contribution of the

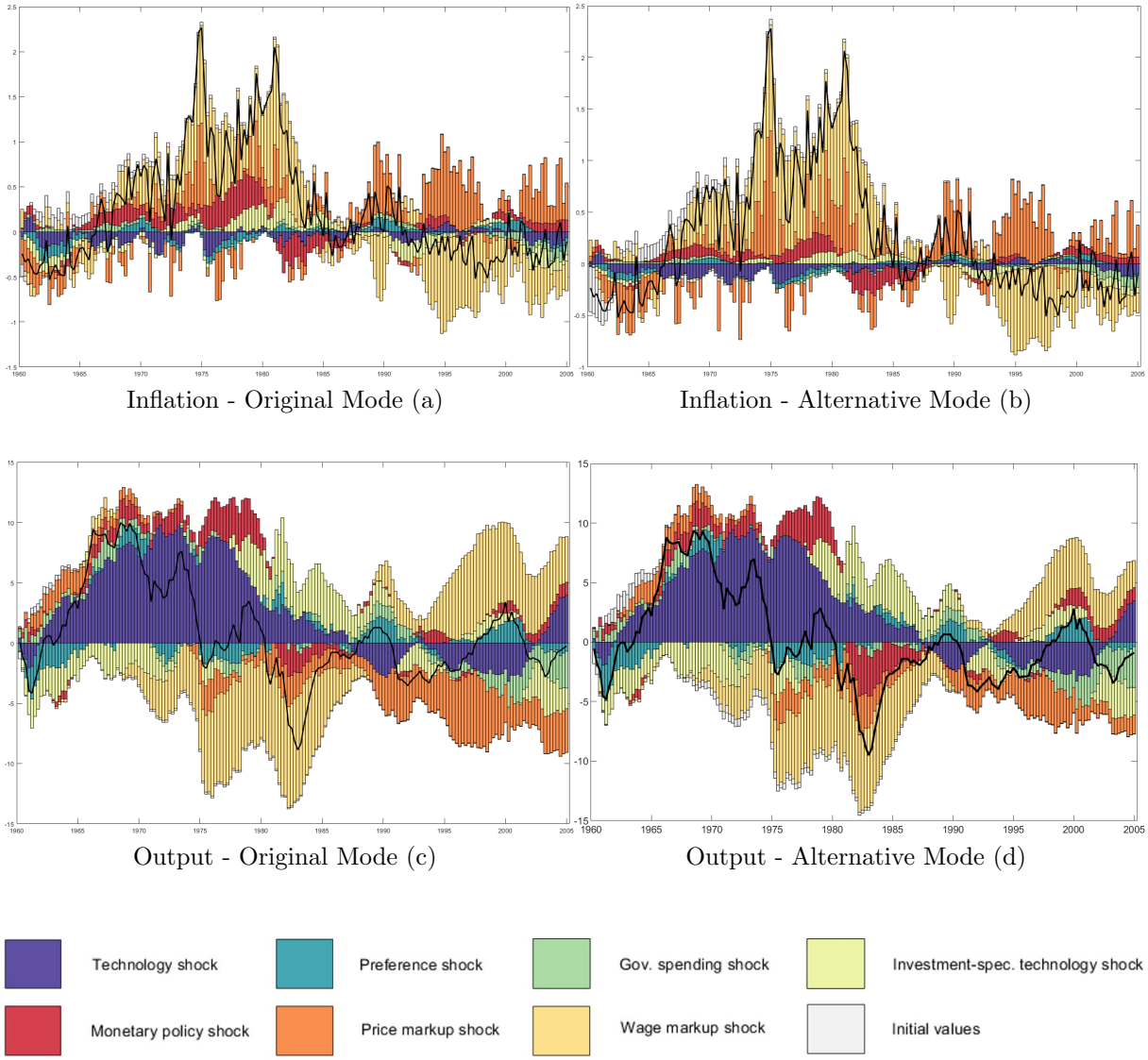
Figure 12: Price Markup Shock



Notes: The impulse responses are obtained with the estimated model parameters at and around the original mode and at the alternative, second mode. The figures plot the impulse response to a price markup shock at the original mode in black together with the 5th and 95th percentiles of the credible interval (CI) around it, based on the implicit assumption that only one mode exists. The impulse response function at the alternative mode is shown in blue where the magnitude of the standard deviation of the shock was taken over from that in the original mode.

various structural shocks to inflation and output. Chart (a) and (b) reflect the reduced role of monetary policy and the increased role of markup shocks for inflation in the seventies at the alternative mode. The alternative mode associates also diminished roles to the markup shocks in the run up of the Great Recession. Chart (c) and (d) compare the historical variance contributions of structural shocks to output at the original and the alternative modes. The two charts paint a similar picture, except for the role of the markup shocks past 1990, from where on in the alternative mode the contribution by both markup shocks is smaller.

Figure 13: Historical Variance Decompositions - Original vs. Alternative Mode



Inflation - Original Mode (a)

Inflation - Alternative Mode (b)

Output - Original Mode (c)

Output - Alternative Mode (d)

Notes: Chart (a) and (b) compare the historical variance decomposition of inflation in the original and in the alternative mode while chart (c) and (d) that of output. Source: authors' calculations.

4 Extension: Sequential Hamiltonian Monte Carlo

A main disadvantage of the HMC algorithm is that it fails to explore multimodal posterior distributions, see e.g. Lan et al. (2014). To demonstrate this in a DSGE environment, we borrow the stylized model from Herbst and Schorfheide (2014) in its original form:

$$y_t = [1\ 1]\, s_t, \qquad s_t = \begin{bmatrix} \phi_1 & 0 \\ \phi_3 & \phi_2 \end{bmatrix} s_{t-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \epsilon_t \qquad \epsilon_t \overset{iid}{\sim} N(0,1) \tag{7}$$

The structural parameters of the model to be estimated, θ_1 and θ_2, are mapped into the reduced form parameters $\phi = [\phi_1, \phi_2, \phi_3]$ as follows:

$$\phi_1 = \theta_1^2, \quad \phi_2 = (1 - \theta_1^2), \quad \phi_3 = \theta_1\theta_2 \text{ with } 0 < \theta_1, \theta_2 < 1.$$

As discussed in Herbst and Schorfheide (2014), this setup creates a global identification problem. The data generated by the parameter set $\tilde{\theta}_1 = \sqrt{1 - \theta_1^2}$ and $\tilde{\theta}_2 = \theta_1\theta_2/\tilde{\theta}_1$ is observationally equivalent to the data generated by $\theta = [\theta_1, \theta_2]$. Setting $\theta = [0.55, 0.55]$ results in an observationally equivalent parameter vector of approximately $\tilde{\theta} = [0.84, 0.36]$ where HMC mixes well. However, a larger distance and thereby veining support between the two modes render HMC mixing poor. Setting $\theta = [0.45, 0.45]$ causes the chain stuck either in the latter mode or in the observationally equivalent one, $\tilde{\theta} = [0.89, 0.23]$. We also observed that notwithstanding starting from $\tilde{\theta}$, the chain jumped sometimes to the other mode, θ, during the warm-up and remained there for the rest of the simulation.

A further interesting experiment, which addresses criticism with respect to the original estimation setup of the SW model, was carried out in Herbst and Schorfheide (2014). They unrestrict the model by using uninformative or less informative priors with a larger variance for the parameters instead of the tight priors set in Smets and Wouters (2007). They allow thereby for the information to obtain a larger weight. Herbst and Schorfheide (2014) reports a bimodal marginal posterior density for a handful of parameters once the less restrictive prior setting is used in which case widely used MCMC based samplers as the RWMH algorithm do not mix properly. Instead, they get stuck in one of the modes, depending on the starting point of the chain. To remedy issues with multimodality several algorithms have already been proposed, e.g. Neal (2001), Liu and Chen (1998), Gilks and Berzuini (2002) and Del Moral et al. (2006). These papers 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 pioneering work by introducing the SMC algorithm to DSGE models to remedy multimodality issues. The proposed SMC framework therein fits into the scheme described by Chopin (2004) and is essentially a sequential importance sampler. In each step the posterior density $p(Y|\theta)^{\beta_n}p(\theta)$ at stage n, at which the likelihood is weighted with $0 \leq \beta_n \leq 1 \; \forall n \in \{1, ..., N\}$, 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 commonly referred to as likelihood tempering.[34] Roughly speaking, at each stage n the importance weights $\{W_j^{(n)}\}_{j=1}^J$ for all parameter draws $\{\theta_j^{(n)}\}_{j=1}^J$, that is, the fraction of the posterior densities at stage $n+1$ and n equaling to $p(Y|\theta^{(n)})^{\beta_{n+1}-\beta_n}$, are multiplied with the weights from the stage before and then normalized to serve as the weights for the importance sampling.[35] 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*. At each stage the draws are 'mutated', that means a new proposal is potentially accepted when applying a MH-step, which is alternatively referred to as 'rejuvenation'. Once the variance of the weights becomes large the draws are resampled using the calculated weights which are reset then to unity.

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, the position of each particle, $\theta_j^{(n)}$, will be updated only at each fourth stage on average. The intuition behind likelihood

[34] Alternatively, one can also carry out data tempering by increasing the number of observations included to calculate the likelihood function at each stage $n \in \{1, ..., N\}$.

[35] For further details on the SMC algorithm we refer to Chopin (2004) and Herbst and Schorfheide (2014).

tempering is that decreasing β_n, the weight of the likelihood function in the posterior density, reduces the energy barrier between distant separated modes enabling commonly applied MCMC samplers to move between modes. However, this feature can be only exploited if the algorithm is able to traverse large distances in the parameter space. As the HMC algorithm is capable of proposing updates θ' to the current draws θ which are distant and in theory always accepted, it can 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 applying 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 approaching infinity this bias will have to disappear even if particles were not rejuvenated at all, see e.g. the annealed importance sampling by Neal (2001), as asymptotic convergence of these algorithms is warranted. However, with increasing amount of parameters the number of particles necessary will increase exponentially. Therefore, a guided approach to rejuvenate the actual parameter draws might be of an advantage. The Sequential Hamiltonian Monte Carlo (SHMC) 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 use the SMC framework proposed by Herbst and Schorfheide (2014) with both multinomial and stratified resampling[36]. Algorithm 4 summarizes the main steps.

[36]For a detailed description of stratified resampling see Herbst and Schorfheide (2015). The aim of this resampling scheme is to reduce the variation in the resampling step and to guarantee that particles with high weights will always be resampled.

Algorithm 4 Sequential Hamiltonian Monte Carlo

1. Search for the different modes by starting the HMC algorithm from different parameter settings.
2. Specify a sequence $\{\beta_n\}_{n=0}^{N}$ with $1 = \beta_N > ... > \beta_{n+1} > \beta_n > ... > \beta_0 \geq 0$ so that the resulting bridging densities $p(Y|\theta^{(n)})^{\beta_n} p(\theta^{(n)})$ are not 'too different'
3. Tune the HMC sampler for each target density $p(Y|\theta^{(n)})^{\beta_n} p(\theta^{(n)})$ separately, depending also on the possible position $\theta_j^{(n)}$ of a given particle $\{\theta_j^{(n)}, W_j^{(n)}\}$ 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 on the current position $\theta_j^{(n)}$ of a given particle $\{\theta_j^{(n)}, W_j^{(n)}\}$.

The algorithm fits into the scheme proposed by Chopin (2004), as already pointed out by Daviet (2018).[37] Therefore, under common regularity conditions and assuming that multinomial resampling is used, almost sure convergence will hold:

$$\frac{1}{J}\sum_{j=1}^{J} h(\theta_j^{(n)}) \xrightarrow{a.s.} \mathbb{E}_{\tilde{\pi}_n}(h)$$
$$\frac{\sum_{j=1}^{J} w_j^{(n)} h(\theta_j^{(n)})}{\sum_{j=1}^{J} w_j^{(n)}} \xrightarrow{a.s.} \mathbb{E}_{\pi_n}(h)$$
$$\frac{1}{J}\sum_{j=1}^{J} h(\hat{\theta}_j^{(n)}) \xrightarrow{a.s.} \mathbb{E}_{\pi_n}(h)$$

where $\tilde{\pi}_n(\cdot) := \int \pi_{n-1}(\theta^{(n-1)}) k^{(n)}(\theta^{(n-1)}, \cdot) d\theta^{(n-1)}$ with $k^{(n)}$ being the stochastic kernel density function implied by the HMC algorithm. Furthermore, $\pi_n(\theta^{(n)}) = \frac{1}{Z_n} p(Y|\theta^{(n)})^{\beta_n} p(\theta^{(n)})$, $w_j^{(n)} \propto \nu_j^{(n)} = \pi_n(\theta_j^{(n-1)}) / \tilde{\pi}_n(\theta_j^{(n-1)})$ and $\hat{\theta}_j^{(n)}$ are 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 is:

$$J^{1/2}\left\{\frac{1}{J}\sum_{j=1}^{J} h(\hat{\theta}_j^{(n)}) - \mathbb{E}_{\pi_n}(h)\right\} \xrightarrow{D} \mathcal{N}(0, \hat{V}_n(h)) \quad \forall n = 1, ..., N \tag{8}$$

[37] For a complete proof of the central limit theorem for a large class of SMC methods see Chopin (2004), while for DSGE models Herbst and Schorfheide (2014) provides an adjusted proof based on Chopin (2004).

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

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

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

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

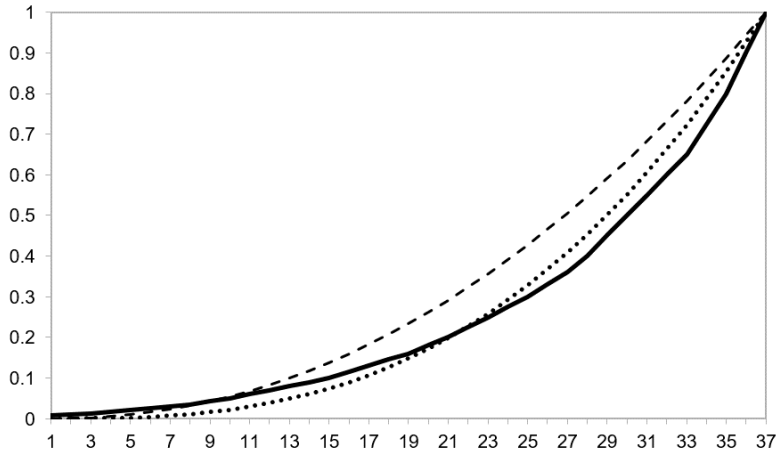
$$\hat{V}_n(h) = V_n(h) + \text{Var}_{\pi_n}(h) \quad \forall n = 1, ..., N$$

To apply the algorithm we estimate again the SW model and loosen the priors in line with Herbst and Schorfheide (2014). We use the same data set as for the estimation of the restricted model. Before executing the algorithm the sampler has to be tuned. We use $N = 37$ stages and $J = 512$ particles not to waste computational resources, which amount is low if compared with SMC frameworks using RWMH for rejuvenation.[38] Our aim is to implement the algorithm in a way that it can be run on stock desktop computers.[39] The tempering schedule $\{\beta_n\}_{n=1}^N$ was calibrated in a fashion that $p(Y|\theta)^{\beta_n}p(\theta)$ serves always sufficiently well as a proposal density for $p(Y|\theta)^{\beta_{n+1}}p(\theta)$, hence the bridge densities are never 'too different'. Even with such a 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. Figure 14 compares the tempering schedules: the solid line shows the tempering schedule used for our SHMC framework and the dashed line the original one from Herbst and Schorfheide (2014) if applied to $N = 37$ stages. 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 framework. Using the

[38] For example Herbst and Schorfheide (2014) uses $J = 12,000$ particles and $N = 500$ stages.

[39] In this setup, the runtime of the algorithm amounts to approximately one day on a stock desktop computer equipped with an AMD Ryzen 3950x (16 cores/32 threads) CPU.

Figure 14: 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 $\lambda = 2.75$.

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 notice that already at relatively low β_n levels mixing is not optimal in the sense that it does not switch often enough between modes. As β_n increases, less information can be extracted from the density with respect to 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 $\pi_0 := p(Y|\theta)^{\beta_0}p(\theta)$ with $\beta_0 = 0.005$, where the HMC sampler is able to mix between the modes. $\beta_0 = 0.005$ implies that the latter initial distribution contains approximately the same amount of information which would be provided by one single data point from the data sample 1960Q1-2004Q4 used for this estimation. To create our initial sample, we simulated 10,000 sample draws from π_0. Afterwards we thinned the obtained sample and used each consecutive tenth sample draw. In our initial samples

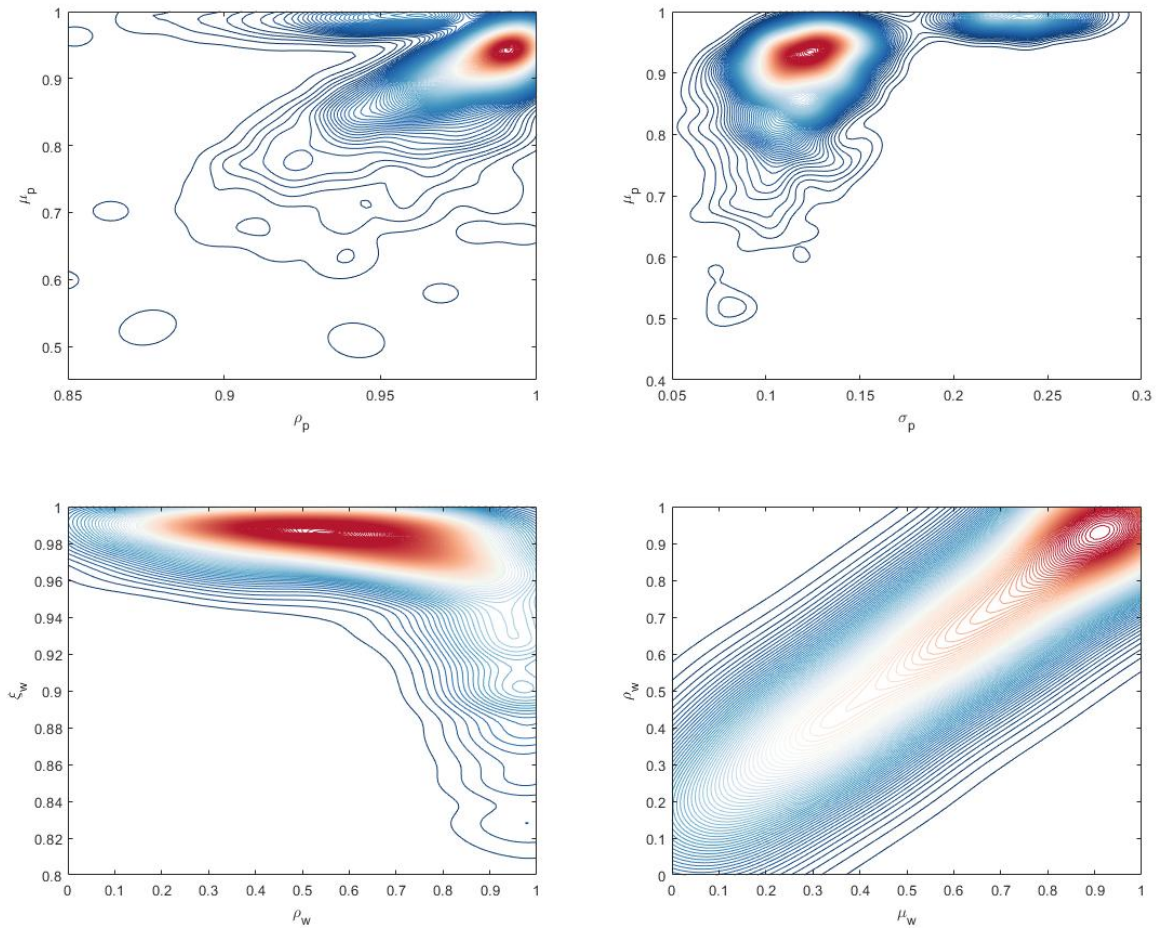
obtained approximately 14 percent of the particles were from the region around the mode which seems to be dominated and to encompass less volume. The latter setup appears to be also in line with existing final results from the literature, see e.g Herbst and Schorfheide (2014) and Lanne and Luoto (2018). The latter work augments the SMC algorithm with non-sequential importance sampling and finds that the portion of the particles stemming from the region around the dominated mode amounts to 20 percent while Herbst and Schorfheide (2014) reports approximately 5 percent. 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, our algorithm resamples mostly at each second stage.[40]

We performed the estimation using both multinomial and stratified resampling. Our estimation results suggest that the posterior density for a handful of parameters is ill-behaved. 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 in Figure 15.[41] In our estimations we obtain that on average approximately 12 percent of the particles are concentrated in the area around the dominated mode if multinomial resampling is applied while the latter share amounts to approximately 9.3 percent if stratified resampling is used. Hence, based on the average the probability that the data was generated by the parameter estimates from the dominated mode is roughly 10 percent which result lies between the shares reported in Herbst and Schorfheide (2014) and Lanne and Luoto (2018), 5 percent and 20 percent, respectively. Furthermore, we also observe similar bimodality for the joint density of the MA(1) term and the standard deviation of

[40] Alternatively, we could have resampled deterministically at each second stage as a resampling at the third consecutive stage occurred rather rarely.

[41] For the scatter plots with the sample draws see Appendix.

Figure 15: Sequential Hamiltonian Monte Carlo - Joint Posterior Kernel Density Estimates



Notes: The plot shows the joint posterior kernel density estimates of the following parameters: $[\rho_p,\mu_p]$ (upper left), $[\sigma_p,\mu_p]$ (upper right), $[\rho_w,\xi_w]$ (lower left) and $[\mu_w,\rho_w]$ (lower right). Sample size equals to 1,024, where two sample draws of the size $J = 512$, respectively, were merged, the first obtained by applying multinomial resampling, the second one by stratified resampling. The divergence rate at the last stage, $\beta_N = 1$, was approximately 0.2 percent and 0.4 percent, respectively, while the overall divergence rate throughout all $N = 37$ stages amounted to approximately 3.8 and 3.7 percent for both samples. Scatter plots with the original sample draws are shown in the Appendix. Source: authors' simulations and calculations.

the error term, σ_p. The parameters determining the wage Phillips curve, ξ_w and ρ_w, the wage rigidity parameter and the AR(1) term of the markup shock process also exhibit 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 markup 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 implies that the markup shock process is overparameterized as also suggested by Lanne and Luoto

(2018)[42]. It is worth to note that the divergence rate at the last stage, when $\beta_N = 1$, was approximately 0.2 percent and 0.4 percent, respectively, while the overall divergence rate throughout all $N = 37$ stages amounted to approximately 3.8 and 3.7 percent for both samples. The divergence rate for low values of β_n was relatively higher implying that the posterior densities at low β_n values exhibit irregularities and might be challenging to sample from. However, it might be very difficult to remedy the latter issue as these irregularities could be also caused by the energy barrier between two modes.

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 in line with existing literature.

5 Conclusion and Outlook

In this paper we review the benchmark DSGE estimation framework and present an advanced alternative, the HMC sampler, which is widely used in other fields of academic research. Subsequently, we implement the HMC 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 use the HMC to estimate a small-scale three-equation textbook New Keynesian model and the SW model. We find that in particular cases the HMC algorithm provides a twenty-fold improvement in speed over the RWMH algorithm. Our estimation results correspond to those in the existing literature, which verifies the accuracy of the estimation method used and the algorithm implemented. In addition, we present the sampling diagnostics in detail, which enables us to conclude that the target density of the three-equation textbook model exhibits a regular shape.

[42] Lanne and Luoto (2018) also reported that restricting the model could result in an improved fit.

We confirm that the RWMH algorithm also operates adequately in this case. For the SW model, however, we found a second mode after the advanced HMC diagnostics exhibited signs of inefficient sampling, even though this estimation was carried out using the original priors and data set. We therefore wish to highlight the fact that advanced sampling diagnostics for the HMC helps identify irregularities in the posterior and parameters that are difficult to sample. Furthermore, we would like to stress that the HMC does not require any posterior mode search, which is a major benefit when compared with the commonly used RMWH algorithm. Finally, we combine the HMC with the SMC algorithm to address a key shortcoming of the HMC, namely that it fails to explore ill-behaved posterior densities properly. We apply this extended framework to estimate the SW model using less informative priors and obtain bimodal posterior densities, whose results are consistent with those in the existing literature. However, we acknowledge that there is scope to improve the runtime of the algorithm.

Given our results, we believe that the superior sampling properties and diagnostics unique to the HMC will provide new opportunities to revisit existing DSGE model estimation exercises. Furthermore, its ability to cope with high-dimensional distributions should make it possible to directly sample the state vector. This would enable us to estimate the model without applying the Kalman filter and hence use shocks other than normally distributed ones. We are also hopeful that additional sophisticated gradient-based estimation methods optimally capable of handling multimodality – such as the wormhole HMC – can be implemented for DSGE estimation as the computational power available today should render these algorithms feasible. At the same time, it should be noted that the latter approach would provide only a numerical solution to multimodality – similarly to the SHMC algorithm – although this issue is of a fundamental nature. We leave these

topics for future research.

References

An, Sungbae and Frank Schorfheide, "Bayesian Analysis of DSGE Models," *Econometric Reviews*, 2007, *26*, 113–172. https://doi.org/10.1080/07474930701220071.

Anderson, G., "Solving Linear Rational Expectations Models: A Horse Race," *Computational Economics*, 2008, *31*, 95–113. https://doi.org/10.1007/s10614-007-9108-0.

Anderson, Gary S., "A reliable and computationally efficient algorithm for imposing the saddle point property in dynamic models," *Journal of Economic Dynamics and Control*, 2010, *34*, 472–489. https://doi.org/10.1016/j.jedc.2009.10.004.

Besag, J., "Discussion of the Paper by Grenander and Miller," *Journal of the Royal Statistical Society A*, 1994, *56*, 591–592.

Betancourt, M., "A Conceptual Introduction to Hamiltonian Monte Carlo," *arXiv preprint*, 2018, *1701.02434v2*. https://arxiv.org/abs/1701.02434.

_ , "Identity Crisis," 2020. https://betanalpha.github.io/assets/case$_$studies/identifiability.html.

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*, 877–888. https://doi.org/10.1017/S0266466600006307.

Blanchard, O. J. and C. M. Kahn, "The Solution of Linear Difference Models under Rational Expectations," *Econometrica*, 1980, *48*, 1305–1311. https://doi.org/10.2307/1912186.

Brooks, S. P. and A. Gelman, "General Methods for Monitoring Convergence of Iterative Simulations," *Journal of Computational and Graphical Statistics*, 1998, *7*, 434–455. https://doi.org/10.2307/1390675.

Cai, Michael, Marco Del Negro, Edward Herbst, Ethan Matlin, Reca Sarfati, and Frank Schorfheide, "Online estimation of DSGE models," *The Econometrics Journal*, 2021, *24*, C33–C58. https://doi.org/10.1093/ectj/utaa029.

Calvo, G. A., "Staggered Prices in a Utility-Maximizing Framework," *Journal of Monetary Economics*, 1983, *12*, 383–398. https://doi.org/10.1016/0304-3932(83)90060-0.

Carpenter, Bob, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell, "Stan: A Probabilistic Programming Language," *Journal of Statistical Software*, 2017, *76*, 1–32. https://doi.org/10.18637/jss.v076.i01.

Chib, Siddhartha and Srikanth Ramamurthy, "Tailored randomized block MCMC methods with application to DSGE models," *Journal of Econometrics*, 2010, *155*, 19–38. https://doi.org/10.1016/j.jeconom.2009.08.003.

Chopin, Nicolas, "Central Limit Theorem for Sequential Monte Carlo Methods and Its Application to Bayesian Inference," *The Annals of Statistics*, 2004, *32*, 2385–2411. https://doi.org/10.1214/009053604000000698.

Christiano, L. J., "Solving Dynamic Equilibrium Models by a Method of Undetermined Coefficients," *Computational Economics*, 2002, *20*, 21–55. https://doi.org/10.1023/A:1020534927853.

_ , **M. Eichenbaum, and C. L. Evans**, "Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy," *Journal of Political Economy*, 2005, *113*, 1–45. https://doi.org/10.1086/426038.

Ciglaric, T., R. Cesnovar, and E. Strumbelj, "Automated OpenCL GPU kernel fusion for Stan Math," *IWOCL '20: International Workshop on OpenCL, Conference Paper, Article No.:14*, 2020. https://doi.org/10.1145/3388333.3388654.

Clarida, Richard, Jordi Gali, and Mark Gertler, "The Science of Monetary Policy: A New Keynesian Perspective," *Journal of Economic Literature*, December 1999, *37*, 1661–1707. https://doi.org/10.1257/jel.37.4.1661.

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

Cúrdia, V. and R. Reis, "Correlated Disturbances and U.S. Business Cycles," *NBER Working Papers 15774*, 2010. https://doi.org/10.3386/w15774.

Daviet, R., "Inference with Hamiltonian Sequential Monte Carlo Simulators," *arXiv*, 2018, *1812.07978v1*. https://arxiv.org/abs/1812.07978.

Del Moral, P., A. Doucet, and A. Jasra, "Sequential Monte Carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006, *68*, 411–436. https://doi.org/10.1111/j.1467-9868.2006.00553.x.

Duane, Simon, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth, "Hybrid Monte Carlo," *Physics Letters B*, 1987, *195*, 216–222. https://doi.org/10.1016/0370-2693(87)91197-X.

Durmus, A., E. Moulines, and E. Saksman, "On the convergence of Hamiltonian Monte Carlo," *arXiv preprint*, 2019, *arXiv:1705.00166v2.* https://arxiv.org/abs/1705.00166.

Fernandez-Villaverde, J. and J. F. Rubio-Ramirez, "Estimating Macroeconomic Models: A Likelihood Approach," *Review of Economic Studies*, 2007, *74*, 1059–1087. https://doi.org/10.1111/j.1467-937X.2007.00437.x.

_ , J. Rubio-Ramirez, 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.

Fernandez-Villaverde, Jesus and Pablo A. Guerron-Quintana, "Estimating DSGE Models: Recent Advances and Future Challenges," *Annual Review of Economics*, 2021, *13*, 229–252. https://doi.org/10.1146/annurev-economics-081020-044812.

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

_ , D. Simpson, A. Vehtari, M. Betancourt, and A. Gelman, "Visualization in Bayesian workflow (with discussion)," *Journal of the Royal Statistical Society A*, 2019, *182*, 389–402. https://doi.org/10.1111/rssa.12378.

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*, 127–146. https://doi.org/10.1111/1467-9868.00280.

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, *29*, 1073–1098. https://doi.org/10.1002/jae.2397.

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

_ **and** _ , "Tempered Particle filtering," *Journal of Econometrics*, 2019, *210*, 26–44. https://doi.org/10.1016/j.jeconom.2018.11.003.

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*, 1593–1623.

Iskrev, Nikolay, "Local identification in DSGE models," *Journal of Monetary Economics*, 2010, *57*, 189–202. https://doi.org/10.1016/j.jmoneco.2009.12.007.

Kim, Jinill, "Constructing and estimating a realistic optimizing model of monetary policy," *Journal of Monetary Economics*, 2000, *45*, 329–359. https://doi.org/10.1016/S0304-3932(99)00054-9.

King, R. G. and M. W. Watson, "The Solution of Singular Linear Di§erence Systems Under Rational Expectations," *International Economic Review*, 1998, *39*, 1015–1026. https://doi.org/10.2307/2527350.

Klein, P., "Using the generalized Schur form to solve a multivariate linear rational expectations model," *Journal of Economic Dynamics and Control*, 2000, *24*, 1405–1423. https://doi.org/10.1016/S0165-1889(99)00045-7.

Komunjer, I. and S. Ng, "Dynamic Identification of Dynamic Stochastic General Equilibrium Models," *Econometrica*, 2011, *79*, 1995–2032. https://doi.org/10.3982/ECTA8916.

Kydland, F. E. and E. C. Prescott, "Time to Build and Aggregate Fluctuations," *Econometrica*, 1982, *50*, 1345–1370. https://doi.org/10.2307/1913386.

Lambert, B., *A Student's Guide to Bayesian Statistics*, SAGE Publications Ltd, 2018.

Lan, H. and A. Meyer-Gohde, "Solvability of perturbation solutions in DSGE models," *Journal of Economic Dynamics and Control*, 2014, *45*, 366–388. https://doi.org/10.1016/j.jedc.2014.06.005.

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

Lanne, M. and J. Luoto, "Data-Driven Identification Constraints for DSGE Models," *Oxford Bulletin of Economics and Statistics*, 2018, *80*, 236–258. https://doi.org/10.1111/obes.12217.

Liu, J. S. and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, 1998, *93*, 1032–1044. https://doi.org/10.1080/01621459.1998.10473765.

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

Mackenze, P. B., "An improved hybrid Monte Carlo method," *Physics Letters B*, 1989, *226*, 369–371. https://doi.org/10.1016/0370-2693(89)91212-4.

Neal, R. M., "Annealed importance sampling," *Statistics and Computing*, 2001, *11*, 125–139. https://doi.org/10.1023/A:1008923215028.

_ , "Slice Sampling," *Annals of Statistics*, 2003, *31*, 705–767. https://doi.org/10.1214/aos/1056562461.

_ , "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*, 61–92. https://doi.org/10.1016/S0304-3932(00)00052-0.

Qi, Y. and T. Minka, "Hessian-based Markov Chain Monte-Carlo Algorithms," in "Proceedings of the First Cape Cod Workshop on Monte Carlo Methods" September 2002.

Roberts, G. O. and O. Stramer, "Langevin Diffusions and Metropolis-Hastings Algorithms," *Methodology and Computing in Applied Probability*, 2002, *4*, 337–357. https://doi.org/10.1023/A:1023562417138.

_ **and R. L. Tweedie**, "Exponential Convergence of Langevin Distributions and Their Discrete Approximations," *Bernoulli*, 1996, *2*, 341–363. https://doi.org/10.2307/3318418.

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. https://doi.org/10.1214/aoap/1034625254.

Rotemberg, J. J., "Aggregate Consequences of Fixed Costs of Price Adjustment," *American Economic Review*, 1983, *73* (3), 433–436.

Schorfheide, F., "Loss function-based evaluation of DSGE models," *Journal of Applied Econometrics*, 2000, *15* (6), 645–670. https://doi.org/10.1002/jae.582.

Sims, C. A., "Solving Linear Rational Expectations Models," *Computational Economics*, 2002, *20*, 1–20. https://doi.org/10.1023/A:1020517101123.

Smets, F. and R. Wouters, "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 2007, *97*, 586–606. https://doi.org/10.1257/aer.97.3.586.

Uhlig, H., "A Toolkit for Analyzing Nonlinear 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. https://doi.org/10.1016/j.aml.2009.01.012.

A Small Scale New Keynesian Model

The model can be summarized by the following equations:[43]

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

$$\hat{\pi}_t = \beta\mathbb{E}_t[\hat{\pi}_{t+1}] + \kappa(\hat{y}_t - \hat{g}_t) \tag{A.2}$$

$$\hat{R}_t = \rho_R\hat{R}_{t-1} + (1-\rho_R)\psi_1\hat{\pi}_t + (1-\rho_R)\psi_2(\hat{y}_t - \hat{g}_t) + \epsilon_{R,t} \tag{A.3}$$

$$\hat{g}_t = \rho_g\hat{g}_{t-1} + \epsilon_{g,t} \tag{A.4}$$

$$\hat{z}_t = \rho_g\hat{z}_{t-1} + \epsilon_{z,t} \tag{A.5}$$

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:

$$YGR_t = \gamma^{(Q)} + 100(\hat{y}_t - \hat{y}_{t-1} + \hat{z}_t) \tag{A.6}$$

$$INFL_t = \pi^{(A)} + 400\hat{\pi}_t \tag{A.7}$$

$$INT_t = \pi^{(A)} + 4\gamma^{(Q)} + 400\hat{R}_t \tag{A.8}$$

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

$$\theta = [\tau, \kappa, \psi_1, \psi_2, \rho_r, \rho_g, \rho_z, \sigma_r, \sigma_g, \sigma_z, r^A, \pi^A, \gamma^Q] \tag{A.9}$$

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

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

Table A.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).

B Smets-Wouters Model: Alternative Mode Comparison

Table B.1: Posterior Estimates of the Smets-Wouters Structural Parameters - Alternative Mode

	Hamiltonian Monte Carlo		Random Walk Metropolis Hastings	
Parameter	Mean	[0.05, 0.95]	Mean	[0.05, 0.95]
φ	5.43	[3.76, 7.17]	5.46	[3.67, 7.23]
σ_c	1.41	[1.18, 1.64]	1.40	[1.17, 1.64]
h	0.69	[0.58, 0.77]	0.69	[0.60, 0.79]
ξ_w	0.80	[0.73, 0.87]	0.80	[0.74, 0.87]
σ_l	2.13	[1.19, 3.12]	2.14	[1.20, 3.04]
ξ_p	0.80	[0.75, 0.85]	0.80	[0.75, 0.85]
ι_w	0.52	[0.33, 0.72]	0.52	[0.32, 0.72]
ι_p	0.31	[0.17, 0.48]	0.31	[0.16, 0.46]
ψ	0.40	[0.24, 0.56]	0.40	[0.23, 0.56]
Φ	1.63	[1.50, 1.77]	1.63	[1.50, 1.76]
r_π	1.97	[1.68, 2.26]	1.96	[1.68, 2.23]
ρ	0.85	[0.82, 0.88]	0.85	[0.82, 0.89]
r_y	0.13	[0.08, 0.17]	0.13	[0.08, 0.17]
r_{dy}	0.22	[0.18, 0.27]	0.22	[0.18, 0.26]
$\bar{\pi}$	0.67	[0.51, 0.83]	0.67	[0.51, 0.83]
$100(\beta^{-1}-1)$	0.14	[0.07, 0.24]	0.14	[0.06, 0.23]
$\bar{l}$	0.61	[-0.92, 2.05]	0.56	[-0.92, 2.00]
$\bar{\gamma}$	0.46	[0.43, 0.49]	0.46	[0.44, 0.50]
α	0.21	[0.18, 0.24]	0.21	[0.18, 0.24]

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 HMC are based on $N = 1{,}000$ draws from the posterior and a burn in of 500 draws. The results for the RWMH algorithm are based on the authors' replication of the model using Johannes Pfeiffer's replication files using Dynare with an acceptance rate of 31.2%, two chains of 500,000 draws and a burn in of 100,000 draws. Thus the resulting number of draws is 800,000.

Table B.2: Posterior Estimates of the Smets-Wouters Model's Shock Processes - Alternative Mode

	Hamiltonian Monte Carlo		Random Walk Metropolis Hastings	
Parameter	Mean	[0.05, 0.95]	Mean	[0.05, 0.95]
σ_a	0.48	[0.44, 0.53]	0.48	[0.43, 0.53]
σ_b	0.21	[0.15, 0.26]	0.21	[0.14, 0.27]
σ_g	0.52	[0.47, 0.56]	0.52	[0.47, 0.56]
σ_I	0.45	[0.37, 0.53]	0.45	[0.38, 0.52]
σ_r	0.23	[0.21, 0.25]	0.23	[0.21, 0.25]
σ_p	0.21	[0.19, 0.24]	0.21	[0.18, 0.24]
σ_w	0.23	[0.20, 0.27]	0.23	[0.20, 0.27]
ρ_a	0.97	[0.96, 0.98]	0.97	[0.96, 0.99]
ρ_b	0.41	[0.19, 0.68]	0.41	[0.17, 0.69]
ρ_g	0.97	[0.96, 0.99]	0.97	[0.96, 0.99]
ρ_I	0.71	[0.61, 0.80]	0.71	[0.61, 0.81]
ρ_r	0.13	[0.05, 0.22]	0.13	[0.04, 0.22]
ρ_p	0.93	[0.89, 0.96]	0.93	[0.89, 0.96]
ρ_w	0.97	[0.93, 0.99]	0.97	[0.94, 0.99]
μ_p	0.98	[0.97, 0.99]	0.98	[0.97, 1.00]
μ_w	0.91	[0.85, 0.95]	0.91	[0.86, 0.96]
ρ_{ga}	0.57	[0.45, 0.70]	0.57	[0.44, 0.70]

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 HMC are based on $N = 1{,}000$ draws from the posterior and a burn in of 500 draws. The results for the RWMH algorithm are based on the authors' replication of the model using Johannes Pfeiffer's replication files using Dynare with an acceptance rate of 31.2%, one two chains of 500,000 draws and a burn in of 100,000 draws. Thus the resulting number of draws is 800,000.

Figure B.1: Historical Variance Decomposition of Inflation in the Smets-Wouters Model at the Original Mode

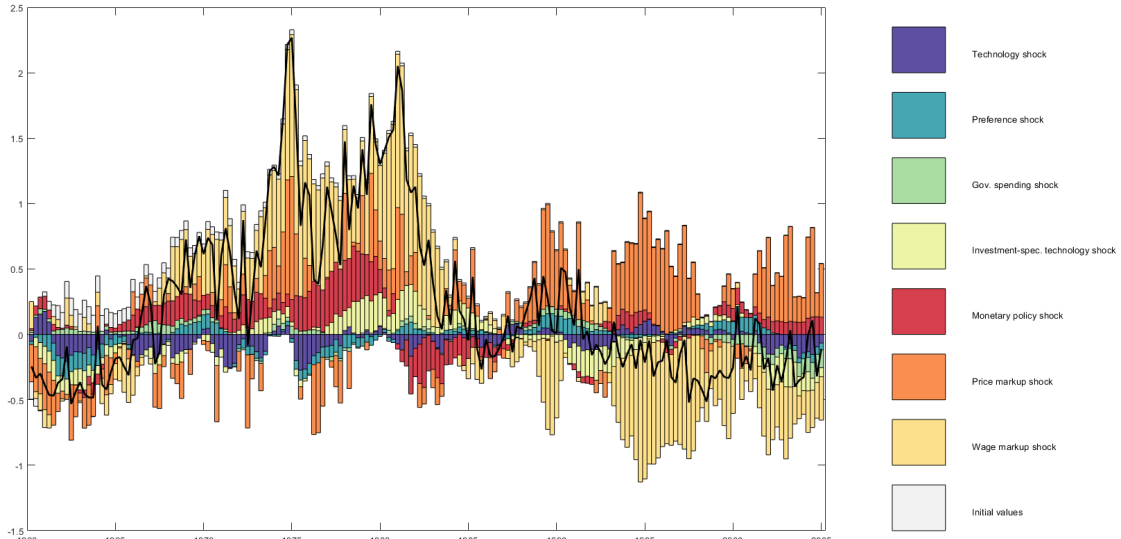


Figure B.2: Historical Variance Decomposition of Inflation in the Smets-Wouters Model at the Alternative Mode

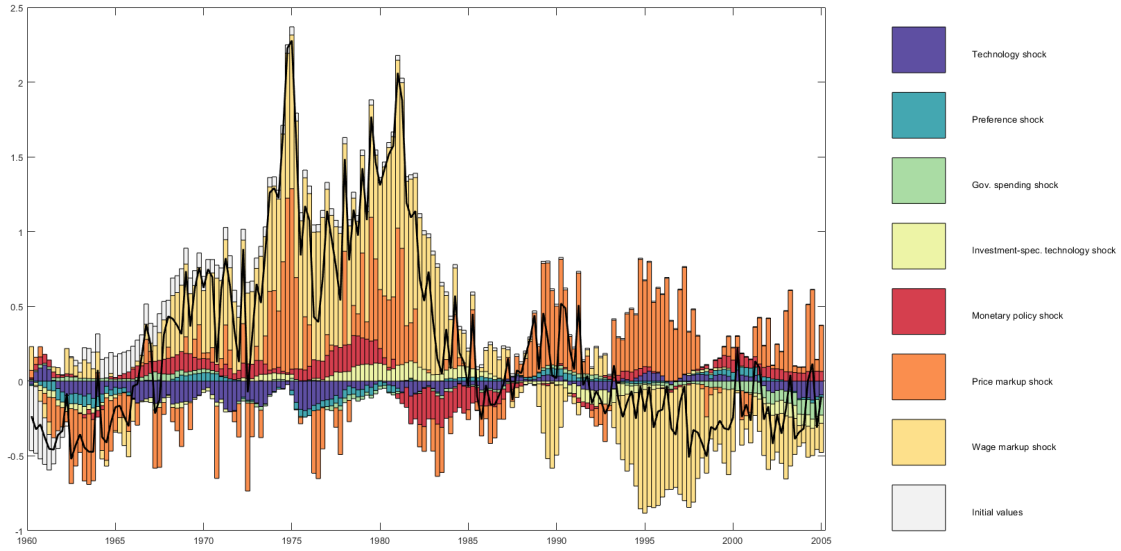


Figure B.3: Historical Variance Decomposition of Output in the Smets-Wouters Model at the Original Mode

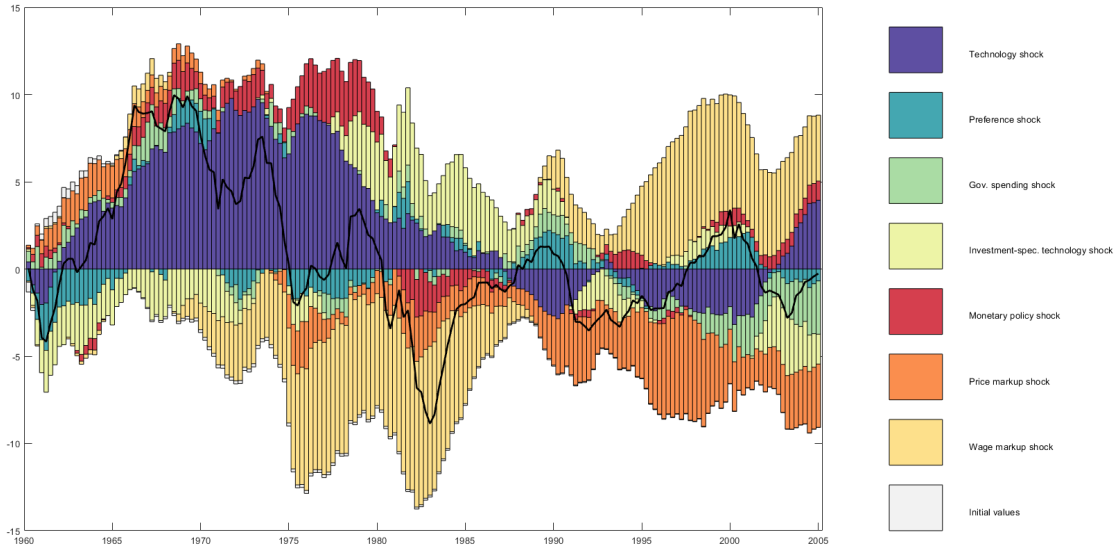
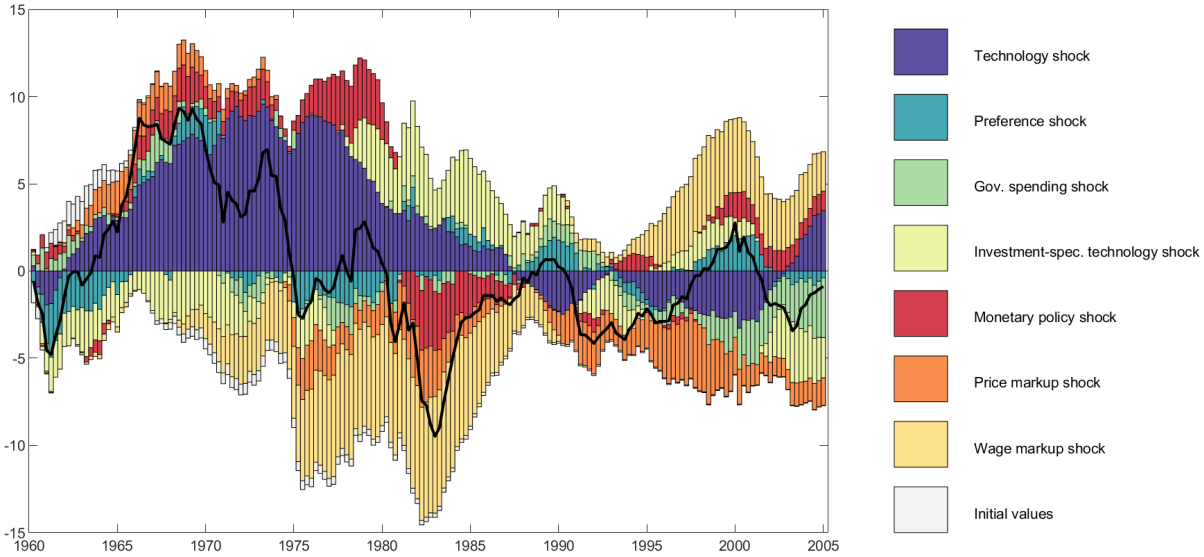


Figure B.4: Historical Variance Decomposition of Output in the Smets-Wouters Model at the Alternative Mode



C Smets-Wouters Model: Hamiltonian Monte Carlo Estimation Diagnostics

Smets-Wouters Model: Diagnostics for Original Mode Estimation

Warnings

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

Numerical diagnostics

	n_eff	Rhat	mean	se_mean	sd	
log-posterior	418.92	1.00	-1191.70	0.22	4.59	
crpi	1163.52	1.00	2.04	0.00	0.17	
crdy	1765.21	1.00	0.21	0.00	0.03	
cry	941.05	1.00	0.10	0.00	0.02	
crr	995.36	1.01	0.82	0.00	0.02	
constelab	1377.36	1.00	0.88	0.03	0.95	
constepinf	1071.22	1.00	0.67	0.00	0.10	
ctrend	497.64	1.00	0.47	0.00	0.02	
constebeta	1084.42	1.00	0.13	0.00	0.05	
cgy	1285.63	1.00	0.57	0.00	0.08	
cmaw	1015.16	1.00	0.89	0.00	0.04	
cmap	532.57	1.01	0.80	0.00	0.08	
calfa	1438.32	1.00	0.21	0.00	0.02	
czcap	950.76	1.00	0.46	0.00	0.10	

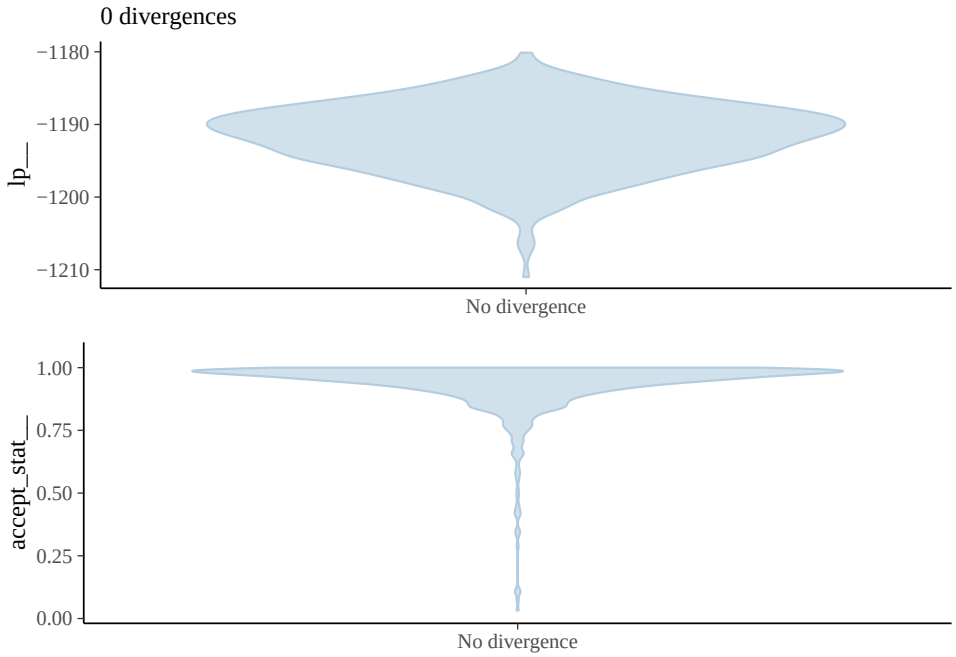
csadjcost	892.60	1.00	5.93	0.03	1.01
csigma	730.47	1.00	1.42	0.01	0.14
chabb	1010.00	1.00	0.73	0.00	0.04
cfc	846.70	1.00	1.65	0.00	0.08
cindw	1275.43	1.00	0.57	0.00	0.13
cprobw	632.40	1.00	0.75	0.00	0.05
cindp	867.03	1.00	0.23	0.00	0.09
cprobp	433.91	1.00	0.65	0.00	0.05
csigl	1523.58	1.00	2.10	0.01	0.55
crhoa	754.90	1.00	0.98	0.00	0.01
crhob	498.21	1.00	0.27	0.00	0.11
crhog	1115.33	1.00	0.97	0.00	0.01
crhoqs	775.30	1.00	0.69	0.00	0.06
crhoms	1309.81	1.00	0.17	0.00	0.06
crhopinf	370.24	1.00	0.96	0.00	0.02
crhow	415.38	1.00	0.97	0.00	0.01
sigmaea	952.76	1.00	0.48	0.00	0.03
sigmaeb	665.30	1.00	0.24	0.00	0.03
sigmaeg	1332.55	1.00	0.52	0.00	0.03
sigmaeqs	988.35	1.00	0.46	0.00	0.05
sigmaem	1140.80	1.00	0.23	0.00	0.01
sigmaepinf	865.82	1.00	0.13	0.00	0.02
sigmaew	1255.44	1.00	0.25	0.00	0.02

Visual diagnostics

Divergence Information

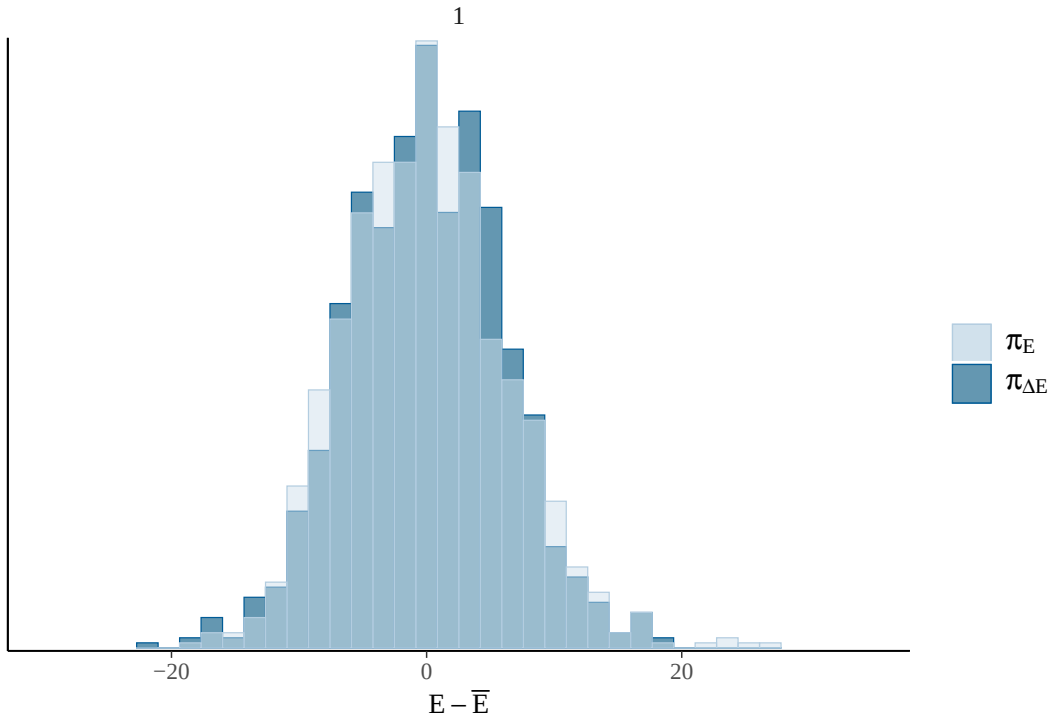
These are plots of the *divergent transition status* (x-axis) against the *log-posterior* (y-axis top panel) and against the *acceptance statistic* (y-axis bottom panel) of the sampling algorithm for all chains. Divergent transitions can indicate problems for the validity of the results. A good plot 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 show 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.

Figure C.1: Log-posterior and acceptance statistics



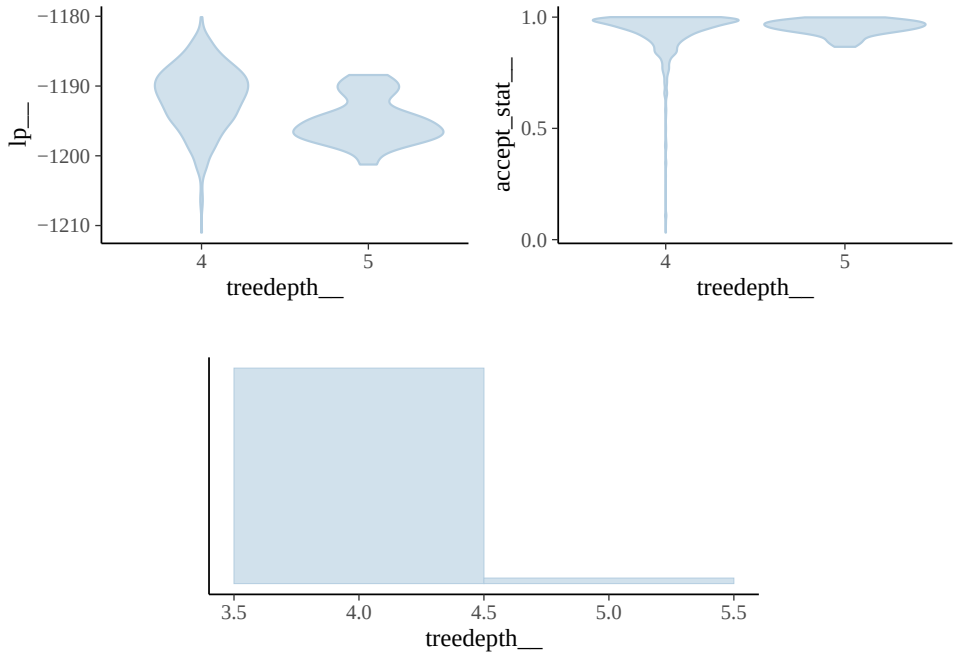
Energy

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.



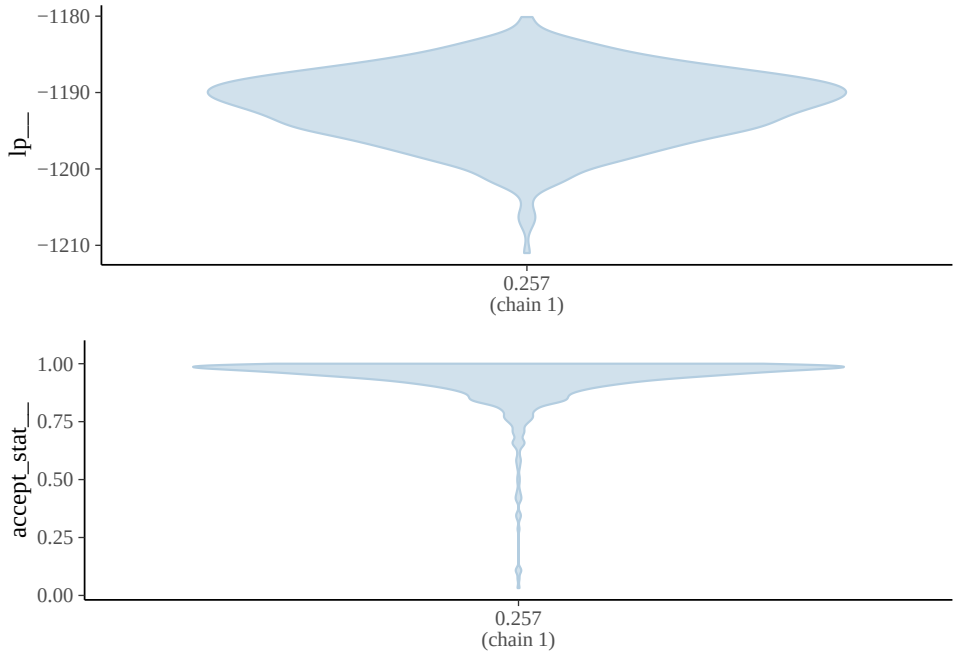
Treedepth Information

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 warm-up time, the latter might be mitigated by reparameterization. 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.



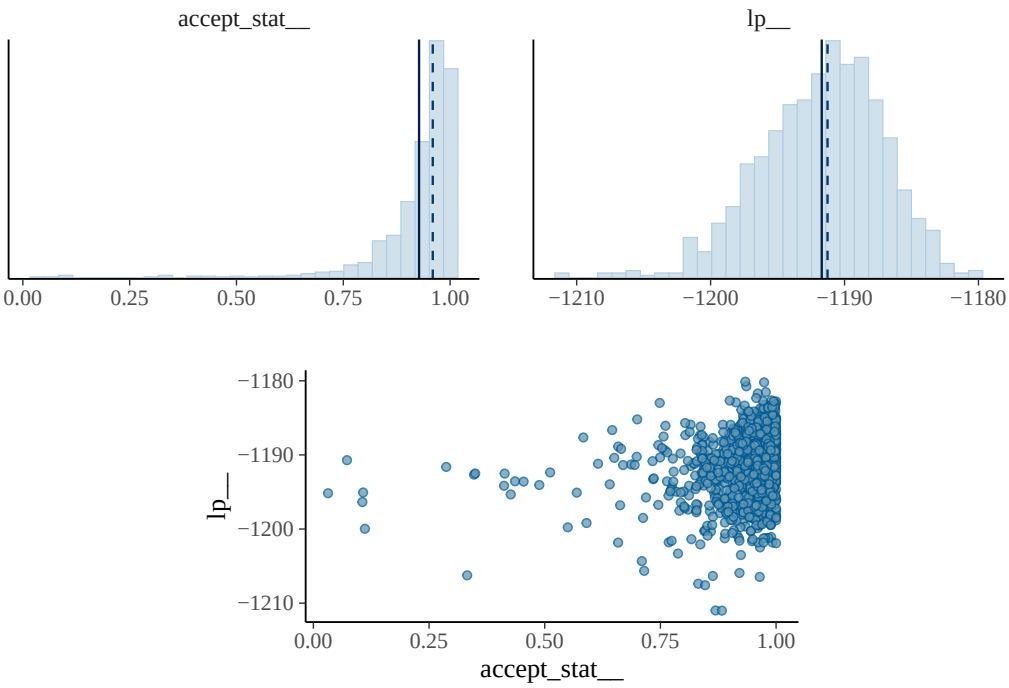
Step Size Information

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.



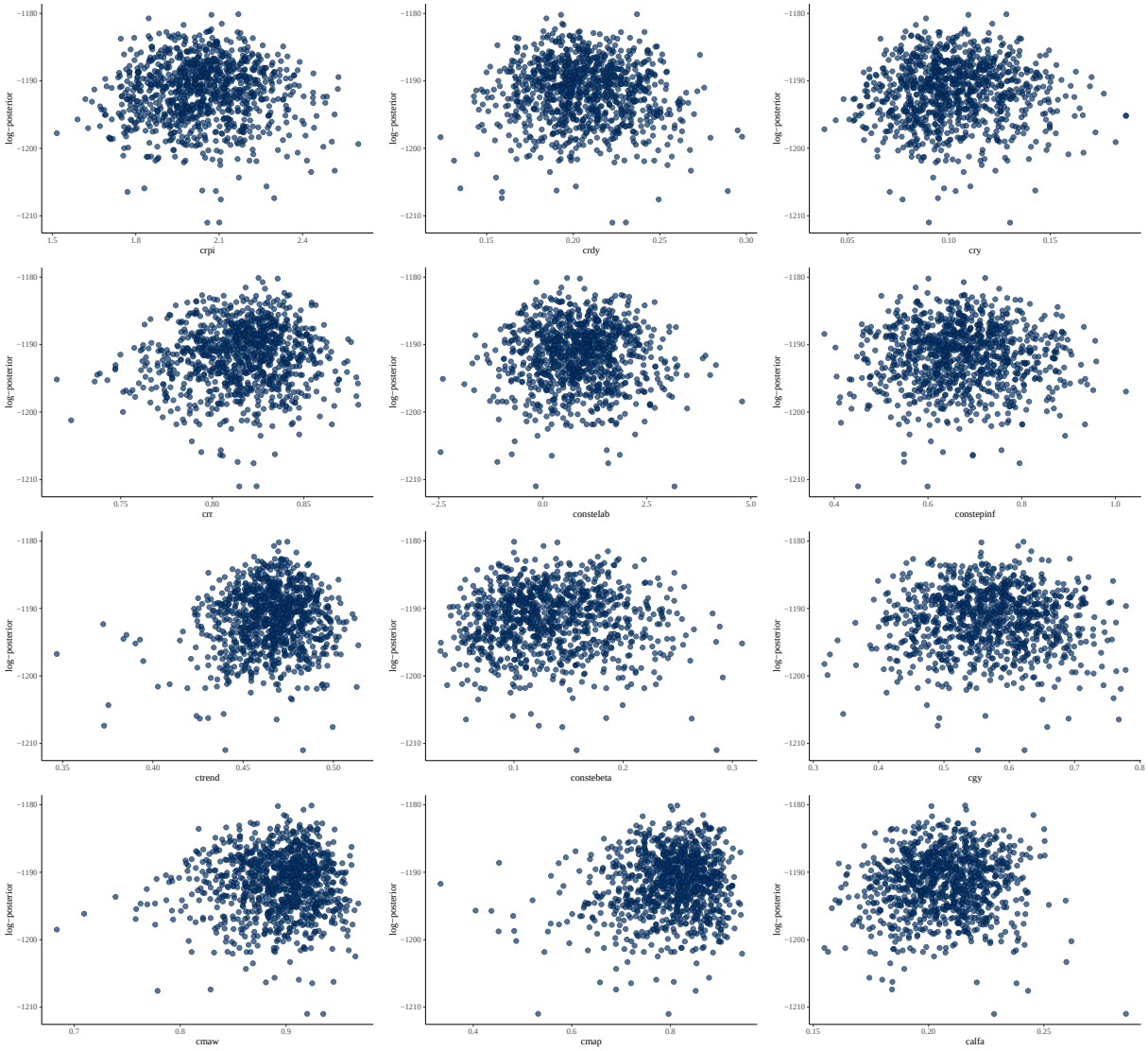
Acceptance Information

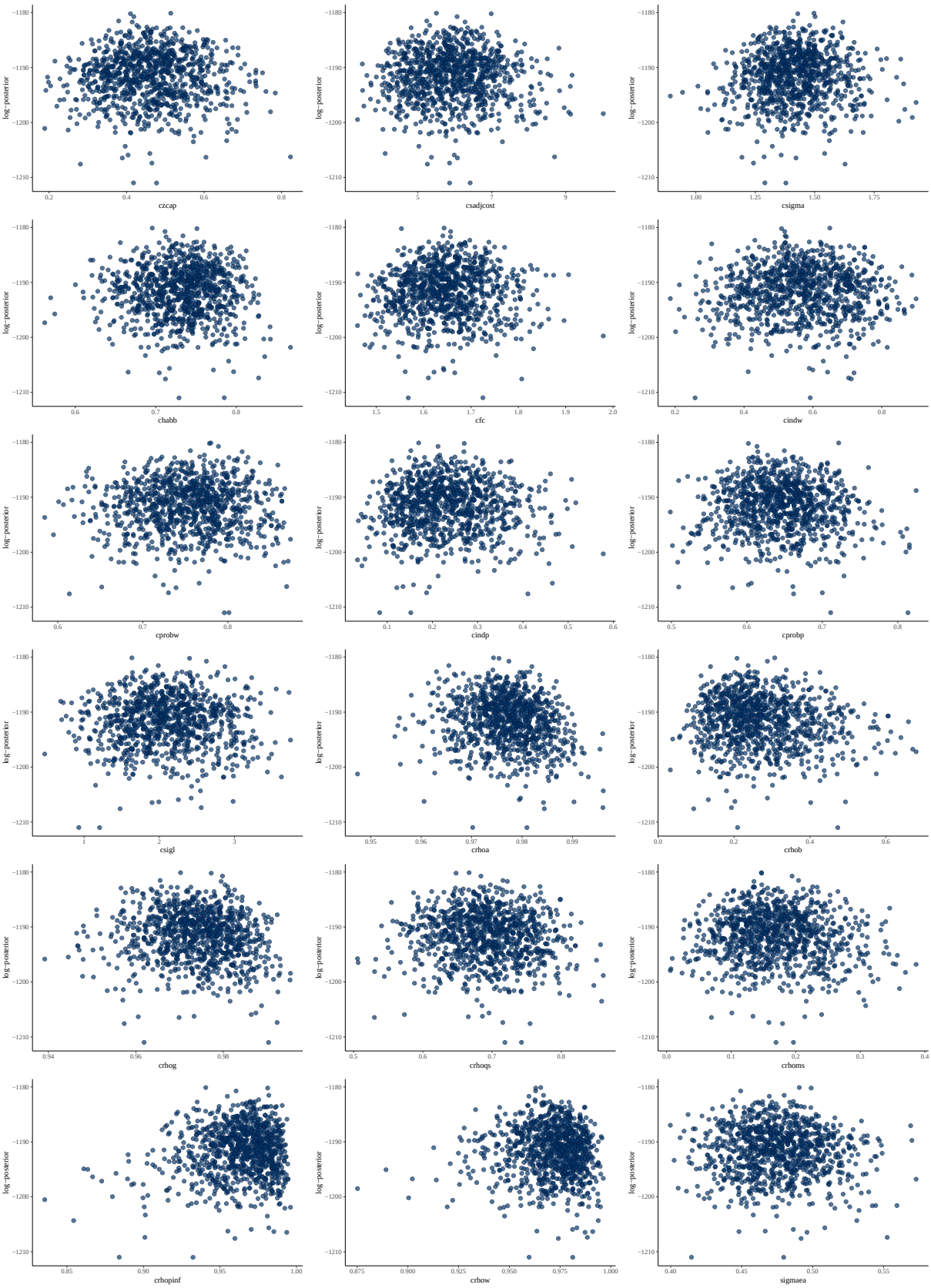
These are plots of the *acceptance statistic* (top left panel), the *log-posterior* (top right panel), and, the *acceptance statistic* (x-axis bottom panel) against the *log-posterior* (y-axis bottom panel) for all chains. The vertical lines indicate the mean (solid line) and median (dashed line). A 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 mitigated by reparameterization 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.

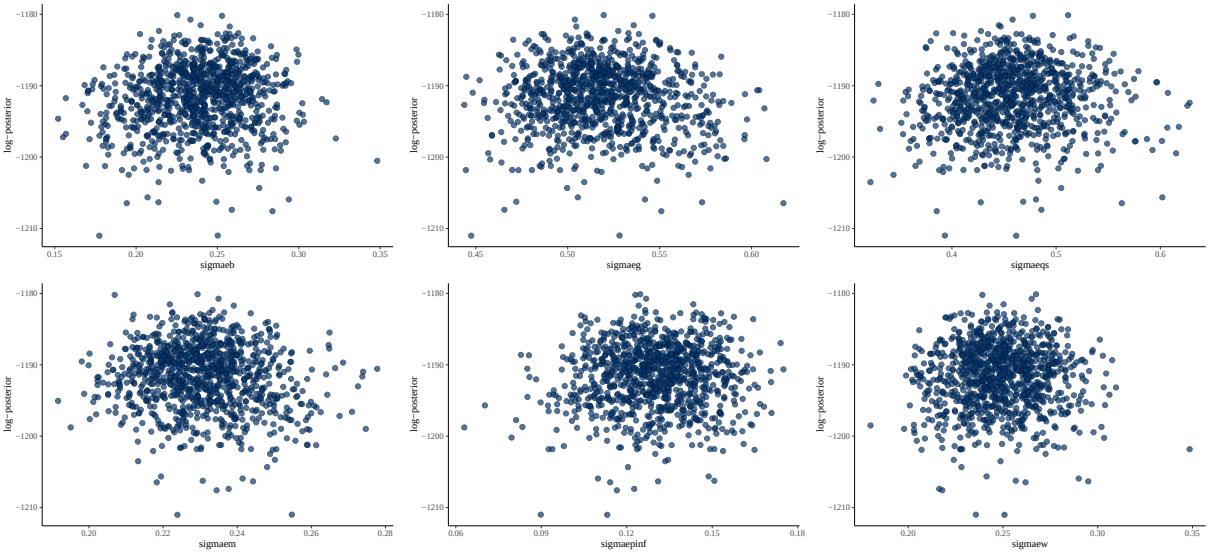


Scatter plots

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, 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.

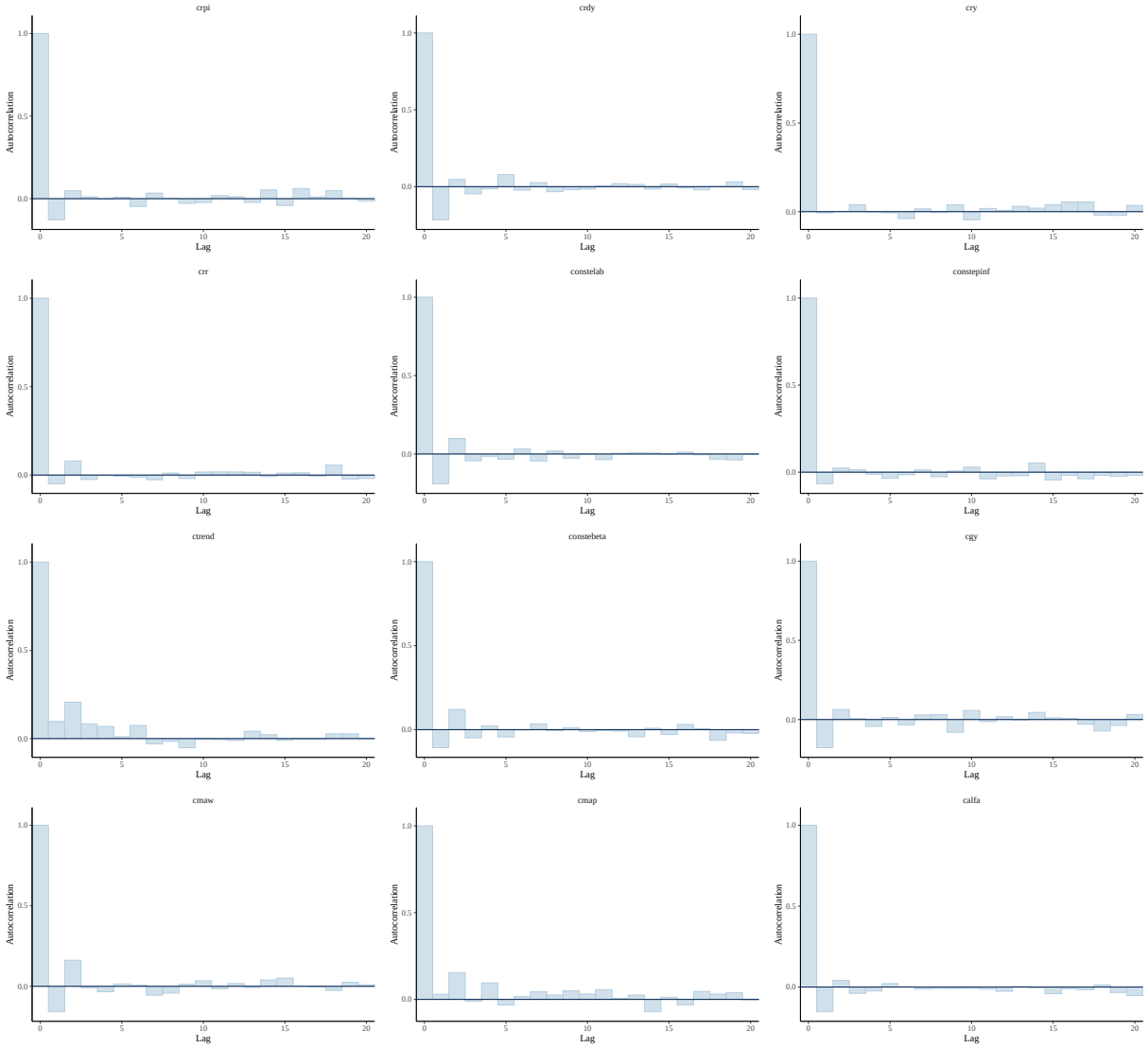


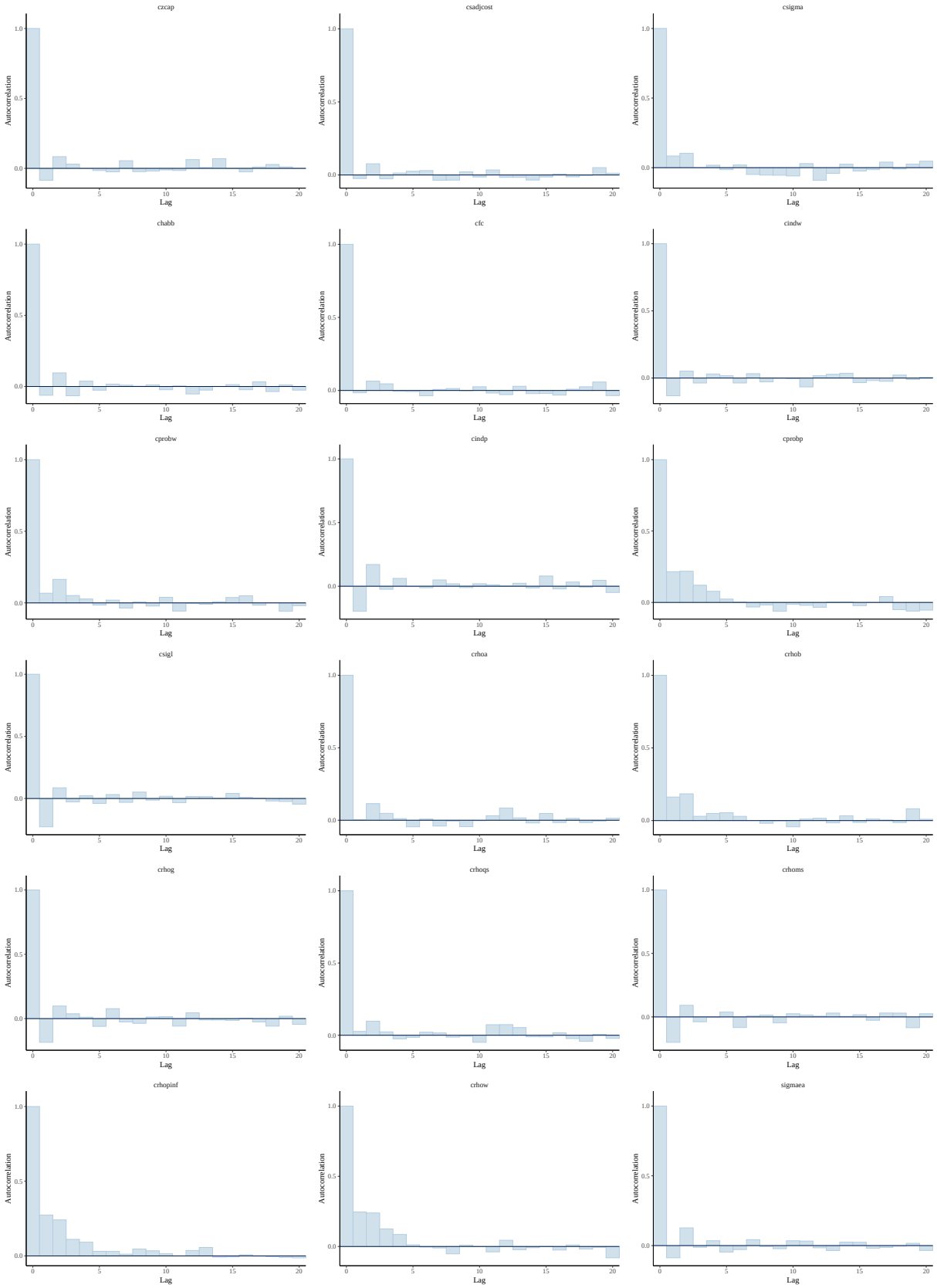

log-posterior
−1180
−1190
−1200
−1210
czcap
csadjcost
csigma
chabb
cfc
cindw
cprobw
cindp
cprobp
csigl
crhoa
crhob
crhog
crhoqs
crhoms
crhopinf
crhow
sigmaea

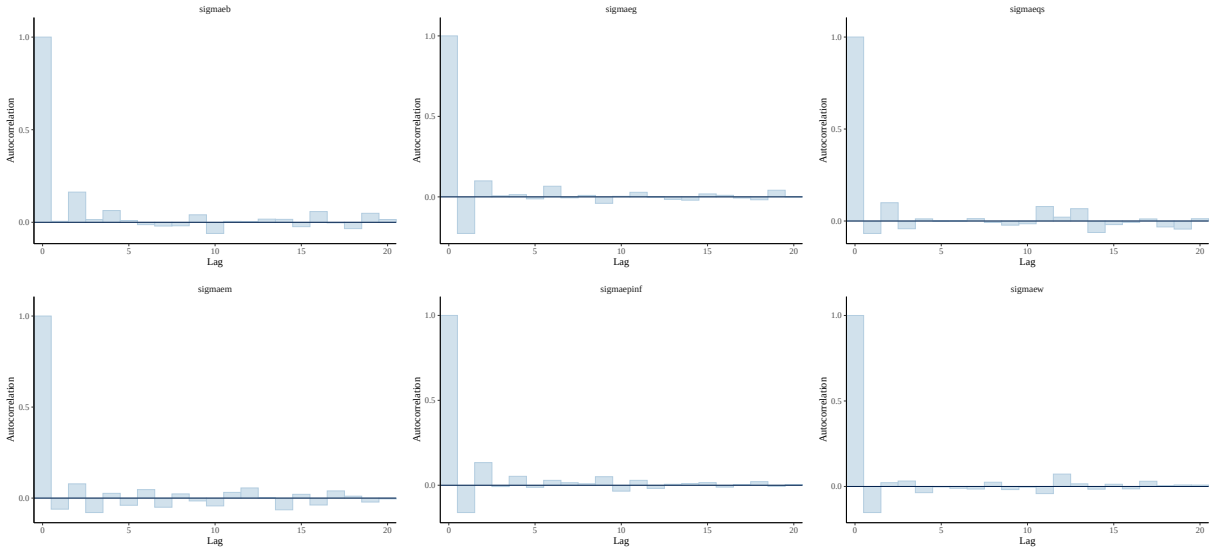

log−posterior
−1180
−1190
−1200
−1210
0.15
0.20
0.25
0.30
0.35
sigmaeb
0.45
0.50
0.55
0.60
sigmaeg
0.4
0.5
0.6
sigmaeqs
0.20
0.22
0.24
0.26
0.28
sigmaem
0.06
0.09
0.12
0.15
0.18
sigmaepinf
sigmaew

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. 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 reparameterization of the model.

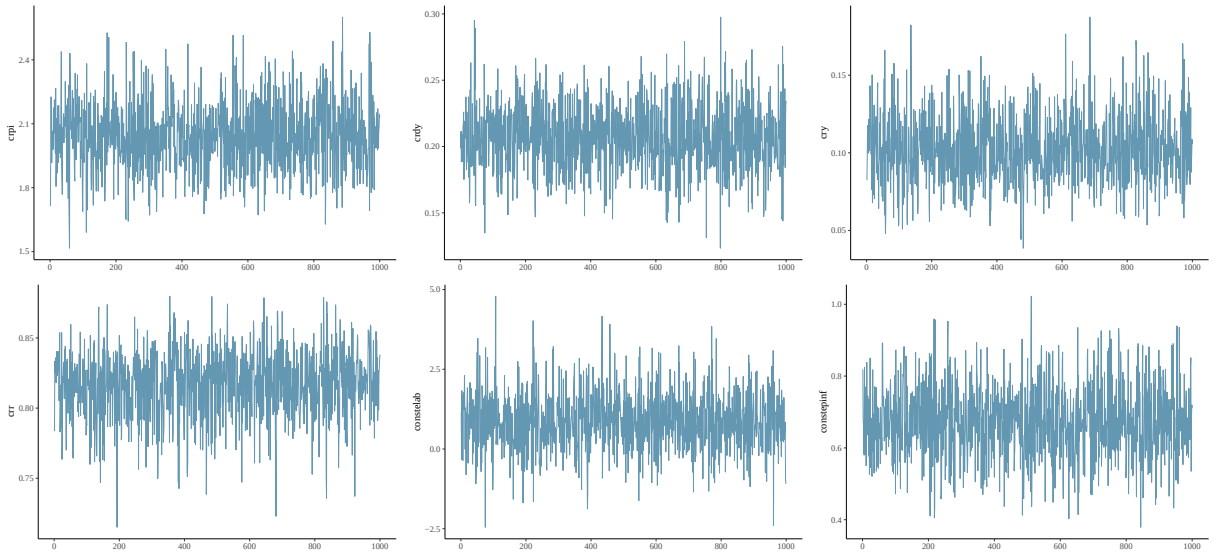


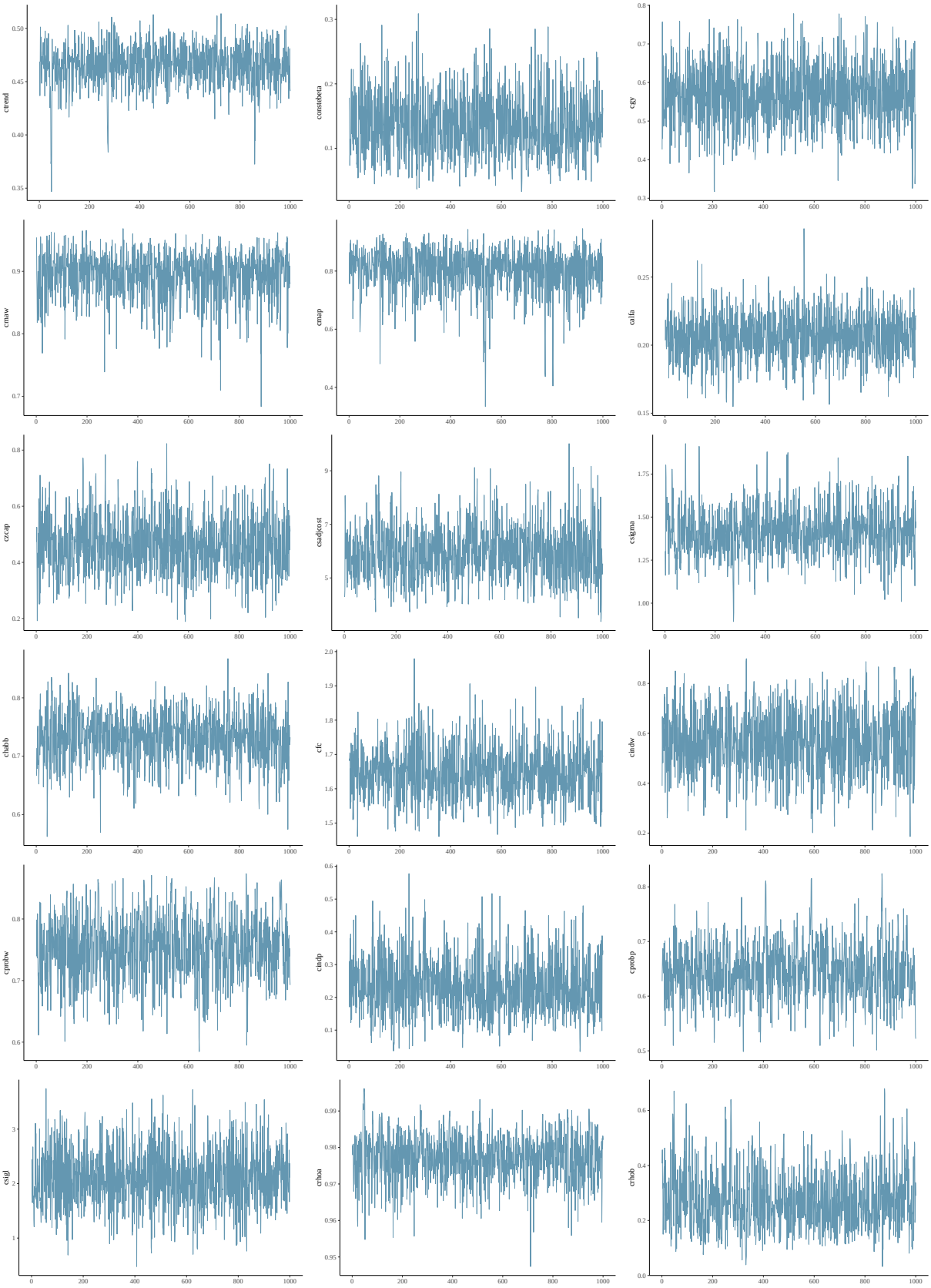

czcap
csadjcost
csigma
chabb
cfc
cindw
cprobw
cindp
cprobp
csigl
crhoa
crhob
crhog
crhoqs
crhoms
crhopinf
crhow
sigmaea
Autocorrelation
Lag

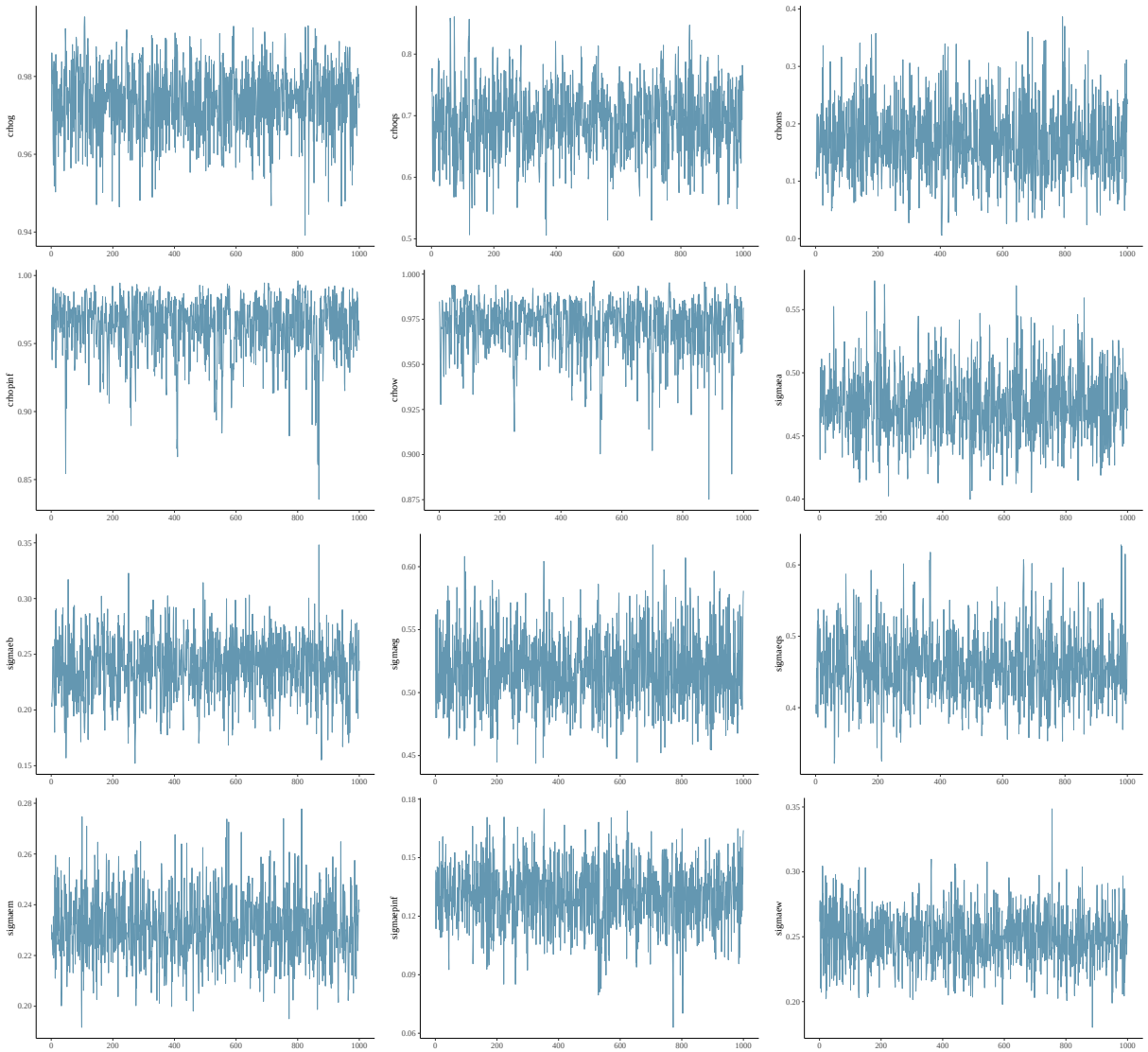

sigmaeb
sigmaeg
sigmaeqs
sigmaem
sigmaepinf
sigmaew
Autocorrelation
Lag
1.0
0.5
0.0
0
5
10
15
20

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. 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.




ctrend
consibeta
cgy
cmaw
cmap
calla
czcap
csaldjcost
csigma
chabb
cfc
cbdw
cpmobw
cbdp
cpmobp
csigl
choa
chob


crhog
crhogs
crhoms
crhopinf
crhow
sigmaeo
sigmaeb
sigmaeg
sigmaegs
sigmaem
sigmaepinf
sigmaew

Warnings

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

Numerical diagnostics

	n_eff	Rhat	mean	se_mean	sd		
log-posterior	365.59	1.00	-1199.89		0.23	4.31	
crpi	892.49	1.00	1.97		0.01	0.17	
crdy	942.12	1.00	0.22		0.00	0.03	
cry	1194.44	1.00	0.13		0.00	0.03	
crr	868.04	1.00	0.85		0.00	0.02	
constelab	846.38	1.00	0.61		0.03	0.91	
constepinf	922.57	1.00	0.67		0.00	0.10	
ctrend	710.82	1.00	0.46		0.00	0.02	
constebeta	962.80	1.00	0.14		0.00	0.05	
cgy	1213.01	1.00	0.57		0.00	0.08	
cmaw	642.98	1.00	0.91		0.00	0.03	
cmap	805.75	1.01	0.98		0.00	0.01	
calfa	1054.77	1.00	0.21		0.00	0.02	
czcap	898.15	1.00	0.40		0.00	0.10	
csadjcost	704.09	1.00	5.43		0.04	1.04	
csigma	562.87	1.00	1.41		0.01	0.14	
chabb	482.26	1.00	0.69		0.00	0.06	

cfc	889.32	1.00	1.63	0.00	0.08
cindw	1312.51	1.00	0.52	0.00	0.12
cprobw	470.28	1.00	0.80	0.00	0.04
cindp	1131.00	1.00	0.31	0.00	0.09
cprobp	788.72	1.01	0.80	0.00	0.03
csigl	1284.08	1.00	2.13	0.02	0.60
crhoa	728.30	1.00	0.97	0.00	0.01
crhob	385.27	1.00	0.41	0.01	0.15
crhog	609.54	1.00	0.97	0.00	0.01
crhoqs	651.13	1.00	0.71	0.00	0.06
crhoms	902.99	1.00	0.13	0.00	0.06
crhopinf	1005.06	1.00	0.93	0.00	0.02
crhow	359.52	1.00	0.97	0.00	0.02
sigmaea	1259.12	1.00	0.48	0.00	0.03
sigmaeb	456.17	1.00	0.21	0.00	0.04
sigmaeg	1121.21	1.00	0.52	0.00	0.03
sigmaeqs	806.33	1.00	0.45	0.00	0.05
sigmaem	1155.58	1.00	0.23	0.00	0.01
sigmaepinf	1240.97	1.00	0.21	0.00	0.02
sigmaew	930.52	1.00	0.23	0.00	0.02

Small Scale NK-Model: Hamiltonian Monte Carlo Estimation Effective Sample Sizes

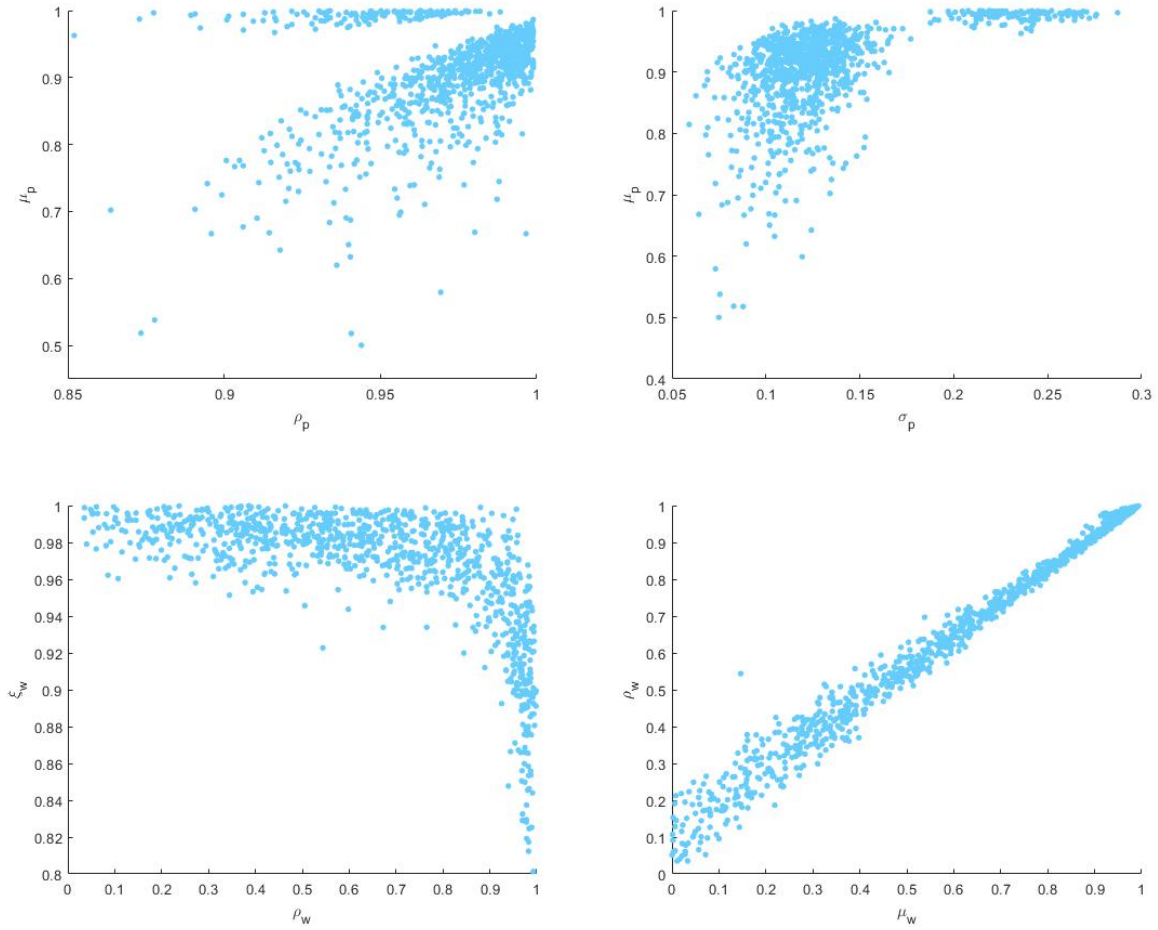
Table C.1: Posterior Estimates of an Overspecified Small Scale NK-DSGE Model with Generated Data

Parameters	$\rho_u = 0, \mu_u = 0$	$\mu_u = 0$	No restriction
τ	1158	590	989
κ	534	640	762
ψ_1	889	349	514
ψ_2	567	358	528
ρ_r	1025	509	906
ρ_g	833	800	157
ρ_z	913	578	78
$r^{(A)}$	252	322	513
$\pi^{(A)}$	272	306	338
$\gamma^{(Q)}$	333	229	542
$100\sigma_r$	902	677	987
$100\sigma_g$	1111	630	1064
$100\sigma_z$	782	737	575
ρ_{ug}	-	-	221
ρ_{uz}	-	-	93
μ_{ug}	-	1023	673
μ_{uz}	-	323	122

Notes: Effective sample sizes for 1,000 draws obtained by applying HMC to the small scale NK-DSGE model from Herbst and Schorfheide (2015) with 200 simulated data points where the error term is generated by an *iid* process. The first column shows results if the error terms for the exogenous processes g and z are modeled as white noise, the second column if as an MA(1) and the third column if ARMA$(1, 1)$.

D Sequential Hamiltonian Monte Carlo Estimation: Smets-Wouters Model

Figure D.1: Sequential Hamiltonian Monte Carlo - Joint Posterior Scatter Plots



Notes: The figure shows the joint posterior scatter plots of the following parameters: $[\rho_p,\mu_p]$ (upper left), $[\sigma_p,\mu_p]$ (upper right), $[\xi_w,\rho_w]$ (lower left) and $[\rho_w,\mu_w]$ (lower right). Sample size equals to 1,024, where two sample draws of the size $J = 512$, respectively, were merged, the first obtained by applying multinomial resampling, the second one by stratified resampling. The divergence rate at the last stage, $\beta_n = 1$, was approximately 0.2 percent and 0.4 percent, respectively, while the overall divergence rate throughout all $N = 37$ stages amounted to approximately 3.8 and 3.7 percent for both samples. Source: authors' simulations.

E Technical Appendix

Hamiltonian Monte Carlo: Methodology and Application to Linearized DSGE Models

E.1 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 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 $P(\theta, Y)$, represented by the corresponding density function $p(\theta, Y)$. Throughout this appendix distributions are represented by their density functions. The aim is to obtain the posterior density, that is, the density function of the model parameters given the data, denoted by $p(\theta|Y)$ which can be expressed by the means of Bayes' rule as follows:

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

$p(Y|\theta)$ is referred to as the likelihood function and $p(\theta)$ is the density function of the prior distribution. Typically, in a Bayesian estimation the *a priori* beliefs about the parameter vector θ 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 to 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 state variables. There exists a variety of DSGE solution methods, e.g. Blanchard and Kahn (1980), Binder and Pesaran (1997), King and Watson (1998), Uhlig (1999), Klein (2000), Kim (2000), Christiano (2002), Anderson (2010). A popular solution technique for a linearized DSGE model was proposed by Sims (2002) which starts with the following representation of the DSGE model:

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

where s_t is the set of state variables, ϵ_t is the structural shocks vector and η_t the vector of

the one step ahead rational expectation forecast errors, $x_t - \mathbb{E}_{t-1}x_t$. $\Gamma_0, \Gamma_1, \Psi, \Pi$ are real matrices of appropriate dimensions. The solution is based on the QZ-decomposition, also referred to as the Schur decomposition. If the above system has a unique stable solution then it can be represented in the following VAR-form:

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

Applying the solution method proposed by Sims (2002), or 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:

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

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

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

where $p(y_t|s_t, \theta)$ and $p(s_t|s_{t-1}, \theta)$ are the conditioned probability density functions of the observables and the states given the parameters and the present and lagged values of the states. 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 RWMH algorithm to sample from the posterior density, see Algorithm E.1.[44] In Algorithm E.1 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. The scaling parameter, c_0, should be also chosen in a way that the acceptance ratio equals to 23.4 percent, 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 20-40 percent.

There are several other modified versions of the MH algorithm. For example, the Block-MH algorithm breaks the parameter vector into blocks and updates at most only one

[44]This algorithm summarizes the main steps extensively described in Herbst and Schorfheide (2015).

Algorithm E.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 involves the solution of the DSGE model for θ, the calculation of $p(\theta)$, building the state space representation and the evaluation of $p(Y|\theta)$ by applying 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 $N(\tilde{\theta}, c_0^2\tilde{\Sigma})$), solve the DSGE model for $\theta^{(0)}$, calculate $p(\theta^{(0)})$, build the state space representation and evaluate $p(Y|\theta^{(0)})$ by applying the Kalman filter.

4. For $n = 1, ..., N$

   (a) Draw θ' from the proposal distribution $\mathcal{N}(\theta^{(n-1)}, c_0^2\tilde{\Sigma})$.

   (b) Solve the DSGE model for θ'.

   (c) Calculate $p(\theta')$, build the new state space representation and evaluate $p(Y|\theta')$ by applying the Kalman filter.

   (d) Accept θ', i.e. $(\theta^{(n)} = \theta')$, with probability $\min\{1, f(\theta^{(n-1)}, \theta'|Y)\}$ and reject $(\theta^{(n)} = \theta^{(n-1)})$ otherwise where

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

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

block of the parameters at once, applied e.g. by Cúrdia and Reis (2010) with fixed blocks. This scheme can be further extended by randomizing the break-up of the parameter vector into blocks in each step, as proposed by Chib and Ramamurthy (2010). A further possibility to improve the algorithm is to apply a more sophisticated proposal density. In particular, the Metropolis-Adjusted Langevin (MALA) algorithm, originally proposed by Besag (1994) and later assessed for its convergence properties by Roberts and Tweedie (1996), suggests to choose again a normal distribution, however the expected value should be adjusted by one step into the direction of the gradient of the negative log-posterior. Updating the current draw along the gradient, this algorithm accounts for the shape of the posterior density and therefore moves the chain, thus the new proposal for θ, into 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. As pointed out in Roberts and Tweedie (1996), using the Langevin-diffusion as an update should be a good choice as it is constructed in a way that under suitable regularity conditions in continuous time it converges to its stationary distribution. Therefore, even before the MH-step the candidate chain itself will approximate the target distribution to be sampled from. Herbst and Schorfheide (2015) uses instead of the normal distribution a t-distribution for the proposal. Furthermore, the algorithm benefits from scaling the step size along the gradient by the Hessian at the posterior mode, also pointed out in Herbst and Schorfheide (2015), based on Roberts and Stramer (2002). The MH-Newton algorithm differs from the latter MALA-algorithm in the way that instead of the Hessian at the posterior mode the Hessian at $\theta^{(n-1)}$ is taken and the step size is randomly chosen. For further details we refer to Qi and Minka (2002).[45]

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 the literature, e.g. Fernandez-Villaverde and Rubio-Ramirez (2007). At the same time, particle filters are also applied if the posterior likelihood is ill-shaped, e.g. the Sequential Monte Carlo (SMC) algorithm in Herbst and Schorfheide (2014), which may even occur when standard models are estimated using first order linear approximations. Our extension of the HMC with the SMC algorithm falls into the latter category of application.

E.2 The Hamiltonian Monte Carlo Method

This section of the paper provides an introduction to the HMC framework. It is aimed to offer intuition to the reader and to reveal a few main theoretical aspects of the methodology.[46] Similarly to advanced MCMC methods from above, the HMC algorithm builds on the information provided by the gradient of the log-posterior density function. In particular, it uses the information in the geometry of the target distribution, that is, its

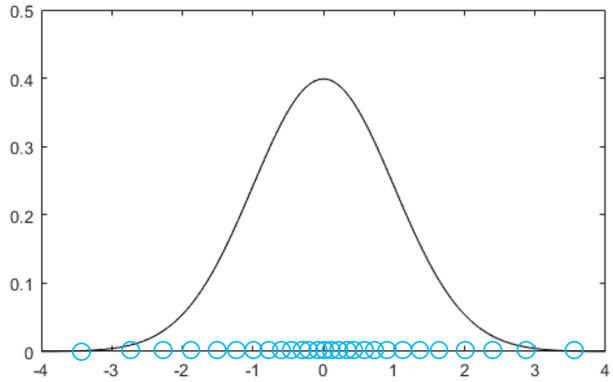
[45] For further discussion and a comparison of the different estimation methods we refer to the work of Herbst and Schorfheide (2015).

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

shape and the equations characterizing it. Its main advantage is that by means of the Hamiltonian equations, a concept 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.

Before providing a formal descriptive introduction to HMC, let us first illustrate its main concept by applying it to a simple example.[47] In physics researchers model the evolution of a mechanical system over time given a particle's position and momentum usually by functions measuring its potential and kinetic energy. Classical examples of a mechanical system are a bouncing ball, a pendulum or an oscillating spring. Let us assume that one intends to sample from a one dimensional standard normal distribution with density function $f(q) = 1/(2\pi)^{1/2}\exp\{q^2/2\}$. Intuitively, the aim is to generate more samples for q in those regions of the domain of q, that is the real line, where the density function $f(q)$ peaks than in regions of its tails.[48] In particular, one would like to generate samples from each location of the domain of q in proportion to the value of the density function, $f(q)$, at that location. This case is illustrated in Figure E.1, where the target density is a normal distribution in black, while the blue circles on the real line represent the desired samples.

Figure E.1: Sample Draws from a Normal Distribution



Notes: The blue circles show desired sample draws from a standard normal target density, displayed in black color.

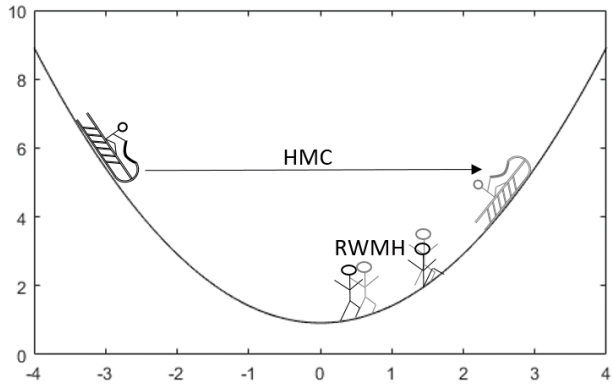
To demonstrate the idea of HMC, let us first consider instead of $f(q)$ the negative logarithm of the density function $g(q) := -\log f(q)$. That is, we flip over $f(q)$ and obtain a valley-shaped function $g(q)$, being the new target density. Next, we imagine a person

[47] Our explanation draws on the example in the excellent book of Lambert (2018).

[48] This intuition, that the typical set is close to the mode does not apply in higher dimensional problems, see Betancourt (2018) for additional details.

using a sleigh and trying to explore the valley covered in snow, i.e. the new target density defined by $g(q)$. We assume that our explorer starts out at some point in the valley, sitting on a sleigh, i.e. starting out somewhere on the surface generated by the function $g(q)$ and is initially pushed randomly with some impulse, either uphill or downhill, see Figure E.2. In contrast, think of the RWMH algorithm as a person exploring the valley on foot and proceeding in equally distant steps in random directions. Assuming that our explorer on the sleigh started its journey downhill, after passing the trough of the valley she will continue sliding uphill. After getting gradually slower she will stop at some point and will start sliding again downhill into the opposite direction.

Figure E.2: Hamoltionan Monte Carlo vs. Random Walk Metropolis Hastings



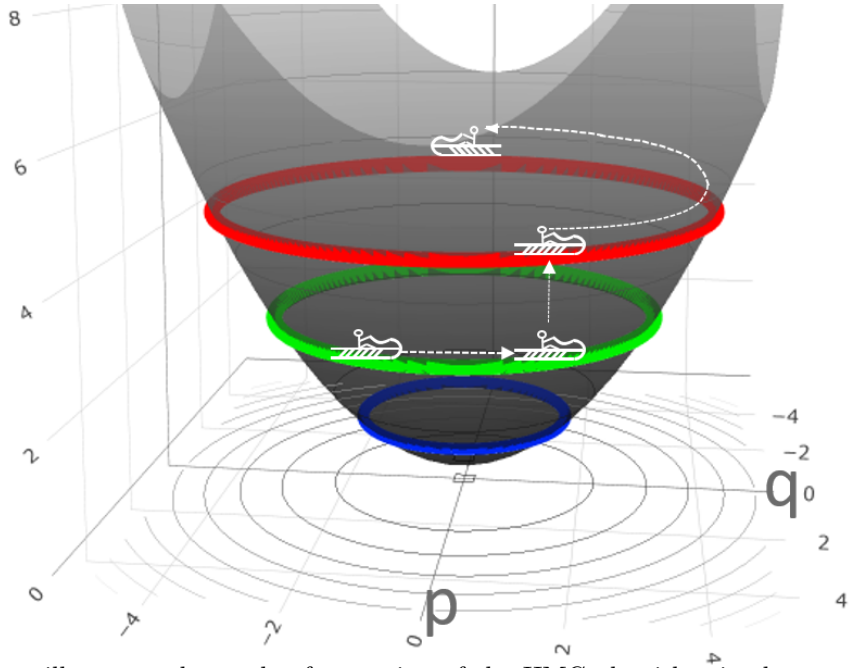
Notes: The figure shows an illustrative comparison between the mode of operation of the RWMH and the HMC algorithm if sampling from a one dimensional normal density.

Assuming also that the snow-covered surface of the valley is frictionless, this motion will continue forever. The motion of the sleigh on a frictionless surface can be perfectly described by the Hamiltonian equation in analogue to the mechanical systems mentioned before. The total energy of the sleigh can be described by its potential energy function $U(q) := g(q)$ and its kinetic energy $K(p)$, depending on its momentum p. By moving up or down the valley the explorer on the sleigh exchanges potential energy $U(q)$ for kinetic energy $K(p)$. As the sleigh slides down (up) the hill, its potential energy will decrease (increase) and its kinetic energy increase (decrease). After some time $t = T$ we record the position q of our sleighing explorer. Given the shape of the terrain the person with the sleigh will be more often in lower regions of the valley than in higher regions. In contrast, a RWMH explorer would find it easy to walk downhill to the trough of the valley, but would struggle to make steps upwards. In other words, under RWMH sampling a proposal is always accepted if the probability of the new proposal q' is higher

compared to the probability of the current draw q, steps downhill, and only accepted randomly depending on the proportion of the probabilities of q' and q, steps uphill. Using RWMH sampling, the explorer would walk down to the valley, and explore the region in the bottom, i.e. the target density around the mode, but always struggle to make steps uphill and proceed only very slowly.

A fair question to ask is, what is the reason for the explorer on the sleigh sliding and arriving always safely at the other side of the valley without struggling. The core of the concept is that by tracking the momentum p and the kinetic energy $K(p)$ the parameter space is extended by the same dimension to measure total energy. This extended space is called the phase space and is fundamental for Hamiltonian dynamics. Defining the kinetic energy by $K(p) := |p|^2/(2m)$, where m corresponds to the mass of the sleigh with the person, we can describe the system entirely with the Hamiltonian equation $H(q, p)$, also referred to as the total energy function.

Figure E.3: Visualization of the Hamoltionan Monte Carlo Algorithm



Notes: The figure illustrates the mode of operation of the HMC algorithm in the extended parameter space if used to sample from a normal density. When the sleigh moves horizontally on the same height the Hamiltonian equations are used to calculate the path and keep the sleigh on the same energy level. The vertical move of the sleigh demonstrates a new impulse draw after the sleigh was stopped and the sample draw was recorded. The new impulse moves the sleigh to a different energy level on which it again continues exploring the extended parameter space horizontally along the new path defined by the Hamiltonian equations. Source: author's illustration.

Visually, extending the parameter space and tracking both position and velocity at the same time allows the sleigh moving from one side to the other side of the valley by sliding

around on the same (energy) level, hence moving only horizontally in this simple case, see Figure E.3. In this extended space, stopping and then pushing the sleigh with a different impulse is equivalent to 'moving' it onto a different height, hence energy level, sliding again along the Hamiltonian, i.e. the contour lines of the extended space. When stopping the sleigh, the explorer 'brakes', throws away kinetic energy $K(p)$, e.g. by wasting it while braking, and records only its position q.

Considering a higher dimensional parameter space with a more irregular density function to sample from, extending the parameter space and applying the Hamiltonian serves as a sort of secure trail of movement for our explorer. It allows for being able to describe an exact path to move along on a (possibly) complicated extended surface so that one remains always at the same energy level. Hence, the new proposal for the parameter will always be accepted.

Turning now to a more formal description, 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 to MCMC applications outside physics, by regarding the position of the sleigh, 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:

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

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

where $2d$ equals the full dimension of the system. The equations of motion can be pre-

sented in a more compact way by defining $z := (q, p)$ so that

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

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 $T_s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ with $(q, p)(t) \to (q, p)(t + s)$. Thus, the Hamiltonian equations describe the law of motion of the system from t to $t + s$.

The Hamiltonian measures total energy in the system, that is potential energy and kinetic energy, consequently for the HMC algorithm it takes the additive form

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

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

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

where M is referred to as the 'mass matrix' which is typically diagonal, and is often a scalar multiple of the identity matrix as it often stands for the mass of some bodies or particles.[49]

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 venturing too deeply into details of volume measures of a phase space it suffices to state that this property is necessary to avoid accounting for a change in the volume when computing the acceptance probability.

[49] In the estimation procedure this assumption can be released and the off-diagonal elements can be also non-zero.

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:

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

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 holds that $J^{-1} = J^T = -J$. The symplecticness condition 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. Consequently, 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 sleigh one can imagine this as stopping the explorer at $q(t+s)$ and pushing the sleigh 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 explorer departed from. The reversibility property is crucial when proving the detailed balanced condition in the probabilistic framework which ensures together with ergodicity that 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:

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

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 (q, p) which results in the following cannonical distribution:

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

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

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

Setting for $U(q)$ the negative logarithm of the target density $-\log(p(Y|\theta)p(\theta))$ and for $K(p)$ the kinetic energy function allows to define an algorithm which samples from the distribution of interest. The iteration is carried out in three steps as described by Algorithm E.2. As the total energy in the system remains constant, in theory the proposal

Algorithm E.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 and the momentum vector (q', p') by applying the Hamiltonian equations deterministically starting from $q = \theta^{(n)}$ and p.
3. Metropolis-Hastings step: accept the new proposal and set $\theta^{(n+1)} = q'$ with probability $\min[1, \exp(-(U(q') - U(q) + K(p') - K(p)))]$

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 in order 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:

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

where P_K is the HMC kernel. The key property that allows to proof that the detailed balance condition holds is the 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 latter 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 and would go beyond the scope of this paper. 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) points out that in theory it is possible that ergodicity fails once a fixed number of integration steps is used for the numerical approximation of the solution and illustrates this based on a short example. Mackenze (1989) proposes that by randomizing the length of the Hamiltonian trajectory this issue can be eliminated while recently more general conditions for ergodicity, and hence for convergence of the HMC algorithm could be proved, see e.g. Livingstone et al. (2018) and Durmus et al. (2019).

E.3 Implementation in *Stan*

Stan 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 often applied to perform computationally highly intensive calculations due to its performance, 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 a powerful and complex concept available to researchers, working out-of-the-box.

E.3.1 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 and the total energy level, which is the sum of potential energy and kinetic energy. One of the key challenges lies in the accurate solution of the Hamiltonian equations. As discussed before, the Hamiltonian system is symplectic. Thus, a dedicated class of symplectic 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, hence a long period of time. *Stan* uses a simple implementation referred to as 'leapfrogging' to solve for the discrete-time approximation of the Hamiltonian equations which is summarized by Algorithm E.3. 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 the evaluation of 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.

Algorithm E.3 Leapfrogging

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

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 is particularly useful when implementing the estimation of DSGE models. Therefore, there is no need for the user to specify any derivatives manually, 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 depends on the solution of the DSGE model is computationally involved for a mid-scale DSGE model, the latter is performed by *Stan* automatically and efficiently due to the availability of symbolic differentiation.

The performance of the algorithm is sensitive to the selection of two parameters: the step size, 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 the distance $||\theta' - \theta||$ 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 higher 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 warm-up 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 percent, significantly higher than 23.4 percent in the RWMH algorithm. In case the divergence rate remains still high, the automated calibration mechanism in *Stan* can be still instructed to target higher acceptance ratios.

It is 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 too close to 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 automatically optimizes 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 directions, doubling the number of steps each time. 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 continues until some proposal would move backwards to its original point of departure, θ, making a U-turn and moving again towards the point of departure. *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 the reversibility condition of the generated Markov chain.

The mass matrix M, being typically a diagonal matrix is tuned automatically during the warm-up. Here, the user is allowed to tune M manually, however the automated tuning process of *Stan* operates sufficiently well. Furthermore, if desired, a dense matrix with non-zero off-diagonal elements can be also applied for the estimation, which could

result in improved efficiency.

A further useful feature which is implemented in *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 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, *Stan* instead reparameterizes 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 of executing calculations with complex numbers, hence a QZ decomposition cannot be implemented. Furthermore, it might be 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 DSGE solution method to remedy these shortcomings is the Binder and Pesaran (1997) solution algorithm.

E.3.2 Binder-Pesaran Algorithm

The main idea of the Binder and Pesaran (1997) algorithm is to rewrite s_t in a way 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 stable 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 than the Sims (2002) canonical formula:

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

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

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

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 might be slightly restrictive at first sight. However, the matrix can become only non-invertible when a linear combination of future expectations in $t+1$ depends only on linear combinations of past values of endogenous variables and shocks which does not seem to be an issue in practice. Anderson (2008) also compared and benchmarked a handful of DSGE solution algorithms on several models and did not report any issues related to the non-invertibility of M_{00}.[50] 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

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

Substituting out s_t again, collecting and rearranging terms yields

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

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

$$\mathrm{C}_{k+1} = (\mathrm{I} - \mathrm{BC}_k)^{-1}\mathrm{A} \tag{E.20}$$

Anderson (2008) reported though that solving this quadratic equation system is very costly and in some cases the iteration failed to converge to the correct solution. A potential reason could be for example that inverses of badly conditioned matrices are inaccurate. As a solution one can reparameterize the model, precondition the equation, or use a different

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

iterative method.[51] We choose the latter and present our approach in detail in the next subsection. Given a solution for C the system of equations can be written as follows:

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

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

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

To arrive to the unique stable solution of the original model it suffices to plug back the definition of S_t:

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

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

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

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

Algorithm E.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 $\mathrm{C}_{k+1} = (\mathrm{I} - \mathrm{BC}_k)^{-1}\mathrm{A}$ with an educated guess or setting $\mathrm{C}_0 = \mathrm{A}$ 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 in a

[51] The quadratic matrix equation could be also solved by other techniques from linear algebra, however this would again involve the calculation of generalized eigenvalues.

way that the solution method can be implemented in the *Stan* software package.

E.3.3 Further Computational Issues

To find a solution to a DSGE model Binder and Pesaran (1997) propose to iterate the solution to the C matrix using the following rule: $C_{k+1} = (I - BC_k)^{-1}A$. Although *Stan* is able to cope with the latter iteration types, the calculation 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 $C_{k+1} = BC_k^2 + A$ until it converges.[52]

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 a unique stable solution exists were 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 on the Sims (2002) algorithm. Although *Stan* is not capable of dealing with complex numbers, external functions can be included into the algorithm and also partial derivatives of external functions could be manually specified. Yet, this is not necessary as the Sims (2002) algorithm is only used 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 we rely on the Intel Math Kernel Library (Intel MKL), a collection of BLAS and LAPACK algorithms, also used by Matlab, which we link into our *Stan* C++ code. Alternatively, one could also possibly assess the uniqueness of the obtained solution by the methodology provided in Lan and Meyer-Gohde (2014).

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

[52] Although there is no guarantee for convergence we were not confronted with the latter issue when executed the algorithm. Furthermore, one can always check the correctness of the solution by plugging it back into the quadratic equation to be solved.

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 achieve 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 accelerate 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 the 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.