

Winschel, Viktor; Krätzig, Markus

**Working Paper**

## Solving, estimating and selecting nonlinear dynamic models without the curse of dimensionality

SFB 649 Discussion Paper, No. 2008,018

**Provided in Cooperation with:**

Collaborative Research Center 649: Economic Risk, Humboldt University Berlin

*Suggested Citation:* Winschel, Viktor; Krätzig, Markus (2008) : Solving, estimating and selecting nonlinear dynamic models without the curse of dimensionality, SFB 649 Discussion Paper, No. 2008,018, Humboldt University of Berlin, Collaborative Research Center 649 - Economic Risk, Berlin

This Version is available at:

<https://hdl.handle.net/10419/25260>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*

SFB 649 Discussion Paper 2008-018

# Solving, Estimating and Selecting Nonlinear Dynamic Models without the Curse of Dimensionality

Viktor Winschel\*  
Markus Krätzig\*\*



\* Universität Mannheim, Germany

\*\* Humboldt-Universität zu Berlin, Germany

This research was supported by the Deutsche  
Forschungsgemeinschaft through the SFB 649 "Economic Risk".

<http://sfb649.wiwi.hu-berlin.de>  
ISSN 1860-5664

SFB 649, Humboldt-Universität zu Berlin  
Spandauer Straße 1, D-10178 Berlin



SFB 649 ECONOMIC RISK BERLIN

# Solving, Estimating and Selecting Nonlinear Dynamic Models without the Curse of Dimensionality<sup>1</sup>

Viktor Winschel<sup>2</sup>, Markus Krätzig<sup>3</sup>

February 7, 2008

**Abstract:** We present a comprehensive framework for Bayesian estimation of structural nonlinear dynamic economic models on sparse grids. The Smolyak operator underlying the sparse grids approach frees global approximation from the curse of dimensionality and we apply it to a Chebyshev approximation of the model solution. The operator also eliminates the curse from Gaussian quadrature and we use it for the integrals arising from rational expectations and in three new nonlinear state space filters. The filters substantially decrease the computational burden compared to the sequential importance resampling particle filter. The posterior of the structural parameters is estimated by a new Metropolis-Hastings algorithm with mixing parallel sequences. The parallel extension improves the global maximization property of the algorithm, simplifies the choice of the innovation variances, allows for unbiased convergence diagnostics and for a simple implementation of the estimation on parallel computers. Finally, we provide all algorithms in the open source software JBendge<sup>4</sup> for the solution and estimation of a general class of models.

**Keywords:** Dynamic Stochastic General Equilibrium (DSGE) Models, Bayesian Time Series Econometrics, Curse of Dimensionality

**JEL classification:** C11, C13, C15, C32, C52, C63, C68, C87

---

<sup>1</sup>We thank Wouter Denhaan, Paul Fackler, Jesús Fernández-Villaverde, James Heckman, Florian Heiss, Kenneth Judd, Michel Juillard, Felix Kübler, Alexander Ludwig, Thomas Mertens, Juan Rubio-Ramírez and the participant of the Institute on Computational Economics at the University of Chicago & Argonne National Laboratory, 2005, for discussion and inspiration. We also thank the anonymous referees for their valuable comments. This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 Economic Risk.

<sup>2</sup>winschel@rumms.uni-mannheim.de, University of Mannheim

<sup>3</sup>mk@mk-home.de, Universität zu Berlin

<sup>4</sup>JBendge is an acronym for "Java based Bayesian Estimation of Nonlinear Dynamic General Equilibrium models". The homepage of this project is at <http://jbendge.sourceforge.net/>. It is licensed under the GPL.

# 1 Introduction

Many modern macroeconomic models with microeconomic foundations and rational expectations are nonlinear. The simplest solution method is to linearize the model and to solve the associated quadratic matrix equation for the optimal policy function, as in Klein (2000). The likelihood implied by the solution can then be evaluated by the Kalman (1960) filter. The drawback of linearization is that it produces biased estimates and implied moments of the observables which may misinform the model evaluation and its further development. Moreover, Fernández-Villaverde and Rubio-Ramírez (2006) find evidence for important nonlinearities in macroeconomic data. Beside a poor empirical performance, a theoretical problem of linear models is that their solution is certainty equivalent, i.e. it does not depend on shock variances and therefore a conclusive welfare analysis of risk is not possible. Kim and Kim (2003), for example, discuss a spurious welfare implication of a linearized model.

One nonlinear estimation approach is the method of moments. It suffers from a small sample bias and does not efficiently use the available information, neither for parameter estimates nor for a model selection criterion. According to the likelihood principle, for example in Berger and Wolpert (1988), all sample information is contained in the likelihood function. Unfortunately a nonlinear likelihood approach is a complex numerical operation with at least four problems. We propose innovations to all of them.

The first problem is to solve the model. Here we use the Smolyak operator for a Chebyshev approximation of the model solution as well as for a Gaussian quadrature of the rational expectations. The second problem is to evaluate the likelihood, where we introduce three new nonlinear state space filters. The third problem is to generate parameter estimates. We do this by a new parallel Metropolis-Hastings algorithm. The fourth problem is to program all these complex interacting algorithms. As a solution we provide the open source software JBendge for a general class of models.

There are many different methods to approximate the model solution. The currently predominant approximation strategy is a local perturbation approach based on the implicit function theorem discussed by Judd and Guu (1997), Gaspar and L. Judd (2005) or Judd and Jin (2002). Current software packages implementing these methods are Juillard (1996), Anderson, Levin, and Swanson (2005) or Schmitt-Grohé and Uribe (2004). Global approximations can be done by Chebyshev polynomials or finite elements. Projection methods are discussed in Judd (1992). A recent comparison of all three approaches for the growth model can be found in Aruoba, Fernández-Villaverde, and Rubio-Ramírez (2006).

Solutions for larger models with more than a few states are currently approximated exclusively by perturbation. The problems of a perturbation approximation is that its validity may be restricted to a ball around the steady state, which size is difficult or impossible to derive and that the approximation quality may be unacceptable if one moves away from the steady state. This is especially obstructive if the economic argumentation is that nonlinearities arise due to the system's fluctuations far from the steady state.

The problem with global approximation is that it is by now widely believed to suffer from the curse of dimensionality. The curse arises if the tensor operator is used to extend the univariate approximation or integration operator to many dimensions. The curse indicates exponentially growing computational costs for a growing dimensionality of the approximated function or integrand. It effectively restricts tensor based global approximation methods to be useful only for a few (in our case - six) dimensions.

We replace the tensor operator by the Smolyak (1963) operator. This operator decreases the approximation burden for many functional forms from exponentially growing costs to a polynomial order. It also implies that integrals beyond six dimensions do not need to be approximated exclusively by Monte Carlo methods. The Smolyak operator can be used for a Chebyshev approximation, as we do, but also for finite elements, wavelets and other forms of the approximating function. Bungartz and Griebel (2004) provide an extensive summary of this active field in numerics. Krüger and Kübler (2004) were to our best knowledge the first who applied the operator in an economic setting for the approximation of the solution of a high dimensional OLG model. We use the operator for a Chebyshev approximation of the model solution as well as for the integrals implied by rational expectations.

Numerical integration problems also appear in the likelihood evaluation of nonlinear state space models. The current approach to these models is dominated by the sequential importance resampling particle filter. The simple particle filter and extensions are described in Doucet, de Freitas, and Gordon (2001). We introduce three new nonlinear filters based on deterministic integration in order to overcome the high computational burden of the particle filter.

The first filter we propose is the simple but very fast deterministic Smolyak Kalman filter which is an improvement of the deterministic unscented Kalman filter by Julier and Uhlmann (1997). The unscented method is in fact a deterministic multivariate integration method which restricts the computational costs to rise linear in the dimensions of the integrand. The reverse of the medal is that the quality of the unscented integration decreases with the dimensionality and nonlinearity of the integrand. Our approach is therefore better suited for nonlinear and high dimensional applications in economet-

rics. Moreover the Smolyak based multivariate Gaussian quadrature can be more easily adapted to a setting with non normal shocks since Gaussian quadrature is readily available for many different densities. The Smolyak Kalman filter, as the unscented filter, is based on the assumption that some densities processed within the filtering recursions are Gaussian.

For a better approximation of non-Gaussian densities, we propose, as the second new filter, an extension of the sequential importance resampling particle filter, where the importance density is generated by the deterministic Smolyak Kalman filter. We call this combination the Smolyak particle filter since it bears similarities to the combined unscented particle filter by van der Merwe, Doucet, de Freitas, and Wan (2000).

Finally, we introduce the third filter which approximates the densities within the filtering recursions by sums of Gaussian densities. The purpose of these filters, is to demonstrate robustness of the estimation results. They also demonstrate that there are various possibilities to apply the Smolyak operator in the nonlinear filter design and that a Monte Carlo approach, as in the sequential importance resampling filter, is not the only way to implement nonlinear filters.

For the estimation problem we extend the Metropolis-Hastings algorithm to a parallel version and implement a feature from the genetic algorithm of Storn and Price (1997), called differential evolution. This feature improves the global maximization properties of the Metropolis-Hastings algorithm and simplifies the choice of the innovation variances since it effectively estimates the optimal variances on the run. It also allows an unbiased convergence diagnostics since the parallel sequences start from different parameter vectors. And finally due to its parallel nature it provides a natural way to implement the estimation on parallel computers.

The curse of dimensionality for numerical approximation and integration arises in many estimation, inference and forecasting problems of Bayesian, classical or nonparametric econometrics. Beside the application which we propose in this paper we expect the Smolyak operator and more elaborate sparse grids approaches to be useful in many other econometric algorithms. One example is presented in a Monte Carlo experiment by Heiss and Winschel (2008) where the Smolyak based mixed logit estimator systematically outperforms simulation based estimators in a maximum likelihood setting.

Only few attempts to estimate nonlinear dynamic structural models with likelihood methods are presented in the literature. Fernández-Villaverde and Rubio-Ramírez present in a number of papers (2004, 2005, 2006) the marginal likelihood as a model selection criterion for non-nested and quasi-true models, discuss a nonlinear structural parameter estimation and compare linear to nonlinear estimates. They approximate the model solution by finite

elements and perturbation, evaluate the likelihood by the particle filter and estimate the posterior by the serial Metropolis-Hastings algorithm. A recent nonlinear estimation is presented by Amisano and Tristani (2007) based on a perturbation solution, a filter called the conditional particle filter by Ionides, Bretó, and King (2006) and the serial Metropolis-Hastings algorithm. The filter is a combination of the deterministic filter similar to the extended Kalman filter and the particle filter and comparable to the unscented particle filter and our Smolyak particle filter.

The rest of the paper is divided into five sections: economics, econometrics, results, conclusion and the appendix. In the economic section 2 we present a generic model class, the optimality conditions for the example model, the nonlinear Chebyshev approximation, the Smolyak operator and the approximation error estimator. In the econometric section 3 we describe the Smolyak Kalman, particle, Smolyak particle and Smolyak sum filter, the parallel Metropolis-Hastings algorithm, the convergence diagnostics and the marginal likelihood. In the result section 4 we summarize the performance of the solution method, the filters and the estimators. In section 5 we conclude. In the appendix we present some details on the linearization of the general model and provide a step by step example of a 2-dimensional Smolyak polynomial approximation.

## 2 Economics

### 2.1 Model

An appropriate model for this paper should allow for an increasing number of states and degrees of nonlinearity of the policy functions. We therefore use a multicountry standard growth model and vary the elasticity of intertemporal substitution and the standard deviation of the productivity shock as a proxy for the nonlinearity of the model solution.

The social planner allocates by solving the dynamic optimization

$$\max_{\{c_{n,t}, l_{n,t}, i_{n,t}\}_{n=1}^N}_{t=0}^{\infty} U = E_0 \sum_{t=0}^{\infty} \sum_{n=1}^N \beta^t U_{n,t}$$

for  $n = 1, \dots, N$  countries and all future periods  $t \geq 0$ . The welfare function  $U$  is a discounted sum of country utilities

$$U_{n,t} = \frac{(c_{n,t}^{\theta_n} (1 - l_{n,t})^{1-\theta_n})^{1-\tau_n}}{1 - \tau_n}$$

with discount factor  $\beta$ , elasticity of intertemporal substitutions  $\tau_n$  and consumption and leisure substitution rates  $\theta_n$ . The policy variables are consumption  $c_{n,t}$ , labor  $l_{n,t}$  and investment  $i_{n,t}$  for each country. The world budget constraint

$$\sum_{n=1}^N (y_{n,t} - c_{n,t} - i_{n,t}) = 0$$

restricts the world output  $\sum_n y_{n,t}$  to be either consumed or invested in one of the countries. The production technologies

$$y_{n,t} = e^{a_{n,t}} k_{n,t}^{\alpha_n} l_{n,t}^{1-\alpha_n}$$

depend on productivities  $a_{n,t}$ , capital  $k_{n,t}$  and labor  $l_{n,t}$  and the technical substitution rates  $\alpha_n$ . The capital and productivity transitions are

$$k_{n,t+1} = i_{n,t} + (1 - \delta_n)k_{n,t} - 0.5\kappa_n i_{n,t}^2 \quad (1)$$

$$a_{n,t+1} = \rho_n a_{n,t} + e_{n,t+1} \quad (2)$$

where  $\delta_n$  are the depreciation rates and  $\rho_n$  the autocorrelation coefficients for the productivity processes with normally distributed shocks  $e_{n,t} \sim \mathcal{N}(0, \sigma_{e_n})$  independent across the countries and time. In the capital transition equation (1) we include capital adjustment costs parameterized by  $\kappa_n$ . These costs assure that in the multicountry model the state of the system is not simply the aggregate capital stock but its distribution across the countries.

We have implemented the algorithms for the general model class:

$$0 = f(s_t, x_t, z_t; \theta)$$

$$z_t = E_e h(s_t, x_t, e_{t+1}, s_{t+1}, x_{t+1}; \theta)$$

$$s_{t+1} = g(s_t, x_t, e_{t+1}; \theta).$$

All functions depend on the structural parameters in  $\theta$ . The model is formulated in terms of the first order equilibrium conditions  $f : \mathbb{R}^{d_s+d_x+d_z} \rightarrow \mathbb{R}^{d_x}$ , expected functions  $h : \mathbb{R}^{d_s+d_x+d_e+d_s+d_x} \rightarrow \mathbb{R}^{d_z}$  and state transitions  $g : \mathbb{R}^{d_s+d_x+d_e} \rightarrow \mathbb{R}^{d_s}$ . The variables are states  $s_t \in S \subseteq \mathbb{R}^{d_s}$ , policies  $x_t \in X \subseteq \mathbb{R}^{d_x}$ , expected variables  $z_t \in Z \subseteq \mathbb{R}^{d_z}$  and stochastic shocks  $e_{t+1}$ , which are usually specified to follow a normal distribution  $e_{t+1} \sim \mathcal{N}(0, \Sigma_e)$ . As the structural shocks are typically modeled to be independent we assume a diagonal  $\Sigma_e$  from now on.

Our solution approach is to solve for the policy functions  $x^* : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_x}$  that map states into policies. The algorithm is a function iteration scheme



where we repeatedly solve the first order conditions  $f(s, x^{(k+1)}, E_e h(\dots, x^{(k)}, \dots)) = 0$  for the next  $k + 1$  iteration of the policy  $x^{(k+1)}$  for given expected variables  $z = E_e h(\dots, x^{(k)}, \dots)$  based on the previous policy  $x^{(k)}$  in iteration step  $k$ . This approach has the advantage that it decomposes one big system, when one solves for  $x$  in  $f(s, x, E_e h(\dots, x, \dots)) = 0$ , into several independent smaller ones - for each grid point we solve one small system for all policies. The independence implies that the algorithm can be parallelized. We will discuss this in more detail in section 2.3.4.

Since the numerical integration can be thought as a function approximation of the integrand with a subsequent analytically simple integration, we can apply the Smolyak operator also to Gaussian quadrature of the rational expectations integrals as

$$E_e h(\dots, e, \dots) = \int h(\dots, e, \dots) p(e) de \approx \sum_j w_j h(\dots, e_j, \dots),$$

where the continuous random variable  $e$  and its density  $p(e)$  is essentially discretized into some realizations  $e_j$  with weights  $w_j$ .

The entire problem of solving the model is to approximate the policy function  $x^*(s_t)$  in

$$\begin{aligned} 0 &= f(s_t, x^*(s_t), \sum_j w_j; \theta) \\ z_t &= h(s_t, x^*(s_t), e_{j,t+1}, s_{j,t+1}, x^*(s_{j,t+1}); \theta) \\ s_{j,t+1} &= g(s_t, x^*(s_t), e_{j,t+1}; \theta) \quad \forall j. \end{aligned}$$

The variables and parameters of the example model correspond to the following variables of the general model class:

$$\begin{aligned} s_t &= \{k_{n,t}, a_{n,t}\}_{n=1}^N \\ x_t &= \{c_{n,t}, l_{n,t}, i_{n,t}\}_{n=1}^N \\ e_t &= \{e_{n,t}\}_{n=1}^N \\ \theta &= \{\tau_n, \theta_n, \alpha_n, \delta_n, \rho_n, \kappa_n, \sigma_{e_n}\}_{n=1}^N \cup \{\beta\}. \end{aligned}$$

In the next section we derive the optimality conditions and map them into the general model functions  $f$ ,  $h$  and  $g$ .

## 2.2 Optimality

The Bellman equation for the allocation problem is

$$V_t = \max_{\{c_{n,t}, l_{n,t}, i_{n,t}, k_{n,t+1}\}_{n=1}^N} \sum_{n=1}^N U_{n,t} + \beta \mathbf{E}_t V_{t+1} + \lambda_B \sum_{n=1}^N (y_{n,t} - c_{n,t} - i_{n,t}) + \sum_{n=1}^N \lambda_n (k_{n,t+1} - (1 - \delta_n)k_{n,t} - i_{n,t} + 0.5\kappa_n i_{n,t}^2)$$

where  $V_t \equiv V(k_{1,t}, \dots, k_{N,t}, a_{1,t}, \dots, a_{N,t}; \theta)$ . The first order conditions for  $n = 1, \dots, N$  are given by

$$\frac{\partial V_t}{\partial c_{n,t}} = \frac{\partial U_{n,t}}{\partial c_{n,t}} - \lambda_B = 0 \quad (3)$$

$$\frac{\partial V_t}{\partial l_{n,t}} = \frac{\partial U_{n,t}}{\partial l_{n,t}} + \lambda_B \frac{\partial y_{n,t}}{\partial l_{n,t}} = 0 \quad (4)$$

$$\frac{\partial V_t}{\partial i_{n,t}} = -\lambda_B + \lambda_n (\kappa_n i_{n,t} - 1) = 0 \quad (5)$$

$$\frac{\partial V_t}{\partial k_{n,t+1}} = \beta \mathbf{E}_t \frac{\partial V_{t+1}}{\partial k_{n,t+1}} + \lambda_n = 0 \quad (6)$$

$$\frac{\partial V_t}{\partial \lambda_n} = k_{n,t+1} - (1 - \delta_n)k_{n,t} - i_{n,t} + 0.5\kappa_n i_{n,t}^2 = 0 \quad (7)$$

The last optimality condition is the budget constraint

$$\frac{\partial V_t}{\partial \lambda_B} = \sum_{n=1}^N (y_{n,t} - c_{n,t} - i_{n,t}) = 0. \quad (8)$$

The derivatives of the unknown value functions in equations (6) can be derived by the envelope theorem. It allows us to write the derivatives for  $n = 1, \dots, N$  as

$$\frac{\partial V_t}{\partial k_{n,t}} = \lambda_B \frac{\partial y_{n,t}}{\partial k_{n,t}} - \lambda_n (1 - \delta_n) \quad (9)$$

since the derivatives of the policy variables with respect to  $k_{n,t}$  are zero by optimality. The Lagrange multipliers  $\lambda_n$  in equation (9) can be substituted by equations (5) and the multiplier  $\lambda_B$  by equation (3) to arrive for  $n = 1, \dots, N$  at

$$\frac{\partial V_t}{\partial k_{n,t}} = \frac{\partial U_{n,t}}{\partial c_{n,t}} \left( \frac{\partial y_{n,t}}{\partial k_{n,t}} + \frac{1 - \delta_n}{1 - \kappa_n i_{n,t}} \right).$$

These equations can then be forwarded one period and plugged into equations (6) to obtain the Euler equations for  $n = 1, \dots, N$

$$\frac{1}{1 - \kappa_n i_{n,t}} \frac{\partial U_{n,t}}{\partial c_{n,t}} - \beta E_t \left( \frac{\partial U_{n,t+1}}{\partial c_{n,t+1}} \left( \frac{\partial y_{n,t+1}}{\partial k_{n,t+1}} + \frac{1 - \delta_n}{1 - \kappa_n i_{n,t+1}} \right) \right) = 0. \quad (10)$$

Equations (3) and (4) imply the intratemporal optimality conditions between consumption and labor supply for  $n = 1, \dots, N$

$$\frac{\partial U_{n,t}}{\partial l_{n,t}} + \frac{\partial U_{n,t}}{\partial c_{n,t}} \frac{\partial y_{n,t}}{\partial l_{n,t}} = 0. \quad (11)$$

Finally, we can substitute the Lagrange multiplier  $\lambda_B$  from the first of the  $N$  equations (3) into the other  $N - 1$  equations of (3) to arrive at  $N - 1$  conditions for  $n = 2, \dots, N$

$$\frac{\partial U_{1,t}}{\partial c_{1,t}} = \frac{\partial U_{n,t}}{\partial c_{n,t}}, \quad (12)$$

which enforce equal marginal utilities across all the countries.

The  $4N$  equations which determine the variables  $c_{n,t}, l_{n,t}, i_{n,t}, k_{n,t+1}$  are the  $N$  Euler conditions (10),  $N$  intratemporal trade offs between consumption and labor (11),  $N - 1$  equalities of marginal utilities (12), the budget constraint (8) and  $N$  capital transitions (7). A simplification is to solve the trade off in equations (11) for consumptions  $c_{n,t}$  and the budget constraint (8) for one investment  $i_{1,t}$ .

The mapping into the general model class is the following: The  $2N - 1$  equations (10) and (12) determine the policy variables  $l_{1,t}, \dots, l_{N,t}$  and  $i_{2,t}, \dots, i_{N,t}$  in the general functions  $f$ . The general model functions  $h$  for  $N$  forward looking variables are given by the arguments of the expected values in the Euler equations (10). The  $N$  capital transitions (7) and the  $N$  productivity transitions (2) form the state transition functions  $g$  of the general model.

## 2.3 Solution

### 2.3.1 Approximation

The first step towards the nonlinear solution is to generate start values for the policy functions. We use a linear approximation of the model around the deterministic steady state described in appendix A.

The nonlinear solution or policy functions  $x^*(s)$  reside in the infinite dimensional space of all functions. In a practical approximation we search in the  $m_i$  dimensional space of polynomials  $\hat{x}^*(s; c) = \sum_{j=1}^{m_i} c_j b_{j-1}(s)$  characterized by

the coefficient vector  $c$ . We use orthogonal Chebyshev polynomials as basis functions  $b_j(s)$  defined by  $b_0(s) = 1$ ,  $b_1(s) = s$  and  $b_{j+1}(s) = 2s b_j(s) - b_{j-1}(s)$  for  $j \geq 1$ . To identify the  $m_i$  elements of the coefficient vector  $c$  we use the same number of policy values at the grid  $s^i = \{s_1^i, s_2^i, \dots, s_{m_i}^i\}$ . In a 1-dimensional approximation, for example with  $m_i = 3$  coefficients, we have to solve the linear equations

$$\begin{bmatrix} b_0(s_1^i) & b_1(s_1^i) & b_2(s_1^i) \\ b_0(s_2^i) & b_1(s_2^i) & b_2(s_2^i) \\ b_0(s_3^i) & b_1(s_3^i) & b_2(s_3^i) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} x^*(s_1^i) \\ x^*(s_2^i) \\ x^*(s_3^i) \end{bmatrix}. \quad (13)$$

It requires the approximation on the left hand side to be exact for the policy values on the right hand side at the grid  $\{s_1^i, s_2^i, s_3^i\}$ . This equation can be solved accurately since the orthogonality property of the Chebyshev polynomials guarantees that the basis matrix, defined as the basis polynomials evaluated at the grid on the left hand side, is well conditioned for the calculation of its inverse. A good grid according to the numerical theory is the Gauss-Lobatto grid defined by  $s_1^i = 0$  and  $s_j^i = -\cos(\pi(j-1)/(m_i-1))$  for  $j = 1, \dots, m_i$  and  $i > 1$ . The range of these points is from  $-1$  to  $1$  and the grid in the desired approximation space is obtained by a simple linear transformation.

### 2.3.2 Tensor Operator

If the functions to be approximated or integrated depend on several variables we need a rule how to extend a univariate approximation operator to many dimensions. The usual extension uses the tensor operator. It combines each element of the univariate grids and basis functions with each other.

The univariate approximation operator for function  $x : [0, 1] \rightarrow \mathbb{R}$  is

$$U^i(x) = \sum_{j=1}^{m_i} a_j^i x(s_j^i)$$

where  $i \in \mathbb{N}$  is the approximation level and  $s_j^i \in [-1, 1]$  are the grid points. In case of a function approximation,  $a_j^i \in \mathbb{C}$  for  $j = 1, \dots, J$  are functions of  $s$  and in case of numerical integration  $a_j^i$  are the weights. The Clenshaw-Curtis function  $m_1 = 1$  and  $m_i = 2^{i-1} + 1$  for  $i > 1$  translates the approximation level  $i$  into the polynomial degree of the approximation.

The multidimensional ( $d > 1$ ) approximation operator based on the tensor product  $\otimes$  is defined as

$$(U^{i_1} \otimes \dots \otimes U^{i_d})(x) = \sum_{j_1=1}^{m_{i_1}} \dots \sum_{j_d=1}^{m_{i_d}} (a_{j_1}^{i_1} \otimes \dots \otimes a_{j_d}^{i_d}) x(s_{j_1}^{i_1}, \dots, s_{j_d}^{i_d}).$$

In order to construct this approximation we need  $\prod_{j=1}^d m_{i_j}$  function evaluations  $x(s_{j_1}^{i_1}, \dots, s_{j_d}^{i_d})$ . This establishes the exponentially growing costs of the approximation, a phenomenon called the curse of dimensionality.

The multivariate Chebyshev approximation can be expressed as the matrix equation  $B(s^i)c = x(s^i)$  where the approximation level is given by  $i = \{i_1, i_2, \dots, i_d\}$  and the basis matrix by  $B(s^i) = b(s^{i_1}) \otimes \dots \otimes b(s^{i_d})$ . The grid  $s^i = s^{i_1} \times \dots \times s^{i_d}$  is constructed by the Cartesian product of the univariate grids.

The tensor product based multivariate Gaussian quadrature extends the univariate grids by the Cartesian product. The associated weights are multiplied. Since the Gaussian quadrature can be thought as approximating the integrand by a polynomial with a subsequent trivial integration, the Smolyak operator for the function approximation also applies for the Gaussian quadrature.

### 2.3.3 Smolyak Operator

Complete polynomials give us an intuition about how the exponential growth of tensor product costs can be decreased. A  $q$ th-degree expansion of a  $d$ -dimensional function uses a linear combination of the basis functions

$$\mathcal{P}_{q,d} = \{s_1^{i_1} \dots s_d^{i_d} \mid \sum_{l=1}^d i_l \leq q, 0 \leq i_1, \dots, i_d\}.$$

A second order approximation of a bivariate function is then given by

$$x^*(s_1, s_2) = c_0 + c_1 s_1 + c_2 s_2 + c_3 s_1 s_2 + c_4 s_1^2 + c_5 s_2^2.$$

This is the same polynomial as used in a Taylor expansion. We see that complete polynomials are not based on the tensor product of two second order univariate polynomials

$$\{1, s_1, s_1^2\} \otimes \{1, s_2, s_2^2\} = \{1, s_1, s_2, s_1 s_2, s_1^2, s_2^2, s_1^2 s_2, s_1 s_2^2, s_1^2 s_2^2\}.$$

The insight is that the products  $s_1^2 s_2$ ,  $s_1 s_2^2$  and  $s_1^2 s_2^2$  can be dropped and in terms of asymptotic convergence, the complete polynomials will give us as good an approximation as the tensor product with far fewer elements, see Judd (1998).

But beyond this intuition we do not know how to combine, for example, the univariate Gauss-Lobatto grid points in order to arrive at a multivariate grid for an optimal construction of the basis matrix. A discussion of the approximating properties of the Smolyak operator beyond this intuition can be found in Bungartz and Griebel (2004).

The usual formulation of the Smolyak operator is

$$A_{q,d}(x) = \sum_{q-d+1 \leq |i| \leq q} (-1)^{q-|i|} \binom{d-1}{q-|i|} (U^{i_1} \otimes \dots \otimes U^{i_d}).$$

where  $d$  is the dimensionality of the function to be approximated,  $q$  is the level of approximation and the multiindex  $i$  translates the approximation level into the univariate approximation levels. This formula highlights the fact that the Smolyak operator is a simple linear combination of some lower level tensor products and that a straightforward implementation of this operator is not complicated.

The intuition of the complete polynomials does not give a clue about how to construct the multivariate grid from the univariate grids and thus where to evaluate the complete polynomials optimally. In contrast to this missing link, the Smolyak operator constructs the multivariate grid as a combination of some lower level Cartesian products  $\times$  of the univariate grids  $s^{i_j} = \{s_1^{i_j}, \dots, s_{m_{i_j}}^{i_j}\}$

$$H_{q,d} = \bigcup_{q-d+1 \leq |i| \leq q} (s^{i_1} \times \dots \times s^{i_d}), \quad (14)$$

analogous to the construction of the multivariate polynomial  $A_{q,d}$ .

We illustrate the complicated looking Smolyak formulas in a step by step approximation in appendix B. For alternative presentations of the operator see Krüger and Kübler (2004) and Heiss and Winschel (2008).

For the integration, as in Heiss and Winschel (2008), we use the Kronrod-Patterson univariate grids derived according to Genz and Keister (1996) and combine them by the Smolyak operator for a multivariate integration.

### 2.3.4 Iteration

There are two complications in the approximation of the solution. The first is rational expectations, which we approximate by a Smolyak based Gaussian quadrature. The second is that the functions we want to approximate are the unknown solutions of the functional. Therefore, we cannot obtain function values at the grid by simple function evaluations, as we assume on the right hand side of equation (13). The policy values are given only implicitly and a root finder or a function iteration has to improve a start value. Table 1 summarizes the iterative procedures.

Approximation step 1 and interpolation step 2(a)ii are given for the tensor product but they are analogous for a Smolyak approximation. Efficient algorithms in the literature are often constructed to optimize the interpolation

Table 1: Function Iteration and Root Finding

<p>0. Initial policy: <math>x^{(0)}</math> at grid <math>s^i</math>.</p> <p>1. Approximate policy function <math>c^{(k)} = B(s^i)^{-1}x^{(k)}</math>.</p> <p>2. Rational expectations:</p> <p>(a) For all discrete shock realizations <math>j = 1, \dots, J</math></p> <p>i. State transition: <math>s'_j = g(s^i, x^{(k)}, e'_j)</math></p> <p>ii. Next policy: <math>x'_j = B(s'_j)c^{(k)}</math></p> <p>(b) Expected variables: <math>z = \sum_j w_j h(s^i, x, e'_j, s'_j, x'_j)</math>.</p> <p>3. Iteration:</p> <p>(a) Function iteration:</p> <p>i. Solve for <math>x^{(k+1)}</math> in <math>f(s^i, x^{(k+1)}, z(x^{(k)})) = 0</math></p> <p>ii. Residual: <math>R = x^{(k+1)} - x^{(k)}</math></p> <p>(b) Root finding:</p> <p>i. Residuals: <math>R = f(s^i, x^{(k)}, z(x^{(k)}))</math></p> <p>ii. Iteration: <math>x^{(k+1)} = x^{(k)} - (\partial R / \partial x)^{-1} R</math></p> <p>4. <math>k=k+1</math>, go to 1. until <math>R \approx 0</math></p>
---

step as a combined construction of approximation coefficients and interpolation. Implicit in the coefficient calculation is the basis matrix inversion. In our iterative application we need to invert the basis matrix only once since it is not changed over iterations and structural parameters.

The choice between the function iteration in 3(a) and the root finding algorithm in 3(b) involves a trade off between a few iterations over one big root finding system in 3(b) and iterations over several small systems in 3(a). For example, in a model with  $d_g = 1000$  grid points and  $d_x = 20$  policy functions the complete system to be solved has  $d_x \times d_g = 20,000$  equations. In the function iteration routine, on the other hand, the solution  $x^{(k+1)}$  can be obtained point wise. Instead of solving one large system for all  $d_x \times d_g$  policy values, we solve  $d_g$  systems with  $d_x$  equations. But since the function iteration usually needs more iterations the overall gain is not clear cut. Moreover, analytical

Jacobians are available for the small systems. In JBendge the Jacobians are automatically derived by the symbolic differentiation engine and do not need to be supplied by the user. A restriction of the function iteration scheme is that we need all policy functions to appear in the contemporary policy vector  $x$  in  $f(s, x, z) = 0$  otherwise we can not solve for them (the Jacobians become singular).

But the main advantage of the function iteration is probably that it can be parallelized since the policies at each grid point are independent of each other and the  $d_g$  small systems of size  $d_x$  can be solved on parallel CPUs. By now we solve these systems sequentially with a standard Newton root finder with analytical derivatives and a line search for the optimal step size.

### 2.3.5 Error

An approximation procedure has to be accompanied by error estimates in order to control its accuracy. The exact policy functions would imply zero residuals in the complete state space and any deviation is therefore due to the policy function approximation.

Judd (1992) proposed to normalize the residual for an economic interpretation. Dividing the residual by  $(1 - l_n(s_t))^{(1-\theta_n)(1-\tau_n)}/(\kappa_n i_{n,t} - 1)$  and taking it to the power of  $1/(\theta_n(1 - \tau_n) - 1)$  gives the Euler equation in terms of consumption

$$r^c(s) = c_n(s) - \left( \frac{\beta E_t h(s, x^*(s), e, s', x^*(s'); \theta)}{(1 - l_n(s))^{(1-\theta_n)(1-\tau_n)}/(\kappa_n i_{n,t} - 1)} \right)^{\frac{1}{\theta(1-\tau)-1}}$$

$$s' = g(s, x^*(s), e').$$

The Euler error is finally given by  $r^E = |r^c(s)/c_n(s)|$ . A  $\log_{10}$  error of  $-3$  means that the utility loss due to the approximation is less than one per 1000 dollars.

## 3 Econometrics

The Bayesian estimation based on models  $M = \{M_1, \dots, M_m\}$  can be summarized by the factorization of the joint density

$$p(\omega, y, \theta_{M_i} | M_i) = p(\omega | y, \theta_{M_i}, M_i) p(y | \theta_{M_i}, M_i) p(\theta_{M_i} | M_i).$$

The models can explain the observables by different unobservables, functional forms of their relations and shock distributions. The unobservables include the parameters and the states  $\theta_{M_i} = \{\theta\} \cup \{s\}$ . The functions and the



shock distributions describing the variables of interest  $p(\omega|y, \theta_{M_i}, M_i)$ , the density of observables (or likelihood)  $p(y|\theta_{M_i}, M_i)$  and the prior density of unobservables  $p(\theta_{M_i}|M_i)$  factorize the joint density  $p(\omega, y, \theta_{M_i}|M_i)$ .

The Bayesian estimation can transparently incorporate prior information. It is a decision theoretical framework which integrates the modeled subject and may also contain the researcher's utility. By that it helps to communicate the results as described in Geweke (2005).

The variable of interest is a very useful construction. It is an arbitrary function and can represent specification tests or economically meaningful variables like a welfare function. Combined with an unobserved state density we can obtain test statistics which do not rely on the asymptotic theory but take into account usual small samples in macroeconometrics.

In the Bayesian formula the likelihood contains the evidence in the data and transforms the prior into the posterior density of the unobservables

$$p(\theta_{M_i}|y, M_i) = \frac{p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)}{p(y|M_i)}.$$

The marginal likelihood

$$p(y|M_i) = \int p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)d\theta_{M_i}$$

allows the data to assign probabilities to model  $M_i$

$$p(M_i|y) = \frac{p(y|M_i)p(M_i)}{p(y)} = \frac{p(y|M_i)p(M_i)}{\sum_{j=1}^m p(y|M_j)p(M_j)}.$$

The ratio of two marginal likelihoods is the Bayes factor. It transforms the model prior into the posterior odds ratio

$$\frac{p(M_i|y)}{p(M_j|y)} = \frac{p(M_i)}{p(M_j)} \frac{p(y|M_i)}{p(y|M_j)}.$$

For equal prior probabilities of the models, the marginal likelihood selects the model with the best in-sample forecast quality. Information theory also points out an interesting interpretation of the marginal likelihood as a selection criterion. It is related to Occam's razor and selects the model which allows a higher compression of the explained data and by that needs less bandwidth for a transmission of data and model. This highlights the close relation between modeling and compression. But it also reveals the shortcoming of this selection criterion: it is an in-sample criterion whereas models should also perform well out-of-sample.

However, we later check whether this measure can properly select between the nonlinear and linear approximation of the example model.

### 3.1 Filtering

The model solving policy functions  $x^*(s)$  can be used to substitute the policy variables in the state transition equation. Augmented with a measurement equation, we obtain the model's empirical implication for the observables in terms of a nonlinear state space model

$$\begin{aligned} s_t &= g^*(s_{t-1}, x^*(s_{t-1}), e_t) = g(s_{t-1}, e_t) && \Leftrightarrow p(s_t|s_{t-1}) \\ y_t &= m^*(s_t, x^*(s_{t-1})) + \epsilon_t = m(s_t) + \epsilon_t && \Leftrightarrow p(y_t|s_t) \end{aligned}$$

where we also switch to a more convenient density representation of the model for the econometric discussion: with distributional assumptions for the state and measurement shocks,  $e_t$  and  $\epsilon_t$ , the state space equations can be expressed in terms of state and measurement densities. For notational convenience the conditioning of these densities on the parameter vector  $\theta$  is suppressed.

The filtering approach evaluates the likelihood of this model by the prediction and filtering step. These steps transform the posterior density of unobserved states  $p(s_{t-1}|y_{1:t-1})$  into the next posterior  $p(s_t|y_{1:t})$  by recursively processing new data  $y_t$  for each period  $t$ . As a by-product we get a likelihood value. The notation  $y_{1:t}$  is a shorthand for  $\{y_1, \dots, y_t\}$ .

For a given parameter vector, we start at time 0 with the prior information about the state

$$p(s_0) = p(s_0|y_0).$$

The prediction step generates the prior density according to the Chapman-Kolmogorov equation

$$p(s_t|y_{1:t-1}) = \int p(s_t, s_{t-1}|y_{1:t-1})ds_{t-1} = \int p(s_t|s_{t-1})p(s_{t-1}|y_{1:t-1})ds_{t-1} \quad (15)$$

where the state density  $p(s_t|s_{t-1})$  is weighted by the last posterior density  $p(s_{t-1}|y_{1:t-1})$ . The new posterior is obtained in the filtering step, where new data  $y_t$  allows to update the prediction density

$$p(s_t|y_{1:t}) = \frac{p(s_t, y_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} = \frac{p(y_t|s_t)p(s_t|y_{1:t-1})}{\int p(y_t|s_t)p(s_t|y_{1:t-1})ds_t}. \quad (16)$$

This equation is the result of a repeated application of Bayes' formula. The normalizing constant

$$l_t = \int p(y_t|s_t)p(s_t|y_{1:t-1})ds_t = p(y_t|y_{1:t-1}) \quad (17)$$

is the period's contribution to the likelihood of the complete sample

$$\mathcal{L}(\theta; y_{1:T}) = p(y_{1:T}|\theta) = \prod_{t=1}^T p(y_t|y_{1:t-1}, \theta) = \prod_{t=1}^T l_t. \quad (18)$$

The state posterior density  $p(s_{1:T}|y_{1:T})$  can be the end of an estimation or just a means to obtain the likelihood. The likelihood can then be either maximized over the parameter vector or used to base the inference on the parameter posterior density

$$p(\theta|y_{1:T}) = \frac{p(y_{1:T}|\theta)p(\theta)}{p(y_{1:T})}.$$

The particle filter estimates and updates the complete posterior density of the unobserved states represented by a sample of the states, called particles. This is done by a costly sequential importance sampling with an inaccurate but simple to implement proposal density. This filter is computationally costly because the proposal density does not use the available information from the current observation. Moreover, as a Monte Carlo approach it does not use any information, like the smoothness, of the involved functions.

Our first new filter is the Smolyak Kalman filter which assumes that the prediction  $p(s_t|y_{1:t-1})$  and posterior densities  $p(s_t|y_{1:t})$  are Gaussian. We therefore need to approximate only the first two moments of the densities and can then use the Kalman update in the filtering step. The moments needed for the Kalman step can be calculated as expected values of nonlinearly transformed random variables. Hence, we can use a deterministic Smolyak Gaussian quadrature for the approximation of these moments. This filter is very fast but the price may be an undesirably high approximation error.

The second new filter is the Smolyak particle filter. This approach improves the particle filter by combining it with the Smolyak Kalman filter. We use the posterior densities obtained by the Smolyak Kalman filter, represented by the deterministically integrated first two moments of the states, as a proposal density for the importance sampling of the particle filter. This procedure incorporates the latest information obtained from the data. It combines the advantages of both filters, the accurate but slow sampling particle filter and the potentially inaccurate but fast deterministic filter.

The last filter we present is based on a Gaussian sum approximation of the involved densities. The filter is again very fast and purely deterministic. It effectively runs several Smolyak Kalman filters in parallel.

These new filters are used to assure robustness of the simplest Smolyak Kalman filter and to demonstrate some alternatives to the computationally costly particle filter. A nice overview for the various approaches can be found in Arulampalam, Maskell, Gordon, and Clapp (2002).

### 3.1.1 Smolyak Kalman Filter

The extended Kalman filter linearizes the state space equations and then applies the Kalman (1960) filter. A widely used improvement is the deterministic unscented filter by Julier and Uhlmann (1997).

The idea of the unscented filter is that approximating a density is easier than approximating a function. The unscented filter approximates the first two moments needed for the Kalman update. The approximation is some kind of Gaussian quadrature where the number of grid points is taken to be  $2d + 1$  where  $d$  is the dimension of the integrand. This is an attempt but not a solution to the curse of dimensionality since the curse in terms of the number of grid points is effectively transformed into another analogous curse in terms of the approximation error. As the unscented filter raises the number of points only linearly the effect is that accordingly the accuracy of the numerical integration decreases with the dimensionality and nonlinearity of the integrands. Therefore, the unscented filter's error of the likelihood approximation comes from restricting the approximation to two moments and their ad hoc approximation. The unscented filter is therefore restricted to a low polynomial exactness and a small number of states.

The Smolyak Kalman filter avoids this ad hoc moment approximation and instead uses a Smolyak Gaussian quadrature. The moments are then updated in the usual way by the Kalman gain in the filtering step. An advantage of this procedure compared to the unscented filter is that the approximation level can be chosen according to the problem at hand and that the filter is also useful for other than normally distributed shocks. An approach to non-Gaussian densities is implemented in the so called scaled unscented transform which extends the unscented transform by parameters to control higher moments different from the ones of the Gaussian density.

We assume that the initial state density is  $\mathcal{N}(s_0; s_0, \Sigma_0^s)$ . The notation  $\mathcal{N}(x; \mu, \Sigma)$  is a shorthand for a Gaussian density with argument  $x$ , mean  $\mu$  and covariance  $\Sigma$ . Assuming that the previous posterior density is Gaussian

$$p(s_{t-1}|y_{1:t-1}) = \mathcal{N}(s_{t-1}; s_{t-1|t-1}, \Sigma_{t-1|t-1}^s),$$

the prior density

$$p(s_t|y_{1:t-1}) = \mathcal{N}(g(s_{t-1}, e_t); s_{t|t-1}, \Sigma_{t|t-1}^s)$$

is characterized by its first two moments

$$s_{t|t-1} = \int \int g(s_{t-1}, e_t) \mathcal{N}(s_{t-1}; s_{t-1|t-1}, \Sigma_{t-1|t-1}^s) \mathcal{N}(e_t; 0, \Sigma_e) ds_{t-1} de_t \quad (19)$$

$$\begin{aligned} \Sigma_{t|t-1}^s &= \int \int g(s_{t-1}, e_t) g(s_{t-1}, e_t)^T \mathcal{N}(s_{t-1}; s_{t-1|t-1}, \Sigma_{t-1|t-1}^s) \mathcal{N}(e_t; 0, \Sigma_e) ds_{t-1} de_t \\ &\quad - s_{t|t-1} s_{t|t-1}^T, \end{aligned} \quad (20)$$

where  $T$  denotes a transpose and  $\Sigma_e$  is the covariance matrix of the state shocks. The measurement density of the observables is given by

$$p(y_t | y_{t-1}) = \mathcal{N}(y_t; y_{t|t-1}, \Sigma_{t|t-1}^y)$$

where the expected value is

$$y_{t|t-1} = \int m(s_t) \mathcal{N}(s_t; s_{t|t-1}, \Sigma_{t|t-1}^s) ds_t. \quad (21)$$

and the covariance is

$$\Sigma_{t|t-1}^y = \int m(s_t) m(s_t)^T \mathcal{N}(s_t; s_{t|t-1}, \Sigma_{t|t-1}^s) ds_t + \Sigma_e - y_{t|t-1} y_{t|t-1}^T, \quad (22)$$

where  $\Sigma_e$  is the covariance matrix of the measurement shocks. We also need the covariance between the observed and unobserved variables

$$\Sigma_{t|t-1}^{sy} = \int s_t m(s_t)^T \mathcal{N}(s_t; s_{t|t-1}, \Sigma_{t|t-1}^s) ds_t - s_{t|t-1} y_{t|t-1}^T. \quad (23)$$

The recursion is closed by the filtering step (16) and we obtain the next posterior density

$$\begin{aligned} p(s_t | y_{1:t}) &= \mathcal{N}(s_t; s_{t|t}, \Sigma_{t|t}^s) \\ K_t &= \Sigma_{t|t-1}^{sy} (\Sigma_{t|t-1}^y)^{-1} \\ s_{t|t} &= s_{t|t-1} + K_t (y_t - y_{t|t-1}) \\ \Sigma_{t|t}^s &= \Sigma_{t|t-1}^s - K_t \Sigma_{t|t-1}^y K_t^T \end{aligned} \quad (24)$$

according to a usual Kalman update. The numerical problem to be solved for this filter is the evaluation of the integrals for the expected value and covariance of the state prediction in equations (19) and (20), the expected value of the observables in equation (21), the innovation covariance in equation (22) and the covariance between the states and observations in equation (23). The integrals involved have the form  $\int f(s) \mathcal{N}(s, \mu, \Sigma^s) ds$  and can therefore be approximated by a Smolyak Gaussian quadrature with some nodes  $s^{(i)}$  and weights  $w^{(i)}$  by  $\sum_{i=1}^N w^{(i)} f(s^{(i)})$ .

The integrals for the mean and the variance of the prior density in equation (19) and (20), respectively, can be approximated by the nodes and weights for the joint density of the states and shocks as the weighted sum

$$\begin{aligned} s_{t|t-1} &= \sum_{i=1}^N w^{(i)} g(s_{t-1|t-1}^{(i)}, e_t^{(i)}) \\ \Sigma_{t|t-1}^s &= \sum_{i=1}^N w^{(i)} g(s_{t-1|t-1}^{(i)}, e_t^{(i)}) g(s_{t-1|t-1}^{(i)}, e_t^{(i)})^T - s_{t|t-1} s_{t|t-1}^T. \end{aligned}$$

The assumption of a normal density allows us to generate nodes and weights from the density  $\mathcal{N}(s_t; s_{t|t-1}, \Sigma_{t|t-1}^s)$  and to calculate the moments of the measurement density in equations (21) and (22) as

$$\begin{aligned} y_{t|t-1} &= \sum_{i=1}^N w^{(i)} m(s_{t|t-1}^{(i)}) \\ \Sigma_{t|t-1}^y &= \sum_{i=1}^N w^{(i)} m(s_{t|t-1}^{(i)}) m(s_{t|t-1}^{(i)})^T + \Sigma_\epsilon - y_{t|t-1} y_{t|t-1}^T. \end{aligned}$$

The covariance between states and measurements in equation (23) can be approximated by the sum

$$\Sigma_{t|t-1}^{sy} = \sum_{i=1}^N w^{(i)} s_{t|t-1}^{(i)} m(s_{t|t-1}^{(i)})^T - s_{t|t-1} y_{t|t-1}^T.$$

### 3.1.2 Particle Filter

In the Smolyak Kalman filter we assumed that the state prior and posterior densities are Gaussian. In general they might be nonstandard and even multimodal. The particle filter provides a general approximation of the posterior density

$$p(s_{0:t}|y_{1:t}) \approx \sum_{i=1}^N w_t^{(i)} \delta(s_{0:t} - s_{0:t}^{(i)}) \quad (25)$$

by a sample of states  $\{s_{0:t}^{(i)}\}_{i=1}^N$  and corresponding weights  $\{w_t^{(i)}\}_{i=1}^N$  with  $\sum_i w_t^{(i)} = 1$ , where  $\delta$  is the Dirac delta function defined by  $\int_{-\infty}^{\infty} f(x) \delta(x - a) dx = f(a)$ .

We can use importance sampling to draw a sample  $s_{0:t}^{(i)}$  from an importance density  $q(s_{0:t}|y_{1:t})$  and to calculate the weights in (25) as

$$w_t^{(i)} \propto \frac{p(s_{0:t}|y_{1:t})}{q(s_{0:t}|y_{1:t})}. \quad (26)$$

For a recursive algorithm we need to factorize the importance density

$$q(s_{0:t}|y_{1:t}) = q(s_t|s_{0:t-1}, y_{1:t})q(s_{0:t-1}|y_{1:t-1}). \quad (27)$$

This allows us to augment the previous sample  $\{s_{0:t-1}^{(i)}\}_{i=1}^N$  with a sample of the next state  $s_t^{(i)} \sim q(s_t|s_{0:t-1}, y_{1:t})$ , for  $i = 1, \dots, N$ , to obtain the next sample  $\{s_{0:t}^{(i)}\}_{i=1}^N$ .

In the following we derive two equations for a sequential update of the weights. In the first variant for the particle filter, the posterior density can be written as

$$\begin{aligned} p(s_{0:t}|y_{1:t}) &= \frac{p(y_t|s_{0:t}, y_{1:t-1})p(s_{0:t}|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &= \frac{p(y_t|s_{0:t}, y_{1:t-1})p(s_t|s_{0:t-1}, y_{1:t-1})p(s_{0:t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &= \frac{p(y_t|s_t)p(s_t|s_{t-1})p(s_{0:t-1}|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &\propto p(y_t|s_t)p(s_t|s_{t-1})p(s_{0:t-1}|y_{1:t-1}). \end{aligned} \quad (28)$$

The identities  $p(y_t|s_{0:t}, y_{1:t-1}) = p(y_t|s_t)$  and  $p(s_t|s_{0:t-1}, y_{1:t-1}) = p(s_t|s_{t-1})$  follow from the Markov properties of the processes  $y_t$  and  $s_t$ . By substituting (27) and (28) into (26), the weight equation becomes

$$w_t^{(i)} \propto \frac{p(s_{0:t-1}^{(i)}|y_{1:t-1})}{q(s_{0:t-1}^{(i)}|y_{1:t-1})} \frac{p(y_t|s_t^{(i)})p(s_t^{(i)}|s_{t-1}^{(i)})}{q(s_t^{(i)}|s_{0:t-1}^{(i)}, y_{1:t})},$$

which can be written recursively as

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(y_t|s_t^{(i)})p(s_t^{(i)}|s_{t-1}^{(i)})}{q(s_t^{(i)}|s_{0:t-1}^{(i)}, y_{1:t})}. \quad (29)$$

The importance density is often chosen to depend only on  $s_{t-1}$  and  $y_t$ . This simplifies the proposal to  $q(s_t|s_{0:t-1}, y_{1:t}) = q(s_t|s_{t-1}, y_t)$  and frees us from the need to save the history of the processes  $s_t$  and  $y_t$  during the recursion.

The second representation of the weights will be used to derive the recursive algorithm for the Smolyak particle filter in the next section. Using

$$\begin{aligned} p(s_{0:t-1}|y_{1:t}) &= p(s_{0:t-1}|y_{1:t-1}, y_t) = \frac{p(s_{0:t-1}, y_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\ &= \frac{p(s_{0:t-1}|y_{1:t-1})p(y_t|y_{1:t-1}, s_{0:t-1})}{p(y_t|y_{1:t-1})} \end{aligned}$$

and

$$\begin{aligned}
p(s_t | s_{0:t-1}, y_{1:t}) &= p(s_t | s_{0:t-1}, y_{1:t-1}, y_t) = \frac{p(s_t, y_{1:t-1} | s_{0:t-1}, y_{1:t-1})}{p(y_{1:t-1} | s_{0:t-1}, y_t)} \\
&= p(s_t | s_{0:t-1}, y_t) \frac{p(y_{1:t-1} | s_t, s_{0:t-1}, y_t)}{p(y_{1:t-1} | s_{0:t-1}, y_t)} \\
&= p(s_t | s_{0:t-1}, y_t),
\end{aligned}$$

the density  $p(s_{0:t} | y_{1:t})$  can be factorized as

$$\begin{aligned}
p(s_{0:t} | y_{1:t}) &= p(s_t, s_{0:t-1} | y_{1:t}) = p(s_t | s_{0:t-1}, y_{1:t}) p(s_{0:t-1} | y_{1:t}) \\
&= \frac{p(s_t | s_{0:t-1}, y_t) p(y_t | s_{0:t-1}, y_{1:t-1}) p(s_{0:t-1} | y_{1:t-1})}{p(y_t | y_{1:t-1})} \\
&\propto p(s_t | s_{t-1}, y_t) p(y_t | s_{t-1}) p(s_{0:t-1} | y_{1:t-1}).
\end{aligned}$$

and we obtain the update equation for the weights

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(y_t | s_{t-1}^{(i)}) p(s_t^{(i)} | s_{t-1}^{(i)}, y_t)}{q(s_t^{(i)} | s_{0:t-1}^{(i)}, y_{1:t})}. \quad (30)$$

A serious problem with the particle filter is that after a few iterations, most particles will have weights close to zero. This means that many particles stop contributing to the approximation. A brute force solution is to increase the number of the particles and so to waste most of the computational resources. A more efficient and essentially genetic solution is to resample the particles according to their weights and to reproduce the ones with the highest weights and to drop the others. If we resample in each iteration, the weights update equations (29) and (30) become

$$w_t^{(i)} \propto \frac{p(y_t | s_t^{(i)}) p(s_t^{(i)} | s_{t-1}^{(i)})}{q(s_t^{(i)} | s_{0:t-1}^{(i)}, y_{1:t})} \quad (31)$$

and

$$w_t^{(i)} \propto \frac{p(y_t | s_{t-1}^{(i)}) p(s_t^{(i)} | s_{t-1}^{(i)}, y_t)}{q(s_t^{(i)} | s_{0:t-1}^{(i)}, y_{1:t})}. \quad (32)$$

We are often interested in an estimate of the posterior density  $p(s_t | y_{1:t})$  instead of  $p(s_{0:t} | y_{1:t})$ . Again, the Markov property of  $s_t$  allows us to factorize the posterior density into

$$p(s_{0:t} | y_{1:t}) = p(s_{0:t-1} | y_{1:t-1}) p(s_t | y_{1:t}) = \dots = p(s_0) \prod_{j=1}^t p(s_j | y_{1:j}).$$



Since in

$$p(s_t|y_{1:t}) = \frac{p(s_{0:t}|y_{1:t})}{p(s_0) \prod_{i=1}^{t-1} p(s_i|y_{1:i})}$$

the denominator is constant at  $t$ , it follows that

$$p(s_t|y_{1:t}) \propto p(s_{0:t}|y_{1:t})$$

and we can use the weights given in (31) and (32) to approximate

$$p(s_t|y_{1:t}) \approx \sum_{i=1}^N w_t^{(i)} \delta(s_t - s_t^{(i)}).$$

The simplest choice for the importance density  $q(s_t|s_{0:t-1}, y_{1:t})$  is the state transition  $p(s_t|s_{t-1})$ . It is very easy to sample from this proposal density. Moreover, it simplifies the weights in equation (31) to  $w_t^{(i)} \propto p(y_t|s_t^{(i)})$ . The particles are therefore weighted according to their likelihood. Small measurement errors therefore aggravate the problem of degenerating particles and enforce the need for resampling. Note, that due to this method to calculate the weights, the filter can not handle state space models without measurement errors. Resampling in turn introduces a bias into the particle filter, see for example in Berzuini, Best, Gilks, and Larizza (1997).

The recursion for the particle filter is given by the following three steps. We start with an equally weighted sample from the previous posterior density

$$s_{t-1}^{(i)} \sim p(s_{t-1}|y_{1:t-1}) \quad \text{for } i = 1, \dots, N.$$

1. In the prediction step, the state transition density is used as a proposal density to generate  $N$  particles

$$s_t^{(i)} \sim p(s_t|s_{t-1}^{(i)}) \quad \text{for } i = 1, \dots, N.$$

This is done by evaluating the state transition equation  $g(s_{t-1}^{(i)}, e_t^{(i)})$  for each particle  $s_{t-1}^{(i)}$  with randomly drawn shocks  $e_t^{(i)}$  from the model's state shock distribution.

2. In the filtering step, the particles are weighted with

$$w_t^{(i)} = p(y_t|s_t^{(i)}).$$

This is done by first evaluating the measurement equation  $y_t^{(i)} = m(s_t^{(i)})$ . With additive measurement shocks we know the distribution of the difference to the observed data. In case of Gaussian errors we have  $y_t - y_t^{(i)} \sim \mathcal{N}(\epsilon_t; 0, \Sigma_\epsilon)$ .

3. The particles are then resampled according to their normalized weights

$$\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{n=1}^N w_t^{(n)}}.$$

The resampled particles finally represent an equally weighted sample from the next posterior

$$s_t^{(i)} \sim p(s_t|y_{1:t}) \quad \text{for } i = 1, \dots, N.$$

The period likelihood is then given by the mean of the weights in step 2.

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N p(y_t|s_t^{(i)}) &\approx \int \int p(s_t|s_{t-1})p(s_{t-1}|y_{1:t-1})p(y_t|s_t)ds_tds_{t-1} \\ &= p(y_t|y_{1:t-1}). \end{aligned}$$

These likelihoods are obtained recursively for each period and at the end of the complete sample we arrive at the sample likelihood in equation (18).

The problem of the filter is that the available observation  $y_t$  is not taken into account in the importance density  $p(s_t|s_{t-1})$ . The consequence is that we may sample in very low probability regions of the states density with many implied particle weights close to zero. This is inefficient and the next filter uses a better proposal density.

### 3.1.3 Smolyak Particle Filter

Our idea to improve the particle filter is similar to the one in Amisano and Tristani (2007) where the proposal density for the particle filter is generated by an algorithm similar to the extended Kalman filter. Another related filter is the unscented particle filter by van der Merwe, Doucet, de Freitas, and Wan (2000), where the unscented Kalman filter is used to generate a proposal density.

The information about the current observation is embedded in the posterior of the Smolyak Kalman filter. We use this posterior as the proposal density in the particle filter. This is more accurate than the proposal generated by the extended Kalman filter and more accurate and general than in the unscented filter. The proposal density is now

$$q(s_t|s_{0:t-1}, y_{1:t}) = \tilde{p}(s_t|s_{t-1}^{(i)}, y_t) \approx p(s_t|s_{t-1}^{(i)}, y_t).$$

The weights in equation (32) can be updated as

$$w_t^{(i)} \propto \frac{p(s_t|s_{t-1}^{(i)}, y_t)p(y_t|s_{t-1}^{(i)})}{\tilde{p}(s_t|s_{t-1}^{(i)}, y_t)}.$$

If we do not correct the error resulting from the approximative proposal we can calculate the weights as

$$\tilde{w}_t^{(i)} \propto p(y_t | s_{t-1}^{(i)}).$$

We construct the proposal density  $\tilde{p}(s_t | s_{t-1}^{(i)}, y_t)$  similar to the posterior of the Smolyak Kalman filter starting with an equally weighted sample from the previous posterior density

$$s_{t-1}^{(i)} \sim p(s_{t-1} | y_{1:t-1}) \quad \text{for } i = 1, \dots, N.$$

In the prediction step, we calculate the mean and variance of the state prediction by Gaussian quadrature over the state shocks  $\{e^{(j)}, w^{(j)}\}_{j=1}^J$

$$\begin{aligned} p(s_t | s_{t-1}^{(i)}, y_{1:t-1}) &= \int p(s_t | s_{t-1}^{(i)}) p(s_{t-1}^{(i)} | y_{1:t-1}) de_t \approx \mathcal{N}(s_t; s_{t|t-1}^{(i)}, \Sigma_{t|t-1}^{s(i)}) \\ s_{t|t-1}^{(i)} &= \sum_{j=1}^J w^{(j)} g(s_{t-1}^{(i)}, e^{(j)}) \\ \Sigma_{t|t-1}^{s(i)} &= \sum_{j=1}^J w^{(j)} g(s_{t-1}^{(i)}, e^{(j)}) g(s_{t-1}^{(i)}, e^{(j)})^T - s_{t|t-1} s_{t|t-1}^T \end{aligned}$$

The moments of the observables

$$\begin{aligned} p(y_t | s_{t-1}^{(i)}, y_{1:t-1}) &= \int p(y_t | s_t) p(s_t | s_{t-1}^{(i)}, y_{1:t-1}) de_t \approx \mathcal{N}(y_t; y_{t|t-1}^{(i)}, \Sigma_{t|t-1}^{y(i)}) \quad (33) \\ y_{t|t-1}^{(i)} &= \sum_{j=1}^J w^{(j)} m(g(s_{t-1}^{(i)}, e^{(j)})) \\ \Sigma_{t|t-1}^{y(i)} &= \sum_{j=1}^J w^{(j)} m(g(s_{t-1}^{(i)}, e^{(j)})) m(g(s_{t-1}^{(i)}, e^{(j)}))^T + \Sigma_\epsilon - y_{t|t-1} y_{t|t-1}^T \\ \Sigma_{t|t-1}^{sy(i)} &= \sum_{j=1}^J w^{(j)} g(s_{t-1}^{(i)}, e^{(j)}) m(g(s_{t-1}^{(i)}, e^{(j)}))^T - s_{t|t-1} y_{t|t-1}^T \end{aligned}$$

allow to calculate the Kalman gain and to update the prediction moments to the ones of the proposal density

$$\begin{aligned} p(s_t | s_{t-1}^{(i)}, y_t) &\approx \mathcal{N}(s_t; s_{t|t}^{(i)}, \Sigma_{t|t}^{s(i)}) \\ K_t &= \Sigma_{t|t-1}^{sy} (\Sigma_{t|t-1}^{y(i)})^{-1} \\ s_{t|t}^{(i)} &= s_{t|t-1}^{(i)} + K_t (y_t - y_{t|t-1}^{(i)}) \\ \Sigma_{t|t}^{s(i)} &= \Sigma_{t|t-1}^{s(i)} - K_t \Sigma_{t|t-1}^{y(i)} K_t^T \end{aligned}$$

from which we draw the next state particle:

$$s_t^{(i)} \sim \tilde{p}(s_t | s_{t-1}^{(i)}, y_{1:t})$$

In the filtering step, we weight these new particles according to

$$w_t^{(i)} = p(y_t | s_{t-1}^{(i)}, y_{1:t-1}).$$

If we assume a normal density for  $p(y_t | s_{t-1}^{(i)}, y_{1:t-1})$  we can evaluate the weights at the already calculated density in equation (33)

$$w_t^{(i)} = \mathcal{N}(y_t; y_{t|t-1}, \Sigma_{t|t-1}^y)$$

Finally, we resample the new particles according to their normalized weights

$$\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{n=1}^N w_t^{(n)}}.$$

These steps again transform one posterior into the next one and the period likelihood can be approximated by the mean of the unnormalized weights

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N p(y_t | s_{t-1}^{(i)}) &\approx \int \int p(s_t | s_{t-1}, y_t) p(s_{t-1} | y_{1:t-1}) p(y_t | s_{t-1}) ds_t ds_{t-1} \\ &= p(y_t | y_{1:t-1}). \end{aligned}$$

Note, that the Smolyak particle filter is applicable in state space models without measurement errors since the weights do not need to be evaluated at the measurement shock density.

### 3.1.4 Smolyak Sum Filter

The last filter we present is based on a sum of Gaussian densities to approximate the posterior and prediction densities. The idea can be traced back to Alsbach and Sorenson (1972). More recently Kotecha and Djurić (2003) revived this approach but they use importance sampling to approximate the involved integrals as they do in their Gaussian particle filter. Instead we propose to use Smolyak Gaussian quadrature again.

It seems that some deterministic approaches to nonlinear filtering were already developed in the seventies. They somehow dropped out of fashion and with the development of the particle filtering research turned towards sampling filters. With the advent of powerful deterministic integration schemes it is worth to reiterate on the deterministic filters.

The basic idea is that any density of practical concern can be approximated as a sum of normal densities

$$p(x) \approx \sum_{i=1}^I \omega_i \mathcal{N}(x; \mu_x^i, \Sigma_x^i) \quad \text{with} \quad \sum_{i=1}^I \omega_i = 1.$$

The steps for the filter are similar to the steps of the Smolyak Kalman filter. The difference is that now we effectively run several Smolyak Kalman filters in parallel. The prediction density is approximated as a sum of normal densities

$$\begin{aligned} p(s_t | y_{1:t-1}) &= \int p(s_t | s_{t-1}) p(s_{t-1} | y_{1:t-1}) ds_{t-1} \\ &\approx \int \sum_{i=1}^I \omega_{i;t-1} p(s_t | s_{t-1}) \mathcal{N}(s_{t-1}, s_{t-1|t-1}^i, \Sigma_{s;t-1|t-1}^i) ds_{t-1} \\ &= \sum_{i=1}^I \omega_{i;t-1} \int p(s_t | s_{t-1}) \mathcal{N}(s_{t-1}, s_{t-1|t-1}^i, \Sigma_{s;t-1|t-1}^i) ds_{t-1} \\ &= \sum_{i=1}^I \omega_{i;t-1} \mathcal{N}(s_t, s_{t|t-1}^i, \Sigma_{s;t|t-1}^i). \end{aligned}$$

This is simply a parallel evaluation of several Smolyak Kalman filter steps to calculate the mean  $s_{t|t-1}^i$  and variance  $\Sigma_{s;t|t-1}^i$  according to equations (19) and (20), respectively. Anderson and Moore (1979) present this approach to nonlinear filtering for a model with additive noise in the measurement and state equations. We therefore have an implicit assumption that the weights are preserved during the prediction step. For non additive state shocks this assumption deserves some further elaboration in future research.

The filtering steps involved are again the same as the ones towards the density in equation (24)

$$\begin{aligned} p(s_t | y_{1:t}) &\propto p(y_t | s_t) \mathcal{N}(s_t, s_{t|t-1}^i, \Sigma_{s;t|t-1}^i) \\ &\approx \int \sum_{i=1}^I \omega_{i;t-1} p(s_t | s_{t-1}) \mathcal{N}(s_{t-1}, s_{t|t-1}^i, \Sigma_{s;t|t-1}^i) ds_{t-1} \\ &= \sum_{i=1}^I \int \omega_{i;t-1} p(s_t | s_{t-1}) \mathcal{N}(s_{t-1}, s_{t|t-1}^i, \Sigma_{s;t|t-1}^i) ds_{t-1} \\ &= \sum_{i=1}^I \omega_{i;t} p(s_t | s_{t-1}) \mathcal{N}(s_{t-1}, s_{t|t-1}^i, \Sigma_{s;t|t-1}^i) ds_{t-1}. \end{aligned}$$

The weights are updated according to

$$\omega_{i;t} = \frac{\omega_{i;t-1}\alpha_{i;t}}{\sum_{i=1}^I \omega_{i;t-1}\alpha_{i;t}}$$

where  $\alpha_{i;t}$  is the likelihood contribution of each summand of the Gaussian sum.

This extension to the Smolyak Kalman filter is simple to program and amounts to a parallel evaluation of several Smolyak Kalman filters which allows to parallelize this filter.

## 3.2 Posterior Density

Once the likelihood for a given parameter vector is evaluated we can use it to derive an estimator. The information accumulation in the Bayesian framework is described by Bayes' formula and the object of interest is the posterior of unobservables  $\theta$

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta} \propto p(y|\theta)p(\theta).$$

An analytical expression is available neither for the likelihood nor for the posterior. But a random number generator drawing from this density can provide a histogram as an approximation.

### 3.2.1 Metropolis-Hastings

The Metropolis-Hastings algorithm allows to generate draws from a target density. As opposed to importance sampling no proposal density is needed. The only prerequisite is that the target density can be evaluated at any point of its domain. The algorithm described in Chib and Greenberg (1995) samples so that the histogram of the sequence of draws  $\hat{\theta}_{1:N}$  approximates the target density for large  $N$ .

The algorithm is summarized in table 2. The parameter space is traversed by a random walk. The newly generated candidate parameter vector  $\hat{\theta}_n^*$  is accepted if its posterior is higher than the posterior of the last accepted parameter vector  $\hat{\theta}_{n-1}$ . This qualifies the algorithm as a maximizer. But even if the candidate's posterior is smaller still there is a chance for it to be accepted. This makes the algorithm a global maximizer. The survival according to the parameter vector's fitness, measured by the posterior value, qualifies it as a genetic algorithm. If the acceptance ratio is tuned by the random walk variances to be around 30% we obtain a representative sample from the target density after convergence.

Table 2: Metropolis-Hastings Algorithm

<ol style="list-style-type: none"> <li>1. Choose start value <math>\theta_1</math> and <math>\Sigma_\epsilon</math> for an acceptance ratio of <math>\approx 30\%</math></li> <li>2. For <math>n = 2</math>, while <math>n - J &lt; N</math>, <math>n = n + 1</math> <ol style="list-style-type: none"> <li>(a) Candidate: <math>\hat{\theta}_n^* = \hat{\theta}_{n-1} + \epsilon</math>, where <math>\mathcal{N}(\epsilon; 0, \Sigma_\epsilon)</math>.</li> <li>(b) Acceptance: <math>\hat{\theta}_n = \begin{cases} \hat{\theta}_n^* &amp; \text{if } U(0, 1) \leq \frac{p(y_{1:t} \hat{\theta}_n^*)p(\hat{\theta}_n^*)}{p(y_{1:t} \hat{\theta}_{n-1})p(\hat{\theta}_{n-1})} \\ \hat{\theta}_{n-1} &amp; \text{otherwise} \end{cases}</math></li> <li>(c) Decide on <math>J</math> by diagnostic tests</li> </ol> </li> <li>3. Disregard burn-in draws <math>\hat{\theta}_{1:J}</math>.</li> </ol>
--

The critical choices of the algorithm are the starting value  $\hat{\theta}_0$ , the density to generate candidates  $\hat{\theta}_n^*$  and the number of draws  $N$ . The choice of  $\hat{\theta}_0$  drives the number of draws before convergence. The start value might be far from a representative draw of the target density and many draws are needed to get into the representative region. The distributional choice is often a random walk with normal shocks. For a normal target density the optimal choice of the innovation variance is  $\Sigma_\epsilon = \text{Cov}(\theta)$ . It has to be scaled so that the acceptance ratio is around 0.3. For a normal target density this is achieved by  $\gamma^{RW} = 2.38/\sqrt{D}$  where  $D$  is the number of estimated parameters. Of course the target density and its covariance  $\text{Cov}(\theta)$  are not known because they are the objects of interest.

The chosen variances in  $\Sigma_\epsilon$  influence the region covered by the sequence. Sampling around the mode of the posterior with large variances will generate candidates far from the current value and a low acceptance probability. Smaller variances increase the acceptance ratio but decrease the region being covered so that low probability regions may be undersampled. The recommended acceptance ratio results from the attempt to balance this trade off.

The next section discusses the diagnostic test to decide on convergence. The decision is about the number  $J$  determining how many burn-in draws  $\hat{\theta}_{1:J}$  are to be ignored. Formal convergence tests are an important part of the analysis as well as eye-balling and after some estimations one acquires a visual feeling for convergence.

### 3.2.2 Convergence Test

A convergence test can be based on one subdivided sequence or on several parallel ones. In general we diagnose convergence if the sequences appear to be drawn from the same distribution. Examining only one sequence will result in overly optimistic diagnostic tests. Gelman and Rubin (1992) pointed out that lack of convergence, in many problems, can easily be detected from many but not from one sequence.

In both cases the diagnostic test is calculated from a 3-dimensional tensor  $\hat{\theta}$  of size  $N \times D \times M$  with elements  $\hat{\theta}_{n,m}^d$ .  $D$  is the number of estimated parameters,  $N$  the number of draws and  $M$  the number of sequences.  $\hat{\theta}_{n,m}$  is a  $1 \times D$  vector and represents the  $n^{\text{th}}$  draw in the  $m^{\text{th}}$  sequence and  $\hat{\theta}_{:,m}$  is a  $N \times D$  matrix and represents all draws in sequence  $m$ .

Brooks and Gelman (1998) proposed the multivariate potential scale reduction factor  $R$  as a diagnostic test. The general idea is to check within and between-sequence-variances and diagnose convergence if they are close to each other. The within-sequence-variance is the  $D \times D$  matrix

$$W = \frac{1}{M(N-1)} \sum_{m=1}^M \sum_{n=1}^N (\hat{\theta}_{n,m} - \bar{\theta}_m)' (\hat{\theta}_{n,m} - \bar{\theta}_m)$$

where  $\bar{\theta}_m = \frac{1}{N} \sum_{n=1}^N \hat{\theta}_{n,m}$  is the  $1 \times D$  mean vector in sequence  $m$ .  $W$  is the mean of the variances in each sequence. The between-sequence-variance  $B/N$  is the  $D \times D$  matrix

$$B/N = \frac{1}{M-1} \sum_{m=1}^M (\bar{\theta}_m - \bar{\theta})' (\bar{\theta}_m - \bar{\theta})$$

where  $\bar{\theta} = \frac{1}{M} \sum_{m=1}^M \bar{\theta}_m$  is the  $1 \times D$  mean of all draws. The combined variance can be estimated as

$$V = \frac{N-1}{N} W + \left(1 + \frac{1}{M}\right) B/N.$$

Convergence is detected for similar  $V$  and  $W$ . A distance measure is calculated by the multivariate potential scale reduction factor

$$R = \frac{N-1}{N} + \frac{M+1}{M} \lambda_{max} \quad \text{where } \lambda_{max} = \max_a \frac{a' V a}{a' W a}.$$

$\lambda_{max}$  can be obtained by taking the largest absolute eigenvalue of  $W^{-1} B/N$ . The following conditions for convergence should be checked:  $V$  and  $W$  should be similar and stabilize and  $R$  should be below 1.1.



These test statistics can be calculated recursively after each draw. Once the conditions are met the burn-in sequence length  $J$  is found and the draws thereafter are taken to represent draws from the posterior of structural parameters.

### 3.2.3 Parallel Extension

The variances of the random walk shocks  $\Sigma_\epsilon$  have to be tuned for an acceptance ratio of around 0.3. It is quite demanding to find good values for all parameters simultaneously and usually many costly training sequences are needed. The variances can then be estimated from these runs. Moreover, as in Fernández-Villaverde and Rubio-Ramírez (2004), robustness should be checked by running several sequences with different start vectors.

We propose to run multiple sequences simultaneously and not sequentially. By that we can assure robustness with respect to start values, calculate unbiased convergence diagnostic tests, estimate the innovation variance on the fly and finally implement ideas from evolutionary algorithms to improve the search for the modus of the posterior in the beginning of the sampling. The pseudo code for this parallel Metropolis-Hastings Algorithm is given in table 3.

The problem of choosing all variances of the random walk shocks is reduced in the proposed parallel variant to the choice of only two scalars  $b$  and  $\gamma^{GE}$ . In the estimated models for this paper the optimal parameters were almost identical for the linear and nonlinear estimation. This means that fast linear estimations can be used to tune the parameters for a more expensive nonlinear estimation run.

The parameter draws are again collected in a  $N \times D \times M$  tensor  $\hat{\theta}$  with elements  $\hat{\theta}_{n,m}^d$  with  $M$  sequences of  $N$  draws for  $D$  parameters.

Our proposal is to add another source of innovation for the generation of the candidate draw  $\hat{\theta}_{n,m_i}^*$  in sequence  $m_i$ . One source is common to the random walk algorithm where a random shock is added to the previous parameter draw. The second, additional source of innovation we propose, is the scaled difference between two parameter vectors from randomly chosen sequences  $m_1$  and  $m_2$ . Parameter  $\gamma^{GE}$  and the shock variance  $b$  determine the relative weight of mixing and random walk innovations.

If the variance of the target density is  $\Sigma = \text{Cov}(\theta)$  then the variance of the difference of two population parameter vectors from the sequences  $m_1$  and  $m_2$  is  $E[(\theta_{m_1} - \theta_{m_2})(\theta_{m_1} - \theta_{m_2})'] = 2\Sigma$ . In case of a converged sequence we get by the law of large numbers  $\lim_{N \rightarrow \infty} \sum_{n=1}^N (\hat{\theta}_{n,m_1} - \hat{\theta}_{n,m_2})(\hat{\theta}_{n,m_1} - \hat{\theta}_{n,m_2})' = 2\Sigma$ .

The intuition behind this procedure is that the variance of the difference between two randomly drawn parameters is the optimal one given that the

Table 3: Parallel Metropolis-Hastings Algorithm

<ol style="list-style-type: none"> <li>1. Choose start values <math>\hat{\theta}_{1,m}</math> for all <math>m = 1, \dots, M</math> and <math>b, \gamma^{GE}</math> for an acceptance ratio of <math>\approx 30\%</math></li> <li>2. For <math>n = 1</math>, while <math>n - J &lt; N</math>, <math>n=n+1</math> <ol style="list-style-type: none"> <li>(a) Repeat for <math>m = 1, \dots, M</math> <ol style="list-style-type: none"> <li>i. Draw <math>m_1</math> and <math>m_2</math> such that <math>m_1 \neq m_2 \neq m</math></li> <li>ii. Candidate: <math>\hat{\theta}_m^* = \hat{\theta}_{m_1,n} + \gamma^{GE}(\hat{\theta}_{m_2,n} - \hat{\theta}_{m_1,n}) + \epsilon</math>, <math>\mathcal{N}(\epsilon; 0, bI)</math></li> <li>iii. Acceptance: <math display="block">\hat{\theta}_{n,m} = \begin{cases} \hat{\theta}_m^* &amp; \text{if } U(0, 1) \leq \frac{p(y_{1:t} \hat{\theta}_m^*)p(\hat{\theta}_m^*)}{p(y_{1:t} \hat{\theta}_m)p(\hat{\theta}_m)} \\ \hat{\theta}_{n,m} &amp; \text{otherwise} \end{cases}</math> </li> </ol> </li> <li>(b) Decide on <math>J</math> by diagnostic tests</li> </ol> </li> <li>3. Disregard burn-in draws <math>\hat{\theta}_{1:J,1:M}</math></li> </ol>
--

sequence has converged. Our idea originated from the diagnosis test where the within and between-sequence-variances are examined and convergence is detected when they are of similar size. ter Braak (2006) derives the same candidate by analogy to the global evolutionary optimization algorithm, called differential evolution, by Storn and Price (1997).

A useful by-product of this parallel Metropolis-Hastings algorithm is that it allows a simple parallelization of the code for the estimation on a computer cluster. Each CPU generates only some sequences and the only information they need to exchange is the matrix  $\hat{\theta}_{n,1:M}$  of accepted draws. Its size is only  $D \times M$  so that overhead costs are driven by the synchronicity of the parallel posterior evaluations.

We have implemented these parallel executions in JBendge, thus reducing the computing costs almost proportional to the number of CPUs.

### 3.3 Marginal Likelihood

Model selection is a difficult but important matter and depends usually on a variety of more or less formal criteria. A frequent and rather informal approach is to examine the model's ability to replicate some moments of the data.

One challenge is that models of interest are often not nested and do not emerge from each other through simple parameter restrictions. In practice functional forms, the number of estimated and calibrated parameters, the unobserved states or the shock distributions may differ across candidate models. Consequently classical likelihood ratio tests are not of much help.

Another problem is that models are inherently wrong since they are at most approximations of the reality and are designed to explain some features of the real world in a given application. Landon-Lane (1998) discusses the Bayesian model selection within 1-dimensional linear processes. The non-nested and quasi-true nature of alternative models can be addressed within a Bayesian framework as described in Fernández-Villaverde and Rubio-Ramírez (2004).

According to Berger and Wolpert (1988) the likelihood contains all relevant information. Given some models  $\{M_1, \dots, M_m\}$  with parameter priors and the density of observables, the unobservables can be integrated out to obtain the marginal likelihood

$$p(y|M_i) = \int_{\Theta_{M_i}} p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)d\theta_{M_i}.$$

The parameter posterior is used for inference, conditional on the adequacy of the model whereas the marginal likelihood is used for a criticism of the model in the light of the data.

Most of the work for calculating the marginal likelihood has already been done once the Metropolis-Hastings algorithm converged and generated parameter draws from the posterior density and the associated posterior values. Gelfand and Dey (1994) show that with any density  $h(\theta_{M_i}|M_i)$  we can write

$$\begin{aligned} & E_{p(\theta_{M_i}|y, M_i)} \left( \frac{h(\theta_{M_i}|M_i)}{p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)} \right) \\ &= \int_{\Theta_{M_i}} \frac{h(\theta_{M_i}|M_i)}{p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)} p(\theta_{M_i}|y, M_i) d\theta_{M_i} \\ &= \int_{\Theta_{M_i}} \frac{h(\theta_{M_i}|M_i)}{p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)} \frac{p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)}{\int_{\Theta_{M_i}} p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)d\theta_{M_i}} d\theta_{M_i} \\ &= \frac{\int_{\Theta_{M_i}} h(\theta_{M_i}|M_i)d\theta_{M_i}}{\int_{\Theta_{M_i}} p(y|\theta_{M_i}, M_i)p(\theta_{M_i}|M_i)d\theta_{M_i}} \\ &= p(y|M_i)^{-1}. \end{aligned}$$

According to the last equation all we have to do is to calculate a weighted mean of the Metropolis-Hastings sequence. Geweke (1999) proposes the fol-

lowing procedure. First we calculate the mean and covariance of the parameter draws for each model  $M_i$

$$\bar{\theta}_{M_i} = \frac{1}{N} \sum_{n=1}^N \hat{\theta}_{n,M_i} \quad \Sigma_{M_i} = \frac{1}{N} \sum_{n=1}^N (\hat{\theta}_{n,M_i} - \bar{\theta}_{M_i})(\hat{\theta}_{n,M_i} - \bar{\theta}_{M_i})'.$$

If  $D$  denotes the number of estimated parameters of a model, we define a  $\chi^2$  critical value for quantile  $p$ , to assure robustness over the quantiles  $p = 0.1, \dots, 0.9$  with

$$\Theta_{M_i} = \{\theta : (\theta - \bar{\theta}_{M_i})' \Sigma_{M_i}^{-1} (\theta - \bar{\theta}_{M_i}) \leq \chi_{1-p}^2(D)\}.$$

With density  $h(\cdot)$

$$h(\theta) = p^{-1} (2\pi)^{-\frac{D}{2}} |\Sigma_{M_i}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\theta - \bar{\theta}_{M_i})' \Sigma_{M_i}^{-1} (\theta - \bar{\theta}_{M_i})\right) I_{\Theta_{M_i}}(\theta)$$

where  $I$  is the indicator function with  $I_S(s) = 1$  if  $s \in S$  and 0 otherwise, we can finally estimate the marginal likelihood by

$$\hat{p}(y|M_i) = \left( \frac{1}{N} \sum_{n=1}^N \frac{h(\hat{\theta}_{n,M_i})}{p(y|\hat{\theta}_{n,M_i}, M_i)p(\hat{\theta}_{n,M_i}|M_i)} \right)^{-1}.$$

## 4 Results

In solution section 4.1 we compare the performance of the Smolyak and the tensor operator within the solution algorithm. In estimation section 4.2.1 we compare the likelihood values obtained by the Smolyak Kalman, Smolyak sum, Smolyak particle and the particle filter. Then in section 4.2.2 we estimate the smallest one country model on simulated data to document the overall performance of the algorithms and filters.

All results are calculated on a cluster with 16 Xeon CPUs at 2.7 GHz with hyper threading. The software is the GNU/Linux openSUSE 10.2 operating system, the Java virtual machine 1.6.0 and the 1.4 beta version of JBendge. Since JBendge is completely programmed in Java it is platform independent and runs on any operating system with the Java virtual machine. The Metropolis-Hastings algorithm is parallelized and runs on all 16 CPUs simultaneously in separate threads.

The model is parameterized equally for all countries by  $\alpha = 0.4, \beta = 0.99, \delta = 0.02, \rho = 0.95, \theta = 0.357, \tau = 2.0, \sigma_a = 0.007$ . The solutions for the multicountry models are calculated for  $\kappa = 0.01$ . The estimations are

done only for the smallest one country model where we use the parameterization  $\tau = 50.0$  and  $\sigma_a = 0.035$ . To simplify the estimation we set capital adjustment costs to zero ( $\kappa = 0$ ) and obtain analytical expressions for the steady state values given in appendix A. The parameterization implies the steady states:  $\bar{a} = 0.0, \bar{k} = 23.2683, \bar{c} = 1.28563, \bar{i} = 0.465366, \bar{l} = 0.312104$  and  $\bar{y} = 1.751$ . They are independent of  $\tau$  or  $\sigma_a$ . The missing investment costs also allow to compare our estimations to the ones obtained by Fernández-Villaverde and Rubio-Ramírez (2005).

## 4.1 Solution

The performance of the Smolyak and tensor operators is measured by the number of grid points, the running time for a solution and the maximal absolute Euler error evaluated at 10,000 random points in the approximation space.

We calculate all solutions with the same Smolyak level for the function approximation and for the numerical integration of the rational expectations. The tolerance level for the change of the policy function during the function iteration is set to  $1E-5$ . The bounds of the approximation space are critical parameters of the solution process. We simulate several data sets to find out the regions visited by the system and set the bounds to  $[20;26]$  for capital and  $[-0.06;0.06]$  for productivity.

The tensor approximation is constructed from univariate approximations with at least three points. This is necessary for the approximation to be nonlinear otherwise only linear terms are present. Therefore the number of points for the simplest tensor approximation is given by  $3^d$ , where  $d$  is the number of states. The Smolyak approximation on the other hand starts from the very beginning with nonlinear second order polynomials. For example, the bivariate terms in the Smolyak approximation  $A_{3,2}$  are  $b_0b_0, b_0b_1, b_0b_2, b_1b_0, b_2b_0$ .

Table 4 documents the results. We use Smolyak levels 2,3 and 4 for the models with 4 and 6 states and afterwards only levels 2 and 3. The Smolyak operator is superior to the tensor operator already for a small model with 4 states where the Euler error on a 41 point Smolyak grid is smaller than the error on a tensor grid with 81 points, although the solution time is the same. For the next level with a similar error the Smolyak operator is more than three times faster and uses about five times less points (137 vs. 625).

The efficiency gain for the model with six states is even more dramatic. Here the Smolyak operator needs only 85 compared to 729 points of the tensor operator for a similar approximation accuracy. The running times are accordingly about 4 times lower for the Smolyak operator.

The tensor operator breaks down for models beyond 6 states while the

Table 4: Smolyak and Tensor based Solutions

States	Operator	Points	Error	Time
4	Smolyak	9	6.6E-4	0.3
		41	8.1E-6	2.5
		137	9.3E-7	24.0
	Tensor	81	4.9E-5	2.5
		625	1.8E-7	88.5
6	Smolyak	13	6.2E-4	0.7
		85	5.1E-5	12.5
		389	9.3E-7	201.5
	Tensor	729	6.5E-5	54.09
8	Smolyak	17	5.9E-4	1.3
		145	3.5E-5	29.9
10		21	7.5E-4	2.3
		221	4.0E-5	69.2
12		25	4.4E-4	3.8
		313	4.8E-5	157.8
14		29	4.3E-4	5.7
		421	3.7E-5	339.1
16		33	4.5E-4	8.5
		545	4.0E-5	724.1
18		37	3.7E-4	12.2
		685	2.6E-5	1819.4
20		41	3.3E-4	17.1
		841	1.9E-5	2107.4
22		45	3.3E-4	23.31
		1013	1.7E-5	4087.4

Smolyak operator is still doing fine. The biggest model we are able to solve has 22 states and it takes around 68 minutes for an approximation error of  $1.7E - 5$ .

## 4.2 Estimation

In the next subsection we compare the likelihood values of the Smolyak Kalman, Smolyak particle, Smolyak sum and the particle filter. We simulate data sets of 100 observations starting from the deterministic steady states generated by very accurate nonlinear solutions of the one country model with the states productivity  $a$  and capital  $k$  and one labor decision  $l$ . The observ-

Table 5: Measurement Errors

	small	large
$\sigma_i$	8.66E-4	4.65E-3
$\sigma_l$	1.10E-3	3.12E-3
$\sigma_y$	1.58E-4	1.75E-2

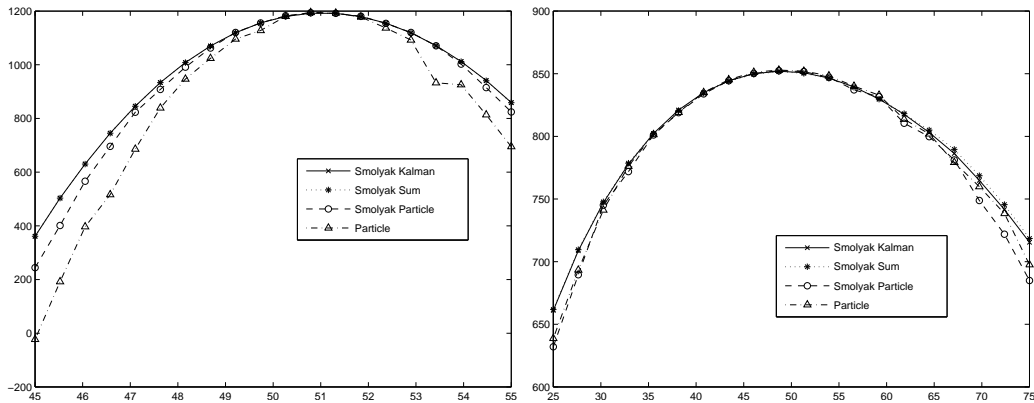
ables in the measurement model are investment  $i$ , labor  $l$  and output  $y$ . The measurement shocks are assumed to be additive.

We present the estimates of two variants of the models, one with small and one with large measurement errors. Their standard deviations are summarized in table 5. The large standard deviations are set to 1% of the steady state values and the small errors to the values in Fernández-Villaverde and Rubio-Ramírez (2005) who use the same model to illustrate and test their algorithms.

#### 4.2.1 Likelihood

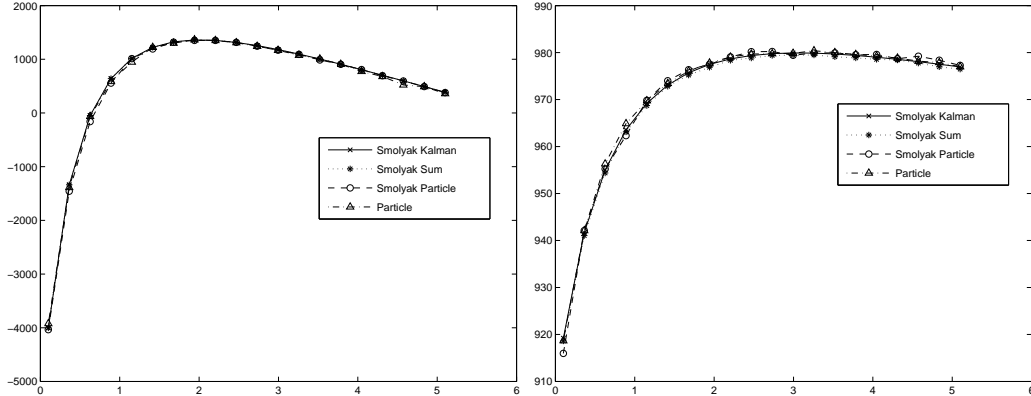
The particle filter is run with 40,000 particles and the Smolyak Kalman filter with integration level 3 for both the time and the measurement steps. The Smolyak particle filter is run with integration level 2 and 500 particles and the Smolyak sum filter with integration level 3 for both the time and the measurement updates and 20 Gaussian summands. All solutions are calculated with level 3 for the policy approximation and the rational expectation integrals.

Figure 1: Likelihood at True Parameters for  $\tau = 50.0$  and  $\sigma_a = 0.035$



Figures 1 and 2 show slices through the multidimensional likelihood. The left plots show likelihood values from the data with small measurement errors

Figure 2: Likelihood at True Parameters for  $\tau = 2.0$  and  $\sigma_a = 0.007$



and the right plots with large ones. We set all parameters to their true values and vary  $\tau$  at the abscissas and plot them against the likelihood values at the ordinates. The results are rather encouraging for all our filters as the values are very similar. It is interesting to see that the particle filter gets into trouble for small measurement errors in the model with  $\tau = 50.0$  and  $\sigma_a = 0.035$  in the left plot of figure 1.

The running times for the filters are very different: the particle filter is hardly useful in combination with a Chebyshev approximation. The construction of the basis matrix for as many as 40,000 particles is very costly and it takes around 120 seconds for one likelihood evaluation. It probably pays off to use a finite element approximation instead, where the trade off is a more costly approximation due to a larger grid, but a less costly likelihood evaluation due to a cheap finite element interpolation. The Smolyak Kalman filter is very fast and it takes around 0.2 seconds for one likelihood evaluation. The Smolyak particle filter is slower and needs around 6 seconds. The Smolyak sum filter is very fast and needs only 0.5 seconds. The likelihood evaluation by a linear Kalman filter takes 0.015 seconds and the extended Kalman filter needs 0.2 seconds.

#### 4.2.2 Parameters

We have implemented an interactive sampling environment in JBendge where the Metropolis-Hastings parameters driving the innovation variance  $\gamma^{GE}$  and  $b$  can be changed while sampling. Together with a regular update of the sequence plots and diagnostic tests the estimation process becomes very flexible and comfortable. Some of the sequences with the lowest posterior values or lowest acceptance ratios can be restarted at the parameter values of the other sequences. This serves the purpose to manually cancel some sequences



Table 6: Bounds of a Uniform Prior

	Low	Hi		Low	Hi
$\alpha$	0.00	1.00	$\sigma_a$	0.0	0.1
$\beta$	0.75	1.00	$\sigma_i$	0.0	0.1
$\delta$	0.00	0.05	$\sigma_l$	0.0	0.1
$\rho$	0.00	1.00	$\sigma_y$	0.0	0.1
$\tau$	0.00	100			
$\theta$	0.00	1.00			

which do not improve anymore. A more advanced global maximization algorithm could do this of course automatically. The second situation when a restart is useful happens while fine tuning the Metropolis-Hastings parameters  $\gamma^{GE}$  and  $b$ . Their effect on the acceptance ratio can usually be inferred from the first few draws and therefore restarts are helpful while searching for the appropriate values.

The usual procedure within our framework has three stages. The first searches for the posterior mode. At this stage the mixing parameter  $\gamma^{GE}$  can be rather large between 0.8 and 2.0 according to the prescription for the differential evolution algorithm of Storn and Price (1997). During this stage the acceptance ratio is usually very low even for parameters  $\gamma^{GE}$  and  $b$  which would later achieve the needed acceptance ratio. There is therefore no use to fine tune these parameters at this stage. During the second stage we sample until the diagnostic tests signal convergence and we also fine tune the parameters  $\gamma^{GE}$  and  $b$  to obtain an acceptance ratio around 0.3. For all estimations parameter  $b$  is set to  $1E - 6$  and  $\gamma^{GE}$  between 0.1 and 0.4. Both parameters are remarkably stable especially across the linear and nonlinear estimations. This helps to find the appropriate values for the nonlinear estimation by fast linear estimation runs. Once the convergence of the sampler is detected we get into the third stage where we sample 50,000 draws.

We run the estimation with around two times more sequences than parameters. The estimated model has 13 parameters and we therefore use 32 sequences for a balanced work load on our 16 CPUs. The usual CPU load on our cluster after convergence is around 800% for the nonlinear and 1500% for the linear estimation. A CPU load of 100% corresponds to one CPU. Since the sequences have to be synchronized and the running times of posterior density evaluations for the parallel draws are different, an exactly linear scaling cannot be expected. However, it is still a dramatic improvement over a run on a single CPU.

For all estimations the Metropolis-Hastings sequences were initiated at random draws from the prior densities. They are taken to be uniform within the bounds in table 6.

It takes around 1000 to 2000 draws for each of the 32 sequences to find the mode of the posterior density and another 1000 to 4000 draws to detect the convergence according to the R, V and W statistics. We sometimes restarted half of the sequences during the maximization process. The complete estimation process takes around 5 minutes for the linear estimation, around 2 hours for the nonlinear estimation with the Smolyak Kalman filter, 4 with the Smolyak sum filter and 20 with the Smolyak particle filter. We only present estimations with these three filters since the particle filter is much too slow to be of practical use. In all of the following tables the "Mean" column shows the mean and "SD" the standard deviation of the posterior density while the "ML" column shows the maximum likelihood estimates. The numbers themselves are coded as  $x_y \equiv x \times 10^y$ .

We will report only estimates of the model with  $\tau = 50$  and  $\sigma_a = 0.35$  for small and large measurement errors. The data is generated with a very accurate solution with integration and solution approximation level 5 and a function iteration tolerance of  $1E - 10$ . The approximation error is around  $1E - 10$ .

Table 7 shows estimations with small measurement errors. The solution and integration Smolyak levels are set to 4 for the rational expectations and the filter integrals. On the left side we report the estimates obtained by the Kalman filter and on the right side we have the estimates from the Smolyak Kalman filter. The likelihood value for the Kalman filter is -54,201 at the true parameters. The minimal and maximal likelihoods obtained during the Metropolis-Hastings sampling after convergence are 1077 and 1098, respectively. The values for the Smolyak Kalman filter are 1197 at the true parameters, 1185 at minimum and 1200 at maximum. The estimates clearly indicate the superior performance of the nonlinear filter. While the mean of the nonlinear posterior estimates and the maximum likelihood values are close to the true parameters of the data generating process, the Kalman filter shows clear biases. The nonlinear filter has problems to accurately estimate the parameter  $\tau$  which is biased and exhibits a large standard deviation of the posterior. The measurement error standard deviations of output and investment are hardly identified with large standard errors  $4.0_{-4}$  and  $2.8_{-4}$  of the same magnitude as the estimates themselves. The standard deviation of the labor measurement error is estimated more accurately. Figure 3 shows the posterior density estimated by the Smolyak Kalman filter. Black vertical bars indicate the true parameter values.

Table 8 shows on the left side estimates with solution, rational expectations

Table 7: Kalman and Smolyak Kalman Filter, Small Errors

	True	Mean	SD	ML	Mean	SD	ML
$\theta$	3.57 <sub>-1</sub>	3.496 <sub>-1</sub>	1.9 <sub>-3</sub>	3.487 <sub>-1</sub>	3.569 <sub>-1</sub>	2.6 <sub>-3</sub>	3.572 <sub>-1</sub>
$\beta$	9.90 <sub>-1</sub>	9.957 <sub>-1</sub>	5.5 <sub>-4</sub>	9.960 <sub>-1</sub>	9.906 <sub>-1</sub>	4.4 <sub>-3</sub>	9.933 <sub>-1</sub>
$\alpha$	4.00 <sub>-1</sub>	3.839 <sub>-1</sub>	5.1 <sub>-3</sub>	3.815 <sub>-1</sub>	4.003 <sub>-1</sub>	6.7 <sub>-3</sub>	4.010 <sub>-1</sub>
$\rho$	9.50 <sub>-1</sub>	9.641 <sub>-1</sub>	3.0 <sub>-3</sub>	9.659 <sub>-1</sub>	9.506 <sub>-1</sub>	1.6 <sub>-3</sub>	9.523 <sub>-1</sub>
$\delta$	2.00 <sub>-2</sub>	1.796 <sub>-2</sub>	1.2 <sub>-3</sub>	1.737 <sub>-2</sub>	1.957 <sub>-2</sub>	5.3 <sub>-4</sub>	1.934 <sub>-2</sub>
$\tau$	5.00 <sub>+1</sub>	1.946 <sub>+1</sub>	2.7 <sub><math>\pm</math>0</sub>	1.784 <sub>+1</sub>	5.255 <sub>+1</sub>	7.6 <sub><math>\pm</math>0</sub>	4.447 <sub>+1</sub>
$\sigma_a$	3.50 <sub>-2</sub>	3.716 <sub>-2</sub>	2.7 <sub>-3</sub>	3.798 <sub>-2</sub>	3.604 <sub>-2</sub>	1.7 <sub>-3</sub>	3.704 <sub>-2</sub>
$\sigma_y$	1.58 <sub>-4</sub>	8.521 <sub>-4</sub>	7.0 <sub>-4</sub>	1.730 <sub>-4</sub>	8.724 <sub>-4</sub>	4.0 <sub>-4</sub>	1.061 <sub>-3</sub>
$\sigma_i$	8.66 <sub>-4</sub>	1.996 <sub>-3</sub>	3.4 <sub>-4</sub>	2.030 <sub>-3</sub>	4.981 <sub>-4</sub>	2.8 <sub>-4</sub>	4.057 <sub>-4</sub>
$\sigma_l$	1.10 <sub>-3</sub>	1.438 <sub>-3</sub>	1.1 <sub>-4</sub>	1.395 <sub>-3</sub>	1.199 <sub>-3</sub>	8.4 <sub>-5</sub>	1.168 <sub>-3</sub>

Table 8: Smolyak Kalman Filters, Small Errors

	True	Mean	SD	ML	Mean	SD	ML
$\theta$	3.57 <sub>-1</sub>	3.586 <sub>-1</sub>	2.6 <sub>-3</sub>	3.596 <sub>-1</sub>	3.436 <sub>-1</sub>	2.4 <sub>-3</sub>	3.430 <sub>-1</sub>
$\beta$	9.90 <sub>-1</sub>	9.914 <sub>-1</sub>	4.7 <sub>-3</sub>	9.948 <sub>-1</sub>	9.529 <sub>-1</sub>	7.9 <sub>-3</sub>	9.608 <sub>-1</sub>
$\alpha$	4.00 <sub>-1</sub>	4.047 <sub>-1</sub>	6.7 <sub>-3</sub>	4.074 <sub>-1</sub>	3.641 <sub>-1</sub>	6.8 <sub>-3</sub>	3.625 <sub>-1</sub>
$\rho$	9.50 <sub>-1</sub>	9.501 <sub>-1</sub>	1.9 <sub>-3</sub>	9.513 <sub>-1</sub>	9.552 <sub>-1</sub>	1.2 <sub>-3</sub>	9.559 <sub>-1</sub>
$\delta$	2.00 <sub>-2</sub>	1.996 <sub>-2</sub>	6.1 <sub>-4</sub>	2.004 <sub>-2</sub>	1.936 <sub>-2</sub>	7.4 <sub>-4</sub>	1.889 <sub>-2</sub>
$\tau$	5.00 <sub>+1</sub>	5.516 <sub>+1</sub>	1.1 <sub>+1</sub>	4.690 <sub>+1</sub>	6.708 <sub>+1</sub>	8.3 <sub><math>\pm</math>0</sub>	5.901 <sub>+1</sub>
$\sigma_a$	3.50 <sub>-2</sub>	3.700 <sub>-2</sub>	1.8 <sub>-3</sub>	3.790 <sub>-2</sub>	3.802 <sub>-2</sub>	2.6 <sub>-3</sub>	3.821 <sub>-2</sub>
$\sigma_y$	1.58 <sub>-4</sub>	8.012 <sub>-4</sub>	4.1 <sub>-4</sub>	1.080 <sub>-3</sub>	1.551 <sub>-3</sub>	5.1 <sub>-4</sub>	1.833 <sub>-3</sub>
$\sigma_i$	8.66 <sub>-4</sub>	5.291 <sub>-4</sub>	2.8 <sub>-4</sub>	3.087 <sub>-4</sub>	6.005 <sub>-4</sub>	4.0 <sub>-4</sub>	4.822 <sub>-5</sub>
$\sigma_l$	1.10 <sub>-3</sub>	1.194 <sub>-3</sub>	8.3 <sub>-5</sub>	1.160 <sub>-3</sub>	1.211 <sub>-3</sub>	9.5 <sub>-5</sub>	1.204 <sub>-3</sub>

Table 9: Smolyak Sum and Smolyak Particle Filters, Small Errors

	True	Mean	SD	ML	Mean	SD	ML
$\theta$	3.57 <sub>-1</sub>	3.593 <sub>-1</sub>	2.0 <sub>-3</sub>	3.611 <sub>-1</sub>	3.560 <sub>-1</sub>	3.7 <sub>-3</sub>	3.607 <sub>-1</sub>
$\beta$	9.90 <sub>-1</sub>	9.927 <sub>-1</sub>	4.5 <sub>-3</sub>	9.968 <sub>-1</sub>	9.870 <sub>-1</sub>	6.8 <sub>-3</sub>	9.952 <sub>-1</sub>
$\alpha$	4.00 <sub>-1</sub>	4.063 <sub>-1</sub>	5.1 <sub>-3</sub>	4.112 <sub>-1</sub>	3.978 <sub>-1</sub>	9.9 <sub>-3</sub>	4.103 <sub>-1</sub>
$\rho$	9.50 <sub>-1</sub>	9.505 <sub>-1</sub>	2.4 <sub>-3</sub>	9.513 <sub>-1</sub>	9.503 <sub>-1</sub>	1.5 <sub>-3</sub>	9.498 <sub>-1</sub>
$\delta$	2.00 <sub>-2</sub>	2.011 <sub>-2</sub>	6.3 <sub>-4</sub>	2.043 <sub>-2</sub>	1.973 <sub>-2</sub>	6.7 <sub>-4</sub>	2.047 <sub>-2</sub>
$\tau$	5.00 <sub>+1</sub>	5.372 <sub>+1</sub>	1.5 <sub>+1</sub>	4.463 <sub>+1</sub>	5.467 <sub>+1</sub>	6.9 <sub><math>\pm</math>0</sub>	5.244 <sub>+1</sub>
$\sigma_a$	3.50 <sub>-2</sub>	3.647 <sub>-2</sub>	1.9 <sub>-3</sub>	3.765 <sub>-2</sub>	3.619 <sub>-2</sub>	1.8 <sub>-3</sub>	3.704 <sub>-2</sub>
$\sigma_y$	1.58 <sub>-4</sub>	9.368 <sub>-4</sub>	3.3 <sub>-4</sub>	1.080 <sub>-3</sub>	9.005 <sub>-4</sub>	3.6 <sub>-4</sub>	1.124 <sub>-3</sub>
$\sigma_i$	8.66 <sub>-4</sub>	4.941 <sub>-4</sub>	2.6 <sub>-4</sub>	2.873 <sub>-4</sub>	5.201 <sub>-4</sub>	2.8 <sub>-4</sub>	3.638 <sub>-4</sub>
$\sigma_l$	1.10 <sub>-3</sub>	1.191 <sub>-3</sub>	9.0 <sub>-5</sub>	1.141 <sub>-3</sub>	1.184 <sub>-3</sub>	9.2 <sub>-5</sub>	1.119 <sub>-3</sub>

Table 10: Kalman and Extended Kalman Filters, Small Errors

	True	Mean	SD	ML	Mean	SD	ML
$\theta$	3.57 <sub>-1</sub>	3.496 <sub>-1</sub>	1.9 <sub>-3</sub>	3.487 <sub>-1</sub>	3.526 <sub>-1</sub>	1.6 <sub>-3</sub>	3.515 <sub>-1</sub>
$\beta$	9.90 <sub>-1</sub>	9.957 <sub>-1</sub>	5.5 <sub>-4</sub>	9.960 <sub>-1</sub>	9.950 <sub>-1</sub>	2.8 <sub>-4</sub>	9.950 <sub>-1</sub>
$\alpha$	4.00 <sub>-1</sub>	3.839 <sub>-1</sub>	5.1 <sub>-3</sub>	3.815 <sub>-1</sub>	3.890 <sub>-1</sub>	4.2 <sub>-3</sub>	3.861 <sub>-1</sub>
$\rho$	9.50 <sub>-1</sub>	9.641 <sub>-1</sub>	3.0 <sub>-3</sub>	9.659 <sub>-1</sub>	9.631 <sub>-1</sub>	1.0 <sub>-3</sub>	9.633 <sub>-1</sub>
$\delta$	2.00 <sub>-2</sub>	1.796 <sub>-2</sub>	1.2 <sub>-3</sub>	1.737 <sub>-2</sub>	1.817 <sub>-2</sub>	8.0 <sub>-4</sub>	1.773 <sub>-2</sub>
$\tau$	5.00 <sub>+1</sub>	1.946 <sub>+1</sub>	2.7 <sub><math>\pm</math>0</sub>	1.784 <sub>+1</sub>	2.045 <sub>+1</sub>	1.5 <sub><math>\pm</math>0</sub>	1.982 <sub>+1</sub>
$\sigma_a$	3.50 <sub>-2</sub>	3.716 <sub>-2</sub>	2.7 <sub>-3</sub>	3.798 <sub>-2</sub>	3.517 <sub>-2</sub>	2.2 <sub>-3</sub>	3.408 <sub>-2</sub>
$\sigma_y$	1.58 <sub>-4</sub>	8.521 <sub>-4</sub>	7.0 <sub>-4</sub>	1.730 <sub>-4</sub>	8.735 <sub>-4</sub>	6.7 <sub>-4</sub>	1.044 <sub>-5</sub>
$\sigma_i$	8.66 <sub>-4</sub>	1.996 <sub>-3</sub>	3.4 <sub>-4</sub>	2.030 <sub>-3</sub>	1.673 <sub>-3</sub>	3.0 <sub>-4</sub>	1.844 <sub>-3</sub>
$\sigma_l$	1.10 <sub>-3</sub>	1.438 <sub>-3</sub>	1.1 <sub>-4</sub>	1.395 <sub>-3</sub>	1.448 <sub>-3</sub>	1.0 <sub>-4</sub>	1.411 <sub>-3</sub>

Table 11: Smolyak Sum and Smolyak Particle Filters, Large Errors

	True	Mean	SD	ML	Mean	SD	ML
$\theta$	3.57 <sub>-1</sub>	3.562 <sub>-1</sub>	5.2 <sub>-3</sub>	3.572 <sub>-1</sub>	3.543 <sub>-1</sub>	6.8 <sub>-3</sub>	3.575 <sub>-1</sub>
$\beta$	9.90 <sub>-1</sub>	9.873 <sub>-1</sub>	7.7 <sub>-3</sub>	9.822 <sub>-1</sub>	9.864 <sub>-1</sub>	8.6 <sub>-3</sub>	9.897 <sub>-1</sub>
$\alpha$	4.00 <sub>-1</sub>	3.979 <sub>-1</sub>	1.4 <sub>-2</sub>	4.007 <sub>-1</sub>	3.928 <sub>-1</sub>	1.8 <sub>-2</sub>	4.016 <sub>-1</sub>
$\rho$	9.50 <sub>-1</sub>	9.437 <sub>-1</sub>	8.4 <sub>-3</sub>	9.400 <sub>-1</sub>	9.472 <sub>-1</sub>	8.9 <sub>-3</sub>	9.444 <sub>-1</sub>
$\delta$	2.00 <sub>-2</sub>	2.038 <sub>-2</sub>	2.2 <sub>-3</sub>	2.053 <sub>-2</sub>	1.966 <sub>-2</sub>	3.0 <sub>-3</sub>	2.085 <sub>-2</sub>
$\tau$	5.00 <sub>+1</sub>	5.719 <sub>+1</sub>	1.9 <sub>+1</sub>	8.012 <sub>+1</sub>	5.403 <sub>+1</sub>	2.0 <sub>+1</sub>	4.873 <sub>+1</sub>
$\sigma_a$	3.50 <sub>-2</sub>	3.357 <sub>-2</sub>	2.7 <sub>-3</sub>	3.303 <sub>-2</sub>	3.385 <sub>-2</sub>	2.4 <sub>-3</sub>	3.422 <sub>-2</sub>
$\sigma_y$	8.75 <sub>-3</sub>	1.680 <sub>-2</sub>	2.1 <sub>-3</sub>	1.597 <sub>-2</sub>	1.696 <sub>-2</sub>	2.1 <sub>-3</sub>	1.745 <sub>-2</sub>
$\sigma_i$	2.33 <sub>-3</sub>	4.969 <sub>-3</sub>	2.5 <sub>-3</sub>	5.671 <sub>-3</sub>	4.726 <sub>-3</sub>	2.4 <sub>-3</sub>	5.044 <sub>-3</sub>
$\sigma_l$	1.56 <sub>-3</sub>	2.873 <sub>-3</sub>	2.1 <sub>-4</sub>	2.700 <sub>-3</sub>	2.882 <sub>-3</sub>	2.1 <sub>-4</sub>	2.888 <sub>-3</sub>

Table 12: Smolyak Sum Filters, Large Errors

	True	Mean	SD	ML	Mean	SD	ML
$\theta$	3.57 <sub>-1</sub>	3.570 <sub>-1</sub>	5.9 <sub>-3</sub>	3.553 <sub>-1</sub>	3.558 <sub>-1</sub>	6.6 <sub>-3</sub>	3.551 <sub>-1</sub>
$\beta$	9.90 <sub>-1</sub>	9.885 <sub>-1</sub>	7.4 <sub>-3</sub>	9.845 <sub>-1</sub>	9.873 <sub>-1</sub>	6.8 <sub>-3</sub>	9.813 <sub>-1</sub>
$\alpha$	4.00 <sub>-1</sub>	3.998 <sub>-1</sub>	1.5 <sub>-2</sub>	3.954 <sub>-1</sub>	3.966 <sub>-1</sub>	1.7 <sub>-2</sub>	3.947 <sub>-1</sub>
$\rho$	9.50 <sub>-1</sub>	9.450 <sub>-1</sub>	8.5 <sub>-3</sub>	9.450 <sub>-1</sub>	9.451 <sub>-1</sub>	9.3 <sub>-3</sub>	9.436 <sub>-1</sub>
$\delta$	2.00 <sub>-2</sub>	2.023 <sub>-2</sub>	2.9 <sub>-3</sub>	1.945 <sub>-2</sub>	2.014 <sub>-2</sub>	3.1 <sub>-3</sub>	2.060 <sub>-2</sub>
$\tau$	5.00 <sub>+1</sub>	6.060 <sub>+1</sub>	2.0 <sub>+1</sub>	7.046 <sub>+1</sub>	5.777 <sub>+1</sub>	1.9 <sub>+1</sub>	6.039 <sub>+1</sub>
$\sigma_a$	3.50 <sub>-2</sub>	3.364 <sub>-2</sub>	2.3 <sub>-3</sub>	3.345 <sub>-2</sub>	3.373 <sub>-2</sub>	2.5 <sub>-3</sub>	3.283 <sub>-2</sub>
$\sigma_y$	8.75 <sub>-3</sub>	1.671 <sub>-2</sub>	2.2 <sub>-3</sub>	1.742 <sub>-2</sub>	1.738 <sub>-2</sub>	2.0 <sub>-3</sub>	1.739 <sub>-2</sub>
$\sigma_i$	2.33 <sub>-3</sub>	5.027 <sub>-3</sub>	2.6 <sub>-3</sub>	3.332 <sub>-3</sub>	4.393 <sub>-3</sub>	2.5 <sub>-3</sub>	4.414 <sub>-3</sub>
$\sigma_l$	1.56 <sub>-3</sub>	2.878 <sub>-3</sub>	2.2 <sub>-4</sub>	2.951 <sub>-3</sub>	2.872 <sub>-3</sub>	2.1 <sub>-4</sub>	2.866 <sub>-3</sub>

Table 13: Kalman Filter, Linear Model, Small Errors

	True	Mean	SD	ML	Mean	SD	ML
$\theta$	3.57 <sub>-1</sub>	3.615 <sub>-1</sub>	3.0 <sub>-3</sub>	3.583 <sub>-1</sub>	3.615 <sub>-1</sub>	3.0 <sub>-3</sub>	3.583 <sub>-1</sub>
$\beta$	9.90 <sub>-1</sub>	9.885 <sub>-1</sub>	1.5 <sub>-3</sub>	9.903 <sub>-1</sub>	9.885 <sub>-1</sub>	1.5 <sub>-3</sub>	9.903 <sub>-1</sub>
$\alpha$	4.00 <sub>-1</sub>	4.125 <sub>-1</sub>	7.8 <sub>-3</sub>	4.040 <sub>-1</sub>	4.125 <sub>-1</sub>	7.8 <sub>-3</sub>	4.040 <sub>-1</sub>
$\rho$	9.50 <sub>-1</sub>	9.451 <sub>-1</sub>	5.0 <sub>-3</sub>	9.509 <sub>-1</sub>	9.451 <sub>-1</sub>	5.0 <sub>-3</sub>	9.509 <sub>-1</sub>
$\delta$	2.00 <sub>-2</sub>	2.267 <sub>-2</sub>	1.8 <sub>-3</sub>	2.063 <sub>-2</sub>	2.267 <sub>-2</sub>	1.8 <sub>-3</sub>	2.063 <sub>-2</sub>
$\tau$	5.00 <sub>+1</sub>	6.086 <sub>+1</sub>	2.0 <sub>+1</sub>	3.786 <sub>+1</sub>	6.086 <sub>+1</sub>	2.0 <sub>+1</sub>	3.786 <sub>+1</sub>
$\sigma_a$	3.50 <sub>-2</sub>	3.260 <sub>-2</sub>	2.2 <sub>-3</sub>	3.155 <sub>-2</sub>	3.260 <sub>-2</sub>	2.2 <sub>-3</sub>	3.155 <sub>-2</sub>
$\sigma_y$	1.58 <sub>-4</sub>	8.753 <sub>-4</sub>	4.1 <sub>-4</sub>	2.369 <sub>-4</sub>	8.753 <sub>-4</sub>	4.1 <sub>-4</sub>	2.369 <sub>-4</sub>
$\sigma_i$	8.66 <sub>-4</sub>	5.518 <sub>-4</sub>	2.8 <sub>-4</sub>	8.196 <sub>-4</sub>	5.518 <sub>-4</sub>	2.8 <sub>-4</sub>	8.196 <sub>-4</sub>
$\sigma_l$	1.10 <sub>-3</sub>	1.145 <sub>-3</sub>	8.5 <sub>-5</sub>	1.095 <sub>-3</sub>	1.145 <sub>-3</sub>	8.5 <sub>-5</sub>	1.095 <sub>-3</sub>

Table 14: Poor identification of  $\tau$

	Smallest $\tau$	True Values	Changes
$\theta$	3.581 <sub>-1</sub>	3.570 <sub>-1</sub>	3.570 <sub>-1</sub>
$\beta$	9.964 <sub>-1</sub>	9.900 <sub>-1</sub>	<b>9.964</b> <sub>-1</sub>
$\alpha$	4.028 <sub>-1</sub>	4.000 <sub>-1</sub>	4.000 <sub>-1</sub>
$\rho$	9.552 <sub>-1</sub>	9.500 <sub>-1</sub>	<b>9.552</b> <sub>-1</sub>
$\delta$	1.911 <sub>-2</sub>	2.000 <sub>-2</sub>	<b>1.911</b> <sub>-2</sub>
$\tau$	3.188 <sub>+1</sub>	3.188 <sub>+1</sub>	3.188 <sub>+1</sub>
$\sigma_a$	3.919 <sub>-2</sub>	3.500 <sub>-2</sub>	3.500 <sub>-2</sub>
$\sigma_y$	1.149 <sub>-3</sub>	1.580 <sub>-4</sub>	1.580 <sub>-4</sub>
$\sigma_i$	3.173 <sub>-4</sub>	8.660 <sub>-4</sub>	8.660 <sub>-4</sub>
$\sigma_l$	1.260 <sub>-3</sub>	1.100 <sub>-3</sub>	1.100 <sub>-3</sub>
	Log Likelihood		
Smolyak Kalman	1193	-8167	1164
Particle	1189	-10761	1159

Figure 3: Parameter Posterior, Smolyak Kalman Filter

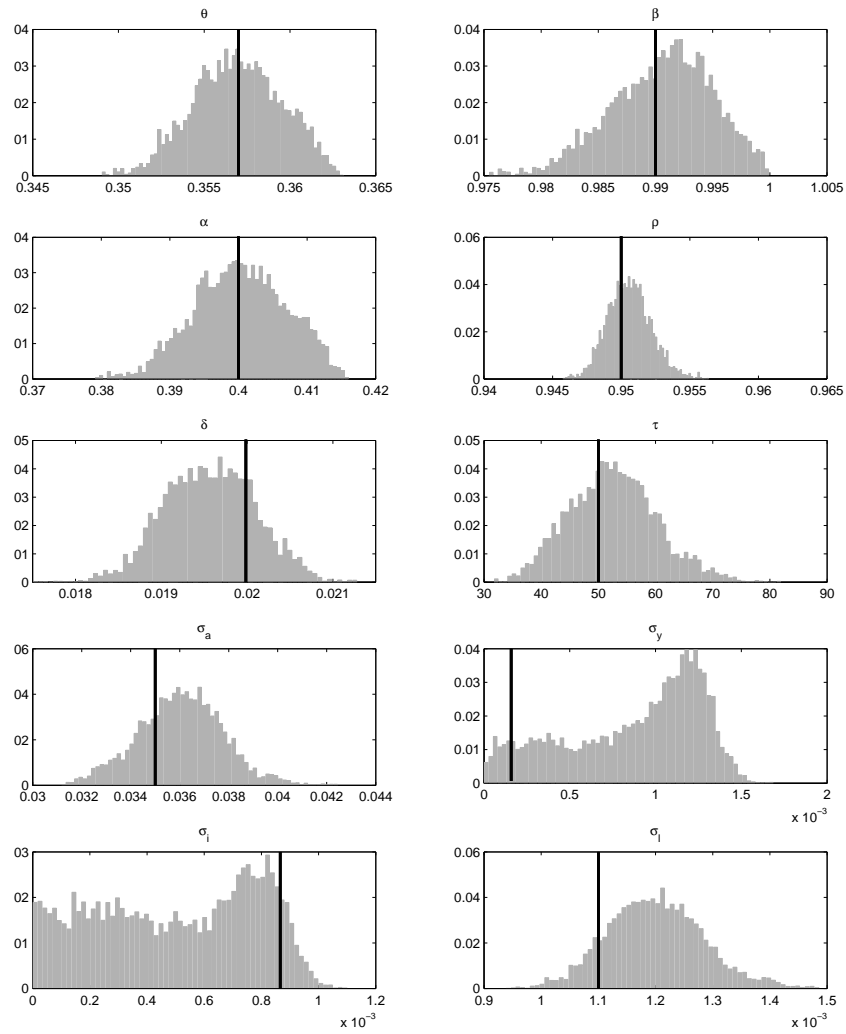


Table 15: Likelihood Slice at  $\sigma_y$

$\sigma_y$	Particle	Smolyak Kalman
$3.78_{-04}$	1,169	1,170
$5.67_{-04}$	1,172	1,171
$7.56_{-04}$	1,174	1,172
$9.45_{-04}$	1,177	1,174
$1.13_{-03}$	1,175	1,172
$1.32_{-03}$	1,177	1,171
$1.51_{-03}$	1,175	1,169
$1.70_{-03}$	1,171	1,167

Table 16: Marginal Likelihood

Small Measurement Errors			
Smolyak Kalman versus			
p	Kalman	Smolyak Sum	Smolyak Particle
0.1	104.5	-1.3	-2.1
0.5	104.0	-1.4	-2.5
0.9	104.2	-1.1	-1.9
Large Measurement Errors			
Smolyak Kalman versus			
p	Smolyak Sum	Smolyak Particle	
0.1	0.4	0.4	
0.5	0.3	0.3	
0.9	0.5	0.2	

and filter integration level 3. The results hardly change compared to the more accurate nonlinear solution and filter in the previous table. We do not report results of the estimates of a solution with level 3 and the filter with level 2 since they hardly differ. On the right hand side of the table we report the estimates with solution and filter level 2. Here we see clear biases and can conclude that the filter integration levels 2 and 3 imply similar results but a solution level of 2 is not sufficient for an accurate estimation. The log likelihood values for the accurate Smolyak Kalman filter are 1169 at the true parameters, 1186 at minimum and 1201 at maximum. The respective values for the other filters are -6546, 1141 and 1160.

Table 9 shows the other two nonlinear filters. On the left side we have the Smolyak sum filter with the solution and the rational expectation approximations at level 3, level 2 for the Smolyak Kalman filter and 5 summands to approximate the densities involved. On the right hand side we see the Smolyak particle filter with the same levels and 500 particles. Both filters deliver comparable estimates. The likelihood values are 1169 at the true parameters for the sum filter, 1185 at minimum and 1201 at maximum. The numbers for the Smolyak particle filter are 1172 at the true parameters, 1185 at minimum and 1203 at maximum.

The result of the nonlinear estimation for the data with small measurement errors is that we obtain accurate estimates for all parameters beside  $\tau$  and the measurement error standard deviations of output and investment. The estimates are essentially the same for all our nonlinear filters.

In table 10 we present the comparison between the Kalman filter estimation and an estimation with the extended Kalman filter based on a solution with level 3. The estimates are similar, especially the ones of  $\tau$ .

The next estimations process data generated with the large set of measurement errors.

The first of these estimations is in table 11 and compares the Smolyak Kalman filter with the solution and integration level 4 and the Smolyak particle filter with the solution level 3, filter level 2 and with 500 particles. Both results are again very similar and differ only in the maximum likelihood estimates of  $\tau$ . Given the high inaccuracy for this parameter, indicated by the high standard deviation of the posterior, this difference is not very surprising. Compared to the estimation with small measurement errors we observe that the means of the posterior and the maximum likelihood estimates are similar and hardly deteriorate with large measurement errors. The difference is that of course the standard deviations of the posterior substantially increase for most estimates. The likelihood values for the Smolyak Kalman filter are 851 at the true parameters, 832 and 853 at minimum and maximum, respectively. For the Smolyak particle filter these numbers are 851, 841 and 852. Again



the measurement shocks are hardly identified and the parameter  $\tau$  is badly estimated as well.

The last nonlinear estimation we report in table 12 tests whether increasing the number of summands in the Smolyak sum filter improves the estimation. Both estimations are run with solution level 3 and level 2 for the filters. On the left side we see the results of the filter with 5 summands while the right side shows the results from the filter with 20 summands. The results are very similar and the only difference is again the maximum likelihood estimates of  $\tau$  which can be attributed to the high inaccuracy of the estimator. The likelihood values for the Smolyak sum filter with 5 summands are 851, 838, 853 and 851, 837, 853 for 20 summands.

We finally investigate whether the problem we encounter with the estimation of  $\tau$  is due to the errors of the approximations or a property of the model. Table 13 shows the first approach to this question where we estimate the parameters with the Kalman filter from the data generated with the linearized model. The estimation is therefore not confounded by any approximation error. The estimates for  $\tau$  are of similar inaccuracy as for the nonlinear estimations. Moreover, we can see that the standard deviations of the output and investment measurement errors are poorly estimated as well.

A more convincing point is made in table 14. It also demonstrates that peaked likelihood slices are not necessarily informative as regards the standard deviation of a parameter estimate, since the likelihood slices seem to be properly peaked for  $\tau$  at the true parameter values in the left plot of figure 1. The left column "Smallest  $\tau$ " shows the figures with the smallest  $\tau$  of the posterior sequences. It exhibits a usual likelihood value of 1193 for this estimation obtained by the Smolyak Kalman filter based on the third level solution and integrations (1197 at true parameters, 1185 at minimum, 1200 at maximum). The particle filter with 40,000 particles delivers a similar value. If we evaluate the likelihood at the "True Values" except for  $\tau$ , where we use the smallest value of 31.88, the log likelihood collapses. If we in addition to  $\tau$  change the parameters  $\beta, \rho$  and  $\delta$  towards the ones we obtained together with the smallest  $\tau$ , the likelihood value recovers. This is done in column "Change" where the bold faced numbers show the changed parameters compared to the true values. Further changes of the parameters towards the "Smallest  $\tau$ " parametrization bring us back to the likelihood of 1193 and 1189.

We conclude that the poor estimates of  $\tau$  are not a problem of our nonlinear filters but a feature of the model and thanks to good global search properties of our parallel Metropolis-Hastings algorithm we are able to find these parameters.

A substantial change of the parameter  $\sigma_y$  from  $1.58_{-4}$  to  $1.149_{-3}$  in the

parameterization of column "Changes" does not very much influence the likelihood value. It changes from 1164 to 1169 for the Smolyak Kalman filter and from 1160 to 1169 for the particle filter. In table 15 we finally look at the likelihood slice for  $\sigma_y$  at the true parameters. The  $\sigma_y$  values are between the minimal and maximal values of the posterior sample. The variations between 1,167 and 1,174 for the Smolyak Kalman filter and 1,169 and 1,177 for the particle filter indicate a flat likelihood, in the sense that this variation is around the variation of the log likelihood we usually see during the Metropolis-Hastings sampling. We ascribe the problem of the measurement errors to the lack of proper identification of this model and the ad hoc solution by adding some measurement errors. The identification problem is also discussed by Fernández-Villaverde and Rubio-Ramírez (2006), where they obtain, however, more accurate estimates.

The last calculation is the marginal likelihood in table 16. The upper part shows the results for the estimations on the data with small measurement errors. The numbers are the differences of the marginal log likelihoods between the Smolyak Kalman filter with solution and filter level 4. A positive number is in favor of the Smolyak Kalman filter and a negative one indicates superiority of the other filters. The upper three columns show that the Smolyak Kalman filter clearly outperforms the Kalman filter (104.5, 104.0, 104.2). The Smolyak sum, calculated with solution level 3 and 5 summands with integration level 2, outperforms the Smolyak Kalman filter. Finally, we have the Smolyak particle filter with solution level 3, integration level 2 and 500 particles. Here, we see a further improvement in the marginal likelihood compared to the Smolyak Kalman filter.

The lower part of the table shows that in case of large measurement errors the performances of the filters are very similar.

## 5 Conclusion

The Smolyak operator is highly effective for the approximation and integration of high dimensional standard growth models. We have demonstrated that global approximations of the model solution beyond small models with around 6 states are possible and that models with around 20 states can be solved in reasonable time. Moreover, the solution algorithm can be parallelized.

The Smolyak operator can also be used for numerical integration which is essential in many econometric applications. The Smolyak Kalman filter is very fast and can be used if the posterior and prediction densities are reasonably approximated by a Gaussian density. If not, they can be approximated

by a Gaussian sum as in the Smolyak sum filter. Both filters are very fast and the Smolyak sum filter can even be parallelized. The particle filter is of little use in our implementation in combination with a Chebyshev approximation where the interpolation is costly and therefore results in very costly likelihood evaluations. The Smolyak particle is slower than both deterministic filters but much faster than the particle filter. It improves the plain vanilla particle filter so that measurement errors do not need to be present.

All these results indicate that deterministic filters research is an interesting road to go in nonlinear filter design. Some approaches were already developed in the seventies but were abandoned probably due to the lack of computer power and useful deterministic integration algorithms. Both are now available and a revival of these kinds of filters as an alternative to the slow Monte-Carlo based filters are worth to be considered. We therefore suggest to further investigate deterministic filtering schemes.

The parallelized Metropolis-Hastings algorithm improves the global maximization properties of the serial algorithm. We were able to discover that a small  $\tau$  can be compensated in terms of the log likelihood with a larger  $\beta$  and  $\rho$  and a smaller  $\delta$ . It is also an improvement that extensive training sequences and runs for robustness checks are not needed. The mixing feature simplifies the choice of the innovation variance. The parallel sequences provide an unbiased convergence diagnostic test and allow to implement the algorithms on parallel computers. A major improvement for the handling of the Metropolis-Hastings algorithm is the interactive feature of JBendge which allows a very comfortable estimation.

## A Linearization

The deterministic steady state  $\bar{s}, \bar{x}$  is defined by

$$\begin{aligned} 0 &= f(\bar{s}, \bar{x}, h(\bar{s}, \bar{x}, 0, \bar{s}, \bar{x})) \\ \bar{s} &= g(\bar{s}, \bar{x}, 0). \end{aligned}$$

For the one country model with states  $a$  and  $k$  and labor policy  $l$  the steady states are

$$\begin{aligned} \bar{a} &= 0 \\ \bar{k} &= -\frac{(\alpha - 1)\alpha^{\frac{1}{1-\alpha}}\beta^{\frac{1}{1-\alpha}}(\beta(\delta - 1) + 1)^{\frac{\alpha}{\alpha-1}}\theta}{-\alpha\delta\beta + \delta\beta + \alpha\theta\beta - \beta - \alpha\theta + 1} \\ \bar{l} &= \frac{(\alpha - 1)(\beta(\delta - 1) + 1)\theta}{\alpha\theta + \beta((\alpha - 1)\delta - \alpha\theta + 1) - 1} \end{aligned}$$

The linearized model is given by

$$\begin{bmatrix} I & 0 \\ -f_z h_{s'} & -f_z h_{x'} \end{bmatrix} \begin{bmatrix} ds' \\ dx' \end{bmatrix} = \begin{bmatrix} g_s & g_x \\ f_s + f_z h_s & f_x + f_z h_x \end{bmatrix} \begin{bmatrix} ds \\ dx \end{bmatrix}$$

where  $ds = s - \bar{s}$  and  $dx = x - \bar{x}$ . Primes denote the next period variables  $\{s', x'\} = \{s_{t+1}, x_{t+1}\}$ . The subscripted functions  $f$ ,  $g$  and  $h$  are Jacobians with respect to the variables in the subscript evaluated at the steady state. The left matrices on both sides of the equation can be decomposed by the generalized Schur or QZ decomposition to arrive at the following equation.

$$\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \begin{bmatrix} Z_{11}^H & Z_{21}^H \\ Z_{12}^H & Z_{22}^H \end{bmatrix} \begin{bmatrix} P \\ CP \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} Z_{11}^H & Z_{21}^H \\ Z_{12}^H & Z_{22}^H \end{bmatrix} \begin{bmatrix} I \\ C \end{bmatrix}$$

This equation can then be solved for the linear policy function defined by  $C = Z_{21} Z_{11}^{-1}$  and the state transition matrix  $P = Z_{11} S_{11}^{-1} T_{11} Z_{11}^{-1}$ , as described in Klein (2000). The linear solution of the model is finally given by

$$\begin{aligned} x_t &= \bar{x} + C(s_t - \bar{s}) \\ s_{t+1} &= \bar{s} + P(s_t - \bar{s}). \end{aligned}$$

## B Smolyak Example

The following example is given in order to clarify the difficult notation of the Smolyak algorithm.

The starting point is the index set. For a  $d = 2$  dimensional approximation at level  $q = 4$  the index set is given by all 2-dimensional vectors whose elements sum is between  $q - d + 1 = 3$  and  $q = 4$ . These vectors are given in table (17) in the two left columns captioned by "Index". The two columns to the right, captioned by "CC" show the Clenshaw-Curtis function values of the index elements. For each index vector we build a tensor product shown in column "Tensor". For example, the first CC vector  $[3, 1]$  means that we have to combine the univariate Chebyshev polynomial in the first dimension with degrees from 0 to 2 with the Chebyshev polynomials for degrees from 0 to 0. This gives the tensor products  $[[0, 0], [1, 0], [2, 0]]$  representing the three bivariate polynomials:  $[b_0(s_1)b_0(s_2), b_1(s_1)b_0(s_2), b_2(s_1)b_0(s_2)]$  where  $s_1$  and  $s_2$  are values in the first and the second dimension, respectively. The column captioned by "Smolyak" finally gives the degrees of non-repeating bivariate polynomial combinations which are shown in the last column. This repetitive pattern is captured by the binomial coefficient term  $(-1)^{q-|i|} \binom{d-1}{q-|i|}$ . Thus, the Smolyak approximation  $A_{4,2}$  is characterized by thirteen coefficients  $c_1, \dots, c_{13}$

and needs function evaluations at thirteen grid points to identify them in the 2-dimensional space. The approximating polynomial is finally given as the sum of the last column:  $x(s_1, s_2) = b_0(s_1)b_0(s_2)c_1 + b_1(s_1)b_0(s_2)c_2 + \dots + b_0(s_1)b_4(s_2)c_{13}$ .

Table 18 shows how the grid is constructed. The operation is similar to the operations on the basis functions. Apparently, the "Index" and "CC" columns are identical in both tables. The 1-dimensional grids are [0] for a one point grid, [-1, 0, 1] for three points and [-1, -1/√2, 0, 1/√2, 1] for five points.

The basis matrix can now be calculated using both tables. The Smolyak column of table 17 represents the rows of the basis matrix and each row is evaluated at the vectors of the rows of the Smolyak column of table 18. The inverted 13 × 13 basis matrix identifies thirteen coefficients in

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{13} \end{bmatrix} = \begin{bmatrix} b_0(-1)b_0(0) & \dots & b_0(-1)b_4(0) \\ b_0(0)b_0(0) & \dots & b_0(0)b_4(0) \\ \vdots & \ddots & \vdots \\ b_0(0)b_0(1/\sqrt{2}) & \dots & b_0(0)b_4(1/\sqrt{2}) \end{bmatrix}^{-1} \begin{bmatrix} f(-1, 0) \\ f(0, 0) \\ \vdots \\ f(0, 1/\sqrt{2}) \end{bmatrix}$$

Table 17: Smolyak Polynomial  $A_{q,d}$  for  $d = 2$  and  $q = 4$

Index		CC		Tensor		Smolyak		Polynomial
$i_1$	$i_2$	$m_{i_1}$	$mi_2$	$j_1$	$j_2$	$j_1$	$j_2$	
2	1	3	1	0	0	0	0	$b_0(s_1)b_0(s_2)c_1$
				1	0	1	0	$b_1(s_1)b_0(s_2)c_2$
				2	0	2	0	$b_2(s_1)b_0(s_2)c_3$
1	2	1	3	0	0			
				0	1	0	1	$b_0(s_1)b_1(s_2)c_4$
				0	2	0	2	$b_0(s_1)b_2(s_2)c_5$
3	1	5	1	0	0			
				1	0			
				2	0			
				3	0	3	0	$b_3(s_1)b_0(s_2)c_6$
				4	0	4	0	$b_4(s_1)b_0(s_2)c_7$
2	2	3	3	0	0			
				0	1			
				0	2			
				1	0			
				1	1	1	1	$b_1(s_1)b_1(s_2)c_8$
				1	2	1	2	$b_1(s_1)b_2(s_2)c_9$
				2	0			
				2	1	2	1	$b_2(s_1)b_1(s_2)c_{10}$
				2	2	2	2	$b_2(s_1)b_2(s_2)c_{11}$
1	3	5	1	0	0			
				0	1			
				0	2			
				0	3	0	3	$b_0(s_1)b_3(s_2)c_{12}$
				0	4	0	4	$b_0(s_1)b_4(s_2)c_{13}$

Table 18: Smolyak Grid  $A_{q,d}$  for  $d = 2$  and  $q = 4$

Index		CC		Tensor		Smolyak	
$i_1$	$i_2$	$m_{i_1}$	$m_{i_2}$	$j_1$	$j_2$	$j_1$	$j_2$
2	1	3	1	-1	0	-1	0
				0	0	0	0
				1	0	1	0
1	2	1	3	0	-1	0	-1
				0	0	0	0
				0	1	0	1
3	1	5	1	-1	0	$-1/\sqrt{2}$	0
				$-1/\sqrt{2}$	0		
				0	0	$1/\sqrt{2}$	0
				$1/\sqrt{2}$	0		
1	0						
2	2	3	3	-1	-1	-1	-1
				-1	0		
				-1	1	-1	1
				0	-1		
				0	0		
				0	1		
				1	-1	1	-1
				1	0		
1	1	1	1				
1	3	5	1	0	-1	0	$-1/\sqrt{2}$
				0	$-1/\sqrt{2}$		
				0	0	0	$1/\sqrt{2}$
				0	$1/\sqrt{2}$		
				0	1		

## References

- ALSBACH, D. L., AND H. W. SORENSON (1972): “Nonlinear Bayesian Estimation Using Gaussian Sum Approximations,” *IEEE Transactions on Automatic Control*, Ac-17(4).
- AMISANO, G., AND O. TRISTANI (2007): “Euro Area Inflation Persistence in an Estimated Nonlinear DSGE Model,” Discussion Paper 6373, C.E.P.R.
- ANDERSON, B. D. O., AND J. B. MOORE (1979): *Optimal Filtering*. Dover Publications.
- ANDERSON, G., A. LEVIN, AND E. SWANSON (2005): “Higher-Order Perturbation Solutions to Dynamic, Discrete-Time Rational Expectations Models,” .
- ARULAMPALAM, S., S. MASKELL, N. GORDON, AND T. CLAPP (2002): “A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, 50(2), 174–188.
- ARUOBA, B. S., J. FERNÁNDEZ-VILLAYERDE, AND J. F. RUBIO-RAMÍREZ (2006): “Comparing Solution Methods for Dynamic Equilibrium Economies,” *Journal of Economic Dynamics and Control*, 30, 2477–2508.
- BERGER, J., AND R. WOLPERT (1988): *The Likelihood Principle*. Hayward, California, 2<sup>nd</sup> edn., Institute of Mathematical Statistics.
- BERZUINI, C., N. G. BEST, W. R. GILKS, AND C. LARIZZA (1997): “Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods,” *Journal of the American Statistical Association*, 92(440), 1403–1412.
- BROOKS, S. P., AND A. GELMAN (1998): “General methods for monitoring convergence of iterative simulations,” *Journal of Computational & Graphical Statistics*, 7(4), 434–455.
- BUNGARTZ, H.-J., AND M. GRIEBEL (2004): “Sparse grids,” *Acta Numerica*, 13, 1–123.
- CHIB, S., AND E. GREENBERG (1995): “Understanding the Metropolis-Hastings Algorithm,” *The American Statistician*, 49(4), 327–335.
- DOUCET, A., N. DE FREITAS, AND N. GORDON (eds.) (2001): *Sequential Monte Carlo Methods in Practice*. Springer, New York.



- FERNÁNDEZ-VILLAVERDE, J., AND J. F. RUBIO-RAMÍREZ (2004): “Comparing Dynamic Equilibrium Models to Data: A Bayesian Approach,” *Journal of Econometrics*, 123, 153–187.
- (2005): “Estimating Dynamic Equilibrium Economies: Linear versus Nonlinear Likelihood,” *Journal of Applied Econometrics*, 20, 891–910.
- (2006): “Estimating Macroeconomic Models: A Likelihood Approach,” Discussion Paper 5513, CEPR.
- GASPAR, J., AND K. L. JUDD (2005): “Solving Large-Scale Rational-Expectations Models,” *Macroeconomic Dynamics*, 1(01), 45–75.
- GELFAND, A., AND D. DEY (1994): “Bayesian Model Choice: Asymptotics and Exact Calculations,” *Journal of the Royal Statistical Society, Series B*, 56, 501–514.
- GELMAN, A., AND D. B. RUBIN (1992): “Inference from Iterative Simulation Using Multiple Sequences,” *Statistical Science*, 7(4), 457–472.
- GENZ, A., AND C. KEISTER (1996): “Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weights,” *Journal of Computational and Applied Mathematics*, 71, 299–309.
- GEWEKE, J. (1999): “Using Simulation Methods for Bayesian Econometric Models: Inference, Development, and Communication,” *Econometric Reviews*, 18, 1–126.
- (2005): *Contemporary Bayesian Econometrics and Statistics*. John Wiley & Sons, New York.
- HEISS, F., AND V. WINSCHERL (2008): “Likelihood Approximation by Numerical Integration on Sparse Grids,” .
- IONIDES, E., C. BRETÓ, AND A. KING (2006): “Inference for nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences of the United States of America, PNAS*, 103(49), 18438–18443.
- JUDD, K. (1992): “Projection methods in economics,” *Journal of Economic Theory*, 58, 410–452.
- JUDD, K. L. (1998): *Numerical Methods in Economics*. MIT Press, Cambridge, Mass.

- JUDD, K. L., AND S.-M. GUU (1997): “Asymptotic methods for aggregate growth models,” *Journal of Economic Dynamics and Control*, 21, 1025–1042.
- JUDD, K. L., AND H.-H. JIN (2002): “Perturbation Methods for General Dynamic Stochastic Models,” Discussion paper, Hoover Institution, Stanford.
- JUILLARD, M. (1996): “Dynare: A program for the resolution and simulation of dynamic models with forward variables through the use of a relaxation algorithm,” Discussion paper, CEPREMAP.
- JULIER, S. J., AND J. K. UHLMANN (1997): “A new extension of the Kalman filter to nonlinear systems,” in *The proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls, Orlando, Florida*.
- KALMAN, R. E. (1960): “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D), 35–45.
- KIM, J., AND S. KIM (2003): “Spurious welfare reversals in international business cycle models,” *Journal of International Economics*, 60(2), 471–500.
- KLEIN, P. (2000): “Using the generalized Schur form to solve a multivariate linear rational expectations model,” *Journal of Economic Dynamics and Control*, 24(10), 1405–1423.
- KOTECHA, J., AND P. DJURIĆ (2003): “Gaussian Sum Particle Filter,” *IEEE Transactions on Signal Processing*, 51, 2602–2612.
- KRÜGER, D., AND F. KÜBLER (2004): “Computing equilibrium in OLG models with stochastic production,” *Journal of Economic Dynamics and Control*, 28, 1411–1436.
- LANDON-LANE, J. S. (1998): “Bayesian Comparison of Dynamic Macroeconomic Models,” Ph.D. thesis, University of Minnesota.
- MIRANDA, M. J., AND P. L. FACKLER (2002): *Applied Computational Economics and Finance*. MIT Press, Cambridge MA.
- SCHMITT-GROHÉ, S., AND M. URIBE (2004): “Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function,” *Journal of Economic Dynamics and Control*, 28, 755–775.

- SMOLYAK, S. (1963): “Quadrature and interpolation formulas for tensor products of certain classes of functions,” *Soviet Math.Dokl*, 4, 240–243.
- STORN, R., AND K. PRICE (1997): “Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimisation*, pp. 341–359.
- TER BRAAK, C. J. (2006): “A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces,” *Statistics and Computing*, 16(3), 239–249.
- VAN DER MERWE, R., A. DOUCET, N. DE FREITAS, AND E. WAN (2000): “The Unscented Particle Filter,” *Advances in Neural Information Processing Systems*, 13, 584–590.

## SFB 649 Discussion Paper Series 2008

For a complete list of Discussion Papers published by the SFB 649, please visit <http://sfb649.wiwi.hu-berlin.de>.

- 001 "Testing Monotonicity of Pricing Kernels" by Yuri Golubev, Wolfgang Härdle and Roman Timonfeev, January 2008.
- 002 "Adaptive pointwise estimation in time-inhomogeneous time-series models" by Pavel Cizek, Wolfgang Härdle and Vladimir Spokoiny, January 2008.
- 003 "The Bayesian Additive Classification Tree Applied to Credit Risk Modelling" by Junni L. Zhang and Wolfgang Härdle, January 2008.
- 004 "Independent Component Analysis Via Copula Techniques" by Ray-Bing Chen, Meihui Guo, Wolfgang Härdle and Shih-Feng Huang, January 2008.
- 005 "The Default Risk of Firms Examined with Smooth Support Vector Machines" by Wolfgang Härdle, Yuh-Jye Lee, Dorothea Schäfer and Yi-Ren Yeh, January 2008.
- 006 "Value-at-Risk and Expected Shortfall when there is long range dependence" by Wolfgang Härdle and Julius Mungo, January 2008.
- 007 "A Consistent Nonparametric Test for Causality in Quantile" by Kiho Jeong and Wolfgang Härdle, January 2008.
- 008 "Do Legal Standards Affect Ethical Concerns of Consumers?" by Dirk Engelmann and Dorothea Kübler, January 2008.
- 009 "Recursive Portfolio Selection with Decision Trees" by Anton Andriyashin, Wolfgang Härdle and Roman Timofeev, January 2008.
- 010 "Do Public Banks have a Competitive Advantage?" by Astrid Matthey, January 2008.
- 011 "Don't aim too high: the potential costs of high aspirations" by Astrid Matthey and Nadja Dwenger, January 2008.
- 012 "Visualizing exploratory factor analysis models" by Sigbert Klinke and Cornelia Wagner, January 2008.
- 013 "House Prices and Replacement Cost: A Micro-Level Analysis" by Rainer Schulz and Axel Werwatz, January 2008.
- 014 "Support Vector Regression Based GARCH Model with Application to Forecasting Volatility of Financial Returns" by Shiyi Chen, Kiho Jeong and Wolfgang Härdle, January 2008.
- 015 "Structural Constant Conditional Correlation" by Enzo Weber, January 2008.
- 016 "Estimating Investment Equations in Imperfect Capital Markets" by Silke Hüttel, Oliver Mußhoff, Martin Odening and Nataliya Zinych, January 2008.
- 017 "Adaptive Forecasting of the EURIBOR Swap Term Structure" by Oliver Blaskowitz and Helmut Herwatz, January 2008.
- 018 "Solving, Estimating and Selecting Nonlinear Dynamic Models without the Curse of Dimensionality" by Viktor Winschel and Markus Krätzig, February 2008.

**SFB 649, Spandauer Straße 1, D-10178 Berlin**  
**<http://sfb649.wiwi.hu-berlin.de>**

This research was supported by the Deutsche  
Forschungsgemeinschaft through the SFB 649 "Economic Risk".

