Econometrica Supplementary Material

SUPPLEMENT TO "SOLVING, ESTIMATING, AND SELECTING NONLINEAR DYNAMIC MODELS WITHOUT THE CURSE OF DIMENSIONALITY"
(*Econometrica*, Vol. 78, No. 2, March 2010, 803–821)

BY VIKTOR WINSCHEL AND MARKUS KRÄTZIG[1]

This supplement contains details on the derivation of the dynamic first-order optimality conditions, the linearization of the general model, the new filters, the parallelized Metropolis–Hastings algorithm, an example of a two-dimensional Smolyak polynomial approximation, and detailed simulation results.

KEYWORDS: Dynamic stochastic general equilibrium (DSGE) models, Bayesian time series econometrics, curse of dimensionality.

S1. INTRODUCTION

THIS PAPER CONTAINS the details of the model linearization and solution, the nonlinear approximation and integration scheme, the details on the Smolyak operator, the nonlinear solution iteration scheme, details on our proposed nonlinear state space filter, the Metropolis–Hastings algorithm and its parallel extension, the convergence diagnostics for the Metropolis–Hastings algorithms, the calculation of the marginal likelihood model selection criterion, and details on the solver, likelihood evaluation, and estimation performance.

S2. ECONOMICS

S2.1. *Model*

The social planner allocates by solving the dynamic optimization

$$\max_{\{\{c_{n,t},l_{n,t},i_{n,t}\}_{n=1}^{N}\}_{t=0}^{\infty}} U = \mathrm{E}_0 \sum_{t=0}^{\infty}\sum_{n=1}^{N} \beta^t U_{n,t}$$

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

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

[1] 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 participants of the Institute on Computational Economics at the University of Chicago and 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.

DOI: 10.3982/ECTA6297

with discount factor β, elasticity of intertemporal substitutions τ_n, and consumption and leisure substitution rates θ_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 $\Sigma_n y_{n,t}$ 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 $k_{n,t}$, labor $l_{n,t}$, and technical substitution rates α_n. The capital and productivity transitions are

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

$$a_{n,t+1} = \rho_n a_{n,t} + e_{n,t+1}, \tag{S.2}$$

where δ_n are the depreciation rates and ρ_n are 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 (S.1), we include capital adjustment costs parameterized by κ_n.

We have implemented the algorithms for the general model class:

$$\begin{aligned} 0 &= f(s_t, x_t, z_t; \theta), \\ z_t &= \mathrm{E}_e h(s_t, x_t, e_{t+1}, s_{t+1}, x_{t+1}; \theta), \\ s_{t+1} &= g(s_t, x_t, e_{t+1}; \theta). \end{aligned}$$

All functions depend on the structural parameters in θ. The model is formulated in terms of the first-order equilibrium conditions $f : \mathbb{R}^{d_s+d_x+d_z} \to \mathbb{R}^{d_x}$, expected functions $h : \mathbb{R}^{d_s+d_x+d_e+d_s+d_x} \to \mathbb{R}^{d_z}$, and state transitions $g : \mathbb{R}^{d_s+d_x+d_e} \to \mathbb{R}^{d_s}$. 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 Σ_e from now on.

Our solution approach is to solve for the policy functions $x^* : \mathbb{R}^{d_s} \to \mathbb{R}^{d_x}$ that map states into policies. The algorithm is a function iteration scheme where we repeatedly solve the first-order conditions $f(s, x^{(k+1)}, \mathrm{E}_e h(\dots, x^{(k)}, \dots)) = 0$ for the next $k + 1$ iteration of the policy $x^{(k+1)}$ for given expected variables $z = \mathrm{E}_e h(\dots, x^{(k)}, \dots)$ 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 $f(s, x, \mathrm{E}_e h(\ldots, x, \ldots)) = 0$, into several independent smaller ones, for each grid point we solve one small system for all policies.

Since the numerical integration can be thought, of 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

$$\mathrm{E}_e h(\ldots, e, \ldots) = \int h(\ldots, e, \ldots) p(e)\, de \approx \sum_j w_j h(\ldots, e_j, \ldots),$$

where the continuous random variable e and its density $p(e)$ are 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

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

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.

S2.2. *Optimality*

The Bellman equation for the allocation problem is

$$\begin{aligned}
V_t = \max_{\{c_{n,t}, l_{n,t}, i_{n,t}, k_{n,t+1}\}_{n=1}^N} &\left(\sum_{n=1}^N U_{n,t} + \beta \mathrm{E}_t V_{t+1} + \lambda_B \sum_{n=1}^N (y_{n,t} - c_{n,t} - i_{n,t}) \right. \\
&\left. + \sum_{n=1}^N \lambda_n (k_{n,t+1} - (1 - \delta_n) k_{n,t} - i_{n,t} + 0.5 \kappa_n i_{n,t}^2) \right),
\end{aligned}$$

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

$$\frac{\partial V_t}{\partial c_{n,t}} = \frac{\partial U_{n,t}}{\partial c_{n,t}} - \lambda_B = 0, \tag{S.3}$$

$$\frac{\partial V_t}{\partial l_{n,t}} = \frac{\partial U_{n,t}}{\partial l_{n,t}} + \lambda_B \frac{\partial y_{n,t}}{\partial l_{n,t}} = 0, \tag{S.4}$$

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

$$\frac{\partial V_t}{\partial k_{n,t+1}} = \beta \mathrm{E}_t \frac{\partial V_{t+1}}{\partial k_{n,t+1}} + \lambda_n = 0, \tag{S.6}$$

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

The last optimality condition is the budget constraint

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

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

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

since the derivatives of the policy variables with respect to $k_{n,t}$ are zero by optimality. The Lagrange multipliers λ_n in equation (S.9) can be substituted by equations (S.5) and the multiplier λ_B can be substituted by equation (S.3), and consequently we arrive at

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

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

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

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

$$\frac{\partial U_{n,t}}{\partial l_{n,t}} + \frac{\partial U_{n,t}}{\partial c_{n,t}} \frac{\partial y_{n,t}}{\partial l_{n,t}} = 0. \tag{S.11}$$

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

$$\frac{\partial U_{1,t}}{\partial c_{1,t}} = \frac{\partial U_{n,t}}{\partial c_{n,t}}, \tag{S.12}$$

which enforce equal marginal utilities across all the countries.

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

The mapping into the general model class is the following: The $2N-1$ equations (S.10) and (S.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 (S.10). The N capital transitions (S.7) and the N productivity transitions (S.2) form the state transition functions g of the general model.

S2.3. *Solution*

S2.3.1. *Approximation*

The first step toward 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.

S2.3.2. *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

$$\bar{a} = 0,$$

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

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 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

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

S2.3.3. *Nonlinear Solution*

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 $b_{j+1}(s) = 2sb_j(s) - b_{j-1}(s)$ 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 $s^i = \{s_1^i, s_2^i, \ldots, s_{m_i}^i\}$. In a one-dimensional approximation, for example, with $m_i = 3$ coefficients, we have to solve the linear equations

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

This 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 $s^1_1 = 0$ and $s^i_j = -\cos(\pi(j-1)/(m_i-1))$ 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.

S2.3.4. *Tensor Operator*

If the functions to be approximated or integrated depend on several variables, we need a rule for 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 $x : [0, 1] \to \mathbb{R}$ is

$$U^i(x) = \sum_{j=1}^{m_i} a^i_j x(s^i_j),$$

where $i \in \mathbb{N}$ is the approximation level and $s^i_j \in [-1, 1]$ are the grid points. In the case of a function approximation, $a^i_j \in \mathbb{C}$ for $j = 1, \dots, J$ are functions of s; in the case of numerical integration, a^i_j 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

$$(U^{i_1} \otimes \cdots \otimes U^{i_d})(x) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} \bigl(a^{i_1}_{j_1} \otimes \cdots \otimes a^{i_d}_{j_d}\bigr) x\bigl(s^{i_1}_{j_1}, \dots, s^{i_d}_{j_d}\bigr).$$

To construct this approximation, we need $\prod_{j=1}^{d} m_{i_j}$ function evaluations $x(s^{i_1}_{j_1}, \dots, s^{i_d}_{j_d})$. 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 is given by $B(s^i) = b(s^{i_1}) \otimes \cdots \otimes b(s^{i_d})$. The grid $s^i = s^{i_1} \times \cdots \times s^{i_d}$ 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 of 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.

S2.3.5. *Smolyak Operator*

The usual formulation of the Smolyak operator is

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

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 Smolyak operator constructs the multivariate grid as a combination of some lower level Cartesian products $\times$ of the univariate grids $s^{i_j} = \{s_1^{i_j}, \dots, s_{m_{i_j}}^{i_j}\}$,

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

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

For alternative presentations of the operator, see Kübler and Krüger (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.

S3. SMOLYAK EXAMPLE

The following example is given 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 two-dimensional vectors whose elements sum is between $q - d + 1 = 3$ and $q = 4$. These vectors are given in Table S.I in the two left columns captioned “Index.” The two columns to the right, captioned “CC,” show the Clenshaw–Curtis function values of the index elements. For each index vector, we build a tensor product shown in the 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]] that represent 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 dimensions, respectively. The column captioned

TABLE S.I

SMOLYAK POLYNOMIAL $A_{q,d}$ 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	Polynomial
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	$b_0(s_1)b_4(s_2)c_{13}$

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

Table S.II 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 one-dimensional grids are [0] for a one point grid, $[-1, 0, 1]$ for a three points grid, and $[-1, -1/\sqrt{2}, 0, 1/\sqrt{2}, 1]$ for a five point grid.

The basis matrix can now be calculated using both tables. The Smolyak column of Table S.I represents the rows of the basis matrix and each row is eval-

TABLE S.II

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	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		

uated at the vectors of the rows of the Smolyak column of Table S.II. The inverted 13 × 13 basis matrix identifies 13 coefficients in

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

S3.1. *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 (S.13). The policy values are given only implicitly and a root finder or a function iteration has to improve a start value. Table S.III 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 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 pointwise. 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

TABLE S.III

FUNCTION ITERATION AND ROOT FINDING

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

Jacobians are available for the small systems. In JBendge, the Jacobians are automatically derived by a 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 cannot solve for them (the Jacobians become singular).

However, 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.

S3.2. *Error*

An approximation procedure has to be accompanied by error estimates so as 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,

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

$$\mathbf{s}' = g(s, x^*(s), e').$$

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 1 per 1000 dollars.

S4. ECONOMETRICS

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

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

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

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

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

S4.1. *Filtering*

The policy functions $x^*(s)$ that solve the model 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

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

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 ε_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 θ is suppressed.

The filtering approach evaluates the likelihood of this model by the prediction and filtering steps. 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

$$\begin{aligned} p(s_t|y_{1:t-1}) &= \int p(s_t, s_{t-1}|y_{1:t-1})\,ds_{t-1} \\ &= \int p(s_t|s_{t-1})p(s_{t-1}|y_{1:t-1})\,ds_{t-1}, \end{aligned} \tag{S.15}$$

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 allow us to update the prediction density

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

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

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

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

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

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 simply implementable 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 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 the 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).

S4.1.1. *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 to solve, but not yet 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 μ, and covariance Σ. Assuming that the previous posterior density is Gaussian,

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

the prior density

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

is characterized by its first two moments,

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

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

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

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

where the expected value is

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

and the covariance is

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

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

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

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

$$
\begin{aligned}
\text{(S.24)} \qquad & p(s_t|y_{1:t}) = \mathcal{N}\big(s_t; s_{t|t}, \Sigma^s_{t|t}\big), \\
& K_t = \Sigma^{sy}_{t|t-1}\big(\Sigma^y_{t|t-1}\big)^{-1}, \\
& s_{t|t} = s_{t|t-1} + K_t\big(y_t - y_{t|t-1}\big), \\
& \Sigma^s_{t|t} = \Sigma^s_{t|t-1} - K_t \Sigma^y_{t|t-1} K_t^T
\end{aligned}
$$

according to a usual Kalman update. The numerical problem to be solved for this filter is evaluation of the integrals for the expected value and covariance of the state prediction in equations (S.19) and (S.20), the expected value of the observables in equation (S.21), the innovation covariance in equation (S.22), and the covariance between the states and observations in equation (S.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 equations (S.19) and (S.20), respectively, can be approximated by the nodes and weights for the joint density of the states and shocks as the weighted sum

$$
\begin{aligned}
s_{t|t-1} &= \sum_{i=1}^N w^{(i)} g\big(s^{(i)}_{t-1|t-1}, e^{(i)}_t\big), \\
\Sigma^s_{t|t-1} &= \sum_{i=1}^N w^{(i)} g\big(s^{(i)}_{t-1|t-1}, e^{(i)}_t\big) g\big(s^{(i)}_{t-1|t-1}, e^{(i)}_t\big)^T - s_{t|t-1} s^T_{t|t-1}.
\end{aligned}
$$

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

$$
\begin{aligned}
y_{t|t-1} &= \sum_{i=1}^N w^{(i)} m\big(s^{(i)}_{t|t-1}\big), \\
\Sigma^y_{t|t-1} &= \sum_{i=1}^N w^{(i)} m\big(s^{(i)}_{t|t-1}\big) m\big(s^{(i)}_{t|t-1}\big)^T + \Sigma_\varepsilon - y_{t|t-1} y^T_{t|t-1}.
\end{aligned}
$$

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

$$
\Sigma^{sy}_{t|t-1} = \sum_{i=1}^N w^{(i)} s^{(i)}_{t|t-1} m\big(s^{(i)}_{t|t-1}\big)^T - s_{t|t-1} y^T_{t|t-1}.
$$

S4.1.2. *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

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

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 δ 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 (S.25) as

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

For a recursive algorithm we need to factorize the importance density

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

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 text, 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

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

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 (S.27) and (S.28) into (S.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

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

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:

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

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 then reproduce those with the highest weights and drop the others. If we resample in each iteration, the weight update equations (S.29) and (S.30) become

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

and

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

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

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

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 (S.31) and (S.32) to approximate

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

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 (S.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 of calculating the weights, the filter cannot handle state space models without measurement errors. Resampling, in turn, introduces a bias into the particle filter; see, for example, 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

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

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

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

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.

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

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

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 $y_t - y_t^{(i)} \sim \mathcal{N}(\varepsilon_t; 0, \Sigma_\varepsilon)$.

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

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

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

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

The period likelihood is then given by the mean of the weights in Step 2:

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

These likelihoods are obtained recursively for each period and at the end of the complete sample, we arrive at the sample likelihood in equation (S.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 state's density with many implied particle weights close to zero. This is inefficient and the next filter uses a better proposal density.

S4.1.3. *Smolyak Particle Filter*

Our idea to improve the particle filter is similar to that 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 is more accurate and more general than in the unscented filter. The proposal density is now

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

The weights in equation (S.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 that results from the approximative proposal, we can calculate the weights as

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

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

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

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

$$p\big(s_t|s_{t-1}^{(i)}, y_{1:t-1}\big) = \int p\big(s_t|s_{t-1}^{(i)}\big)p\big(s_{t-1}^{(i)}|y_{1:t-1}\big)\, de_t \approx \mathcal{N}\big(s_t; s_{t|t-1}^{(i)}, \Sigma_{t|t-1}^{s(i)}\big),$$

$$s_{t|t-1}^{(i)} = \sum_{j=1}^{J} w^{(j)} g\big(s_{t-1}^{(i)}, e^{(j)}\big),$$

$$\Sigma_{t|t-1}^{s(i)} = \sum_{j=1}^{J} w^{(j)} g\big(s_{t-1}^{(i)}, e^{(j)}\big) g\big(s_{t-1}^{(i)}, e^{(j)}\big)^T - s_{t|t-1} s_{t|t-1}^T.$$

The moments of the observables

$$p\big(y_t|s_{t-1}^{(i)}, y_{1:t-1}\big) = \int p(y_t|s_t)p\big(s_t|s_{t-1}^{(i)}, y_{1:t-1}\big)\,de_t \approx \mathcal{N}\big(y_t; y_{t|t-1}^{(i)}, \Sigma_{t|t-1}^{y(i)}\big), \tag{S.33}$$

$$y_{t|t-1}^{(i)} = \sum_{j=1}^{J} w^{(j)} m\big(g\big(s_{t-1}^{(i)}, e^{(j)}\big)\big),$$

$$\Sigma_{t|t-1}^{y(i)} = \sum_{j=1}^{J} w^{(j)} m\big(g\big(s_{t-1}^{(i)}, e^{(j)}\big)\big) m\big(g\big(s_{t-1}^{(i)}, e^{(j)}\big)\big)^T + \Sigma_\varepsilon - y_{t|t-1} y_{t|t-1}^T,$$

$$\Sigma_{t|t-1}^{sy(i)} = \sum_{j=1}^{J} w^{(j)} g\big(s_{t-1}^{(i)}, e^{(j)}\big) m\big(g\big(s_{t-1}^{(i)}, e^{(j)}\big)\big)^T - s_{t|t-1} y_{t|t-1}^T$$

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

$$p\big(s_t|s_{t-1}^{(i)}, y_t\big) \approx \mathcal{N}\big(s_t; s_{t|t}^{(i)}, \Sigma_{t|t}^{s(i)}\big),$$

$$K_t = \Sigma_{t|t-1}^{sy}\big(\Sigma_{t|t-1}^{y}\big)^{-1},$$

$$s_{t|t}^{(i)} = s_{t|t-1}^{(i)} + K_t\big(y_t - y_{t|t-1}^{(i)}\big),$$

$$\Sigma_{t|t}^{s(i)} = \Sigma_{t|t-1}^{s(i)} - K_t \Sigma_{t|t-1}^{y(i)} K_t^T$$

from which we draw the next state particle:

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

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

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

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 (S.33):

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

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, and the period likelihood can be approximated by the mean of the unnormalized weights

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

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.

S4.1.4. *Smolyak Sum Filter*

The last filter we present is based on a sum of Gaussian densities for the approximation of 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 used importance sampling to approximate the involved integrals as they do in their Gaussian particle filter. Instead we propose to use Smolyak Gaussian quadrature again.

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

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

This is simply a parallel evaluation of several Smolyak Kalman filter steps so as to calculate the mean $s_{t|t-1}^i$ and the variance $\Sigma_{s;t|t-1}^i$ according to equations (S.19) and (S.20), respectively. Anderson and Moore (1979) presented 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 nonadditive state shocks, this assumption deserves some further elaboration in future research.

The filtering steps involved are again the same as those toward the density in equation (S.24):

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

The weights are updated according to

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

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 that allows us to parallelize this filter.

S4.2. *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 the Bayes formula and the object of interest is the posterior of unobservables θ:

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

An analytical expression is not available for either the likelihood or the posterior, but a random number generator that draws from this density can provide a histogram as an approximation.

S4.2.1. *Metropolis–Hastings*

The Metropolis–Hastings algorithm allows us 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 S.IV. 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.

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 $\Sigma_\varepsilon = \mathrm{Cov}(\theta)$. It has to be scaled so that the acceptance ratio is around 0.3. For a normal target density, this is achieved by $\gamma^{\mathrm{RW}} = 2.38/\sqrt{D}$, where D is the number of estimated parameters. Of course, the target density and its covariance $\mathrm{Cov}(\theta)$ are not known because they are the objects of interest.

TABLE S.IV

METROPOLIS–HASTINGS ALGORITHM

```
1. Choose start value $\theta_1$ and $\Sigma_\varepsilon$ for an acceptance ratio of $\approx 30\%$
2. For $n = 2$, while $n - J < N$, $n = n + 1$
   (a) Candidate: $\hat{\theta}_n^* = \hat{\theta}_{n-1} + \varepsilon$, where $\mathcal{N}(\varepsilon; 0, \Sigma_\varepsilon)$
   (b) Acceptance:
       $\hat{\theta}_n = \begin{cases} \hat{\theta}_n^* & \text{if } U(0,1) \le \dfrac{p(y_{1:t}|\hat{\theta}_n^*)p(\hat{\theta}_n^*)}{p(y_{1:t}|\hat{\theta}_{n-1})p(\hat{\theta}_{n-1})} \\ \hat{\theta}_{n-1} & \text{otherwise} \end{cases}$
   (c) Decide on $J$ by diagnostic tests
3. Disregard burn-in draws $\hat{\theta}_{1:J}$
```

The chosen variances in Σ_ε 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 aimed at deciding on convergence. The decision is about the number J that determines how many burn-in draws $\hat{\theta}_{1:J}$ are to be ignored. Formal convergence tests are an important part of the analysis together with eyeballing, and after some estimations, one acquires a visual feeling for convergence.

S4.2.2. *Convergence Test*

A convergence test can be based on one subdivided sequence or on several parallel sequences. 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 three-dimensional tensor $\hat{\theta}$ of size $N \times D \times M$ with elements $\hat{\theta}^d_{n,m}$, where D is the number of estimated parameters, N is the number of draws, and M is the number of sequences. $\hat{\theta}_{n,m}$ is a $1 \times D$ vector that represents the nth draw in the mth sequence and $\hat{\theta}_{:,m}$ is a $N \times D$ matrix that 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

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

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

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

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

$$R = \frac{N-1}{N} + \frac{M+1}{M}\lambda_{\max}, \quad \text{where} \quad \lambda_{\max} = \max_{a} \frac{a'Va}{a'Wa}.$$

$\lambda_{\max}$ 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 stable, 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.

S4.2.3. *Parallel Extension*

The variances of the random walk shocks Σ_ε have to be tuned for an acceptance ratio of around 0.3. It is quite demanding to find good values for all parameters simultaneously; 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. There by 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 S.V.

TABLE S.V

PARALLEL METROPOLIS–HASTINGS ALGORITHM

```
1. Choose start values $\hat{\theta}_{1,m}$ for all $m = 1, \ldots, M$ and $b$, $\gamma^{\mathrm{GE}}$ for an acceptance ratio of $\approx 30\%$
2. For $n = 1$, while $n - J < N$, $n = n + 1$
   (a) Repeat for $m = 1, \ldots, M$
       i. Draw $m_1$ and $m_2$ such that $m_1 \neq m_2 \neq m$
       ii. Candidate: $\hat{\theta}^*_m = \hat{\theta}_{m,n} + \gamma^{\mathrm{GE}}(\hat{\theta}_{m_1,n} - \hat{\theta}_{m_2,n}) + \varepsilon, \mathcal{N}(\varepsilon; 0, bI)$
       iii. Acceptance:
            $\hat{\theta}_{n,m} = \begin{cases} \hat{\theta}^*_m & \text{if } U(0,1) \leq \dfrac{p(y_{1:t}|\hat{\theta}^*_m)p(\hat{\theta}^*_m)}{p(y_{1:t}|\hat{\theta}_m)p(\hat{\theta}_m)} \\ \hat{\theta}_{n,m} & \text{otherwise} \end{cases}$
   (b) Decide on $J$ by diagnostic tests
3. Disregard burn-in draws $\hat{\theta}_{1:J,1:M}$
```

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 γ^{GE}. In the estimated models for this paper, the optimal parameters were almost identical for the linear and nonlinear estimations. 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 $\hat{\theta}^d_{n,m}$ 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 γ^{GE} and the shock variance b determine the relative weight of mixing and random walk innovations.

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

The intuition behind this procedure is that the variance of the difference between two randomly drawn parameters is optimal given that the 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 a similar size. ter Braak (2006) derived 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 the CPUs need to exchange is the matrix $\hat{\theta}_{n,1:M}$ of accepted draws. Its size is only $D \times M$, so 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 to almost proportional to the number of CPUs.

S4.3. *Marginal Likelihood*

One of the challenges in model selection 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.

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

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

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 light of the data.

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

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

According to the last equation, all we have to do is calculate a weighted mean of the Metropolis–Hastings sequence. Geweke (1999) proposed the following procedure: Calculate the mean and covariance of the parameter draws for each model M_i:

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

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

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

With density $h(\cdot)$,

$$h(\theta) = p^{-1} (2\pi)^{-D/2} |\Sigma_{M_i}|^{-1/2} \\ \times \exp\biggl(-\frac{1}{2}(\theta - \bar{\theta}_{M_i})' \Sigma_{M_i}^{-1} (\theta - \bar{\theta}_{M_i})\biggr) I_{\Theta_{M_i}}(\theta),$$

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

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

S5. RESULTS

In solution Section S5.1, we compare the performance of the Smolyak and the tensor operator within the solution algorithm. In estimation Section S5.2.1, we compare the likelihood values obtained by the Smolyak Kalman, Smolyak sum, Smolyak particle, and particle filter. Then in Section S5.2.2, 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 hyperthreading. 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 $\alpha = 0.4$, $\beta = 0.99$, $\delta = 0.02$, $\rho = 0.95$, $\theta = 0.357$, $\tau = 2.0$, and $\sigma_a = 0.007$. 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 Section S2.3.2. 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 τ or σ_a. The missing investment costs also allow us to compare our estimations to those obtained by Fernández-Villaverde and Rubio-Ramírez (2005).

S5.1. *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 we 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, and b_2b_0.

Table S.VI documents the results. We use Smolyak levels 2, 3, and 4 for the models with four and six states, and afterward use only levels 2 and 3. The Smolyak operator is already superior to the tensor operator for a small model with four 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 six states, whereas the Smolyak operator is still doing fine. The biggest model we are able to solve has 22 states and takes around 68 minutes for an approximation error of $1.7E-5$.

S5.2. *Estimation*

In the next subsection, we compare the likelihood values of the Smolyak Kalman filter, Smolyak particle filter, Smolyak sum, and 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 observables in the measurement model are investment i, labor l, and output y. The measurement shocks are assumed to be additive.

TABLE S.VI

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

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 S.VII. 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-

TABLE S.VII

MEASUREMENT ERRORS

	Small	Large
σ_i	$8.66E{-}4$	$4.65E{-}3$
σ_l	$1.10E{-}3$	$3.12E{-}3$
σ_y	$1.58E{-}4$	$1.75E{-}2$

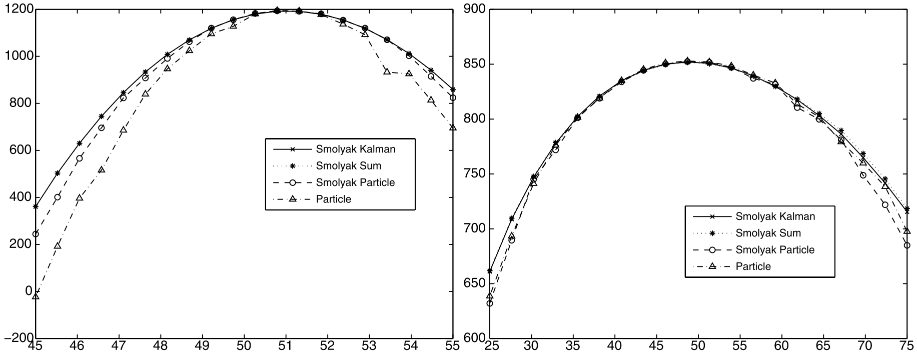

FIGURE S.1.—Likelihood at true parameters for $\tau = 50.0$ and $\sigma_a = 0.035$.

Ramírez (2005), who used the same model to illustrate and test their algorithms.

S5.2.1. *Likelihood*

The particle filter is run with 40,000 particles, and the Smolyak Kalman filter is run 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 is run 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.

Figures S.1 and S.2 show slices through the multidimensional likelihood. The left plots show likelihood values from the data with small measurement errors; the right plots show values with large measurement errors. We set all parameters to their true values and vary τ at the abscissas, and plot them against the likelihood values at the ordinates. The results are rather encouraging for

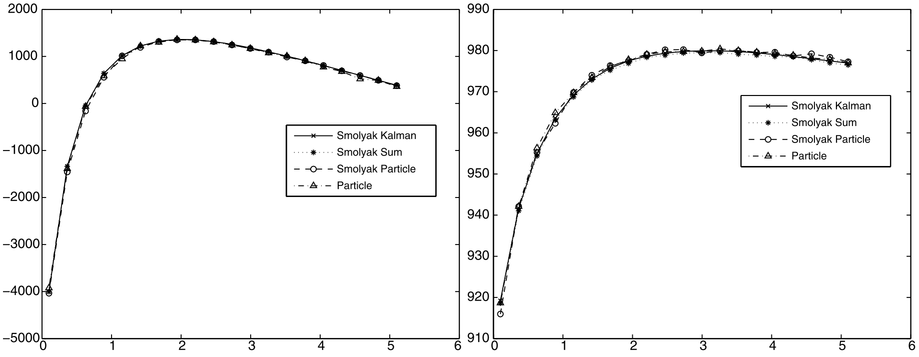

FIGURE S.2.—Likelihood at true parameters for $\tau = 2.0$ and $\sigma_a = 0.007$.

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 S.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.

S5.2.2. *Parameters*

We have implemented an interactive sampling environment in JBendge where the Metropolis–Hastings parameters driving the innovation variance γ^{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 other sequences. This serves the purpose of manually cancelling some sequences that no longer improve. 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 γ^{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 stage searches for the posterior mode. At this stage, the mixing parameter γ^{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 γ^{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 γ^{GE} and b to obtain an acceptance ratio around 0.3. For all estimations, parameter b is set to $1E{-}6$ and γ^{GE} is set between 0.1 and 0.4. Both parameters are remarkably stable especially across the linear and nonlinear estimations. This helps 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.

TABLE S.VIII

BOUNDS OF A UNIFORM PRIOR

	Low	Hi		Low	Hi
α	0.00	1.00	σ_a	0.0	0.1
β	0.75	1.00	σ_i	0.0	0.1
δ	0.00	0.05	σ_l	0.0	0.1
ρ	0.00	1.00	σ_y	0.0	0.1
τ	0.00	100			
θ	0.00	1.00			

We run the estimation with approximately 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 S.VIII.

It takes approximately 1000–2000 draws for each of the 32 sequences to find the mode of the posterior density and another 1000–4000 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 hours with the Smolyak sum filter, and 20 hours 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" shows 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 report only estimates of the model with $\tau = 50$ and $\sigma_a = 0.35$ for small and large measurement errors. The data are generated via a very accurate solution with integration and solution approximation level 5, and a function iteration tolerance of $1E-10$. The approximation error is around $1E-10$.

Linear versus Nonlinear Estimation. We compare the linear and the nonlinear estimations based on nonlinearly simulated data and the simple Smolyak Kalman filter.

TABLE S.IX

KALMAN AND SMOLYAK KALMAN FILTER, SMALL ERRORS

	True	Mean	SD	ML	Mean	SD	ML
θ	3.57_{-1}	3.496_{-1}	1.9_{-3}	3.487_{-1}	3.569_{-1}	2.6_{-3}	3.572_{-1}
β	9.90_{-1}	9.957_{-1}	5.5_{-4}	9.960_{-1}	9.906_{-1}	4.4_{-3}	9.933_{-1}
α	4.00_{-1}	3.839_{-1}	5.1_{-3}	3.815_{-1}	4.003_{-1}	6.7_{-3}	4.010_{-1}
ρ	9.50_{-1}	9.641_{-1}	3.0_{-3}	9.659_{-1}	9.506_{-1}	1.6_{-3}	9.523_{-1}
δ	2.00_{-2}	1.796_{-2}	1.2_{-3}	1.737_{-2}	1.957_{-2}	5.3_{-4}	1.934_{-2}
τ	5.00_{+1}	1.946_{+1}	$2.7_{\pm 0}$	1.784_{+1}	5.255_{+1}	$7.6_{\pm 0}$	4.447_{+1}
σ_a	3.50_{-2}	3.716_{-2}	2.7_{-3}	3.798_{-2}	3.604_{-2}	1.7_{-3}	3.704_{-2}
σ_y	1.58_{-4}	8.521_{-4}	7.0_{-4}	1.730_{-4}	8.724_{-4}	4.0_{-4}	1.061_{-3}
σ_i	8.66_{-4}	1.996_{-3}	3.4_{-4}	2.030_{-3}	4.981_{-4}	2.8_{-4}	4.057_{-4}
σ_l	1.10_{-3}	1.438_{-3}	1.1_{-4}	1.395_{-3}	1.199_{-3}	8.4_{-5}	1.168_{-3}

Table S.IX 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 is close to the true parameters of the data generating process, the Kalman filter shows clear biases. The nonlinear filter has problems accurately estimating the parameter τ, 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 S.3 shows the posterior density estimated by the Smolyak Kalman filter. Black vertical bars indicate the true parameter values.

Lower Nonlinear Accuracy. The next calculations show the accuracy needed for the nonlinear solution approximation and the Smolyak Kalman filter.

Table S.X shows, on the left side, estimates with solution, rational expectations, 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 the results of the estimates with a solution level 3 and filter 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

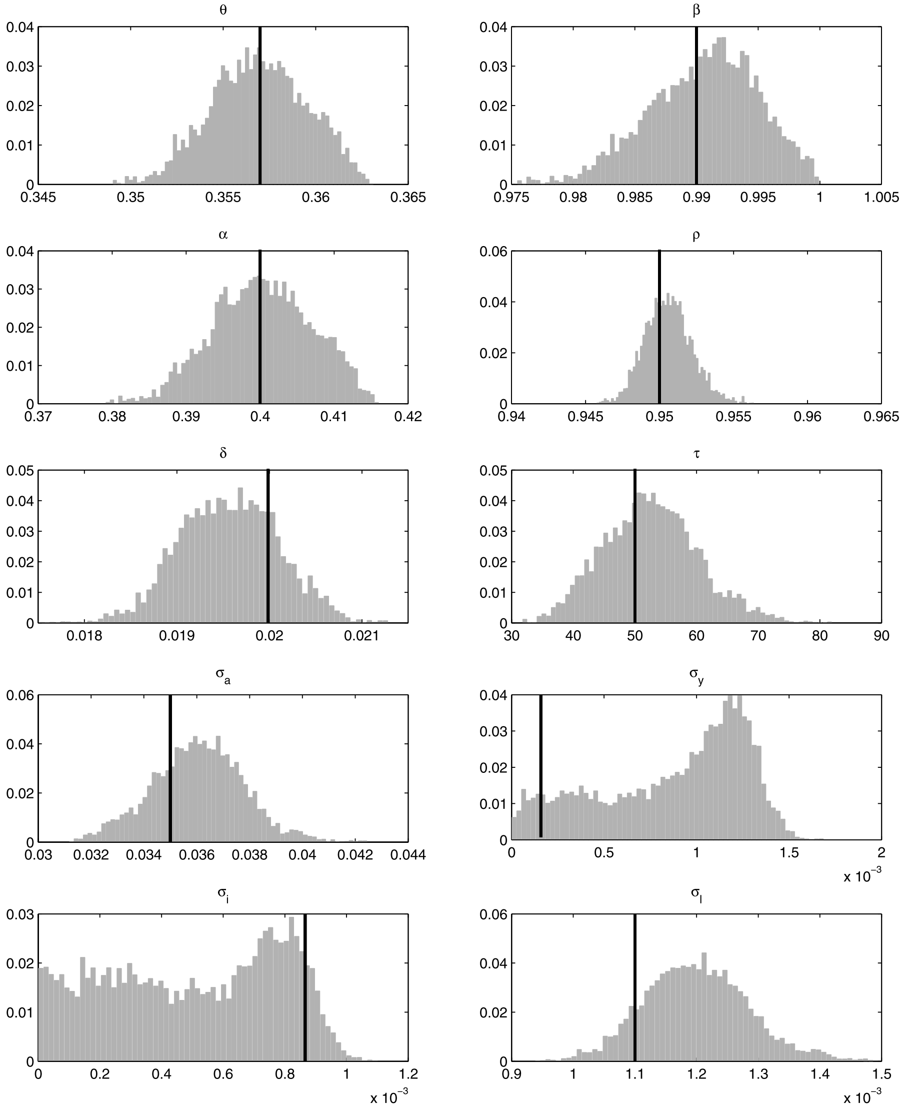

FIGURE S.3.—Parameter posterior, Smolyak Kalman filter.

filter integration levels 2 and 3 imply similar results, but solution level 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 S.X

SMOLYAK KALMAN FILTERS, SMALL ERRORS

	True	Mean	SD	ML	Mean	SD	ML
θ	3.57_{-1}	3.586_{-1}	2.6_{-3}	3.596_{-1}	3.436_{-1}	2.4_{-3}	3.430_{-1}
β	9.90_{-1}	9.914_{-1}	4.7_{-3}	9.948_{-1}	9.529_{-1}	7.9_{-3}	9.608_{-1}
α	4.00_{-1}	4.047_{-1}	6.7_{-3}	4.074_{-1}	3.641_{-1}	6.8_{-3}	3.625_{-1}
ρ	9.50_{-1}	9.501_{-1}	1.9_{-3}	9.513_{-1}	9.552_{-1}	1.2_{-3}	9.559_{-1}
δ	2.00_{-2}	1.996_{-2}	6.1_{-4}	2.004_{-2}	1.936_{-2}	7.4_{-4}	1.889_{-2}
τ	5.00_{+1}	5.516_{+1}	1.1_{+1}	4.690_{+1}	6.708_{+1}	$8.3_{\pm 0}$	5.901_{+1}
σ_a	3.50_{-2}	3.700_{-2}	1.8_{-3}	3.790_{-2}	3.802_{-2}	2.6_{-3}	3.821_{-2}
σ_y	1.58_{-4}	8.012_{-4}	4.1_{-4}	1.080_{-3}	1.551_{-3}	5.1_{-4}	1.833_{-3}
σ_i	8.66_{-4}	5.291_{-4}	2.8_{-4}	3.087_{-4}	6.005_{-4}	4.0_{-4}	4.822_{-5}
σ_l	1.10_{-3}	1.194_{-3}	8.3_{-5}	1.160_{-3}	1.211_{-3}	9.5_{-5}	1.204_{-3}

Nonlinear Filters Comparison. This paragraph compares the Smolyak Kalman filter with the Smolyak particle filter.

Table S.XI shows the other two nonlinear filters. On the left side of the table, 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 five 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 except τ and the

TABLE S.XI

SMOLYAK SUM AND SMOLYAK PARTICLE FILTERS, SMALL ERRORS

	True	Mean	SD	ML	Mean	SD	ML
θ	3.57_{-1}	3.593_{-1}	2.0_{-3}	3.611_{-1}	3.560_{-1}	3.7_{-3}	3.607_{-1}
β	9.90_{-1}	9.927_{-1}	4.5_{-3}	9.968_{-1}	9.870_{-1}	6.8_{-3}	9.952_{-1}
α	4.00_{-1}	4.063_{-1}	5.1_{-3}	4.112_{-1}	3.978_{-1}	9.9_{-3}	4.103_{-1}
ρ	9.50_{-1}	9.505_{-1}	2.4_{-3}	9.513_{-1}	9.503_{-1}	1.5_{-3}	9.498_{-1}
δ	2.00_{-2}	2.011_{-2}	6.3_{-4}	2.043_{-2}	1.973_{-2}	6.7_{-4}	2.047_{-2}
τ	5.00_{+1}	5.372_{+1}	1.5_{+1}	4.463_{+1}	5.467_{+1}	$6.9_{\pm 0}$	5.244_{+1}
σ_a	3.50_{-2}	3.647_{-2}	1.9_{-3}	3.765_{-2}	3.619_{-2}	1.8_{-3}	3.704_{-2}
σ_y	1.58_{-4}	9.368_{-4}	3.3_{-4}	1.080_{-3}	9.005_{-4}	3.6_{-4}	1.124_{-3}
σ_i	8.66_{-4}	4.941_{-4}	2.6_{-4}	2.873_{-4}	5.201_{-4}	2.8_{-4}	3.638_{-4}
σ_l	1.10_{-3}	1.191_{-3}	9.0_{-5}	1.141_{-3}	1.184_{-3}	9.2_{-5}	1.119_{-3}

TABLE S.XII

KALMAN AND EXTENDED KALMAN FILTERS, SMALL ERRORS

	True	Mean	SD	ML	Mean	SD	ML
θ	3.57_{-1}	3.496_{-1}	1.9_{-3}	3.487_{-1}	3.526_{-1}	1.6_{-3}	3.515_{-1}
β	9.90_{-1}	9.957_{-1}	5.5_{-4}	9.960_{-1}	9.950_{-1}	2.8_{-4}	9.950_{-1}
α	4.00_{-1}	3.839_{-1}	5.1_{-3}	3.815_{-1}	3.890_{-1}	4.2_{-3}	3.861_{-1}
ρ	9.50_{-1}	9.641_{-1}	3.0_{-3}	9.659_{-1}	9.631_{-1}	1.0_{-3}	9.633_{-1}
δ	2.00_{-2}	1.796_{-2}	1.2_{-3}	1.737_{-2}	1.817_{-2}	8.0_{-4}	1.773_{-2}
τ	5.00_{+1}	1.946_{+1}	$2.7_{\pm 0}$	1.784_{+1}	2.045_{+1}	$1.5_{\pm 0}$	1.982_{+1}
σ_a	3.50_{-2}	3.716_{-2}	2.7_{-3}	3.798_{-2}	3.517_{-2}	2.2_{-3}	3.408_{-2}
σ_y	1.58_{-4}	8.521_{-4}	7.0_{-4}	1.730_{-4}	8.735_{-4}	6.7_{-4}	1.044_{-5}
σ_i	8.66_{-4}	1.996_{-3}	3.4_{-4}	2.030_{-3}	1.673_{-3}	3.0_{-4}	1.844_{-3}
σ_l	1.10_{-3}	1.438_{-3}	1.1_{-4}	1.395_{-3}	1.448_{-3}	1.0_{-4}	1.411_{-3}

measurement error standard deviations of output and investment. The estimates are essentially the same for all our nonlinear filters.

Extended Kalman Filter. In Table S.XII 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 those of τ.

The next estimations process data generated with large measurement errors.

Nonlinear Filters Comparison. The first of these estimations is in Table S.XIII, which compares the Smolyak Kalman filter with solution and integration level 4 and the Smolyak particle filter with solution level 3, filter level 2, and 500 particles. Both results are again very similar and differ only in the maximum likelihood estimates of τ. Given the high inaccuracy for this parameter, indicated by the high standard deviation of the posterior, this difference is

TABLE S.XIII

SMOLYAK SUM AND SMOLYAK PARTICLE FILTERS, LARGE ERRORS

	True	Mean	SD	ML	Mean	SD	ML
θ	3.57_{-1}	3.562_{-1}	5.2_{-3}	3.572_{-1}	3.543_{-1}	6.8_{-3}	3.575_{-1}
β	9.90_{-1}	9.873_{-1}	7.7_{-3}	9.822_{-1}	9.864_{-1}	8.6_{-3}	9.897_{-1}
α	4.00_{-1}	3.979_{-1}	1.4_{-2}	4.007_{-1}	3.928_{-1}	1.8_{-2}	4.016_{-1}
ρ	9.50_{-1}	9.437_{-1}	8.4_{-3}	9.400_{-1}	9.472_{-1}	8.9_{-3}	9.444_{-1}
δ	2.00_{-2}	2.038_{-2}	2.2_{-3}	2.053_{-2}	1.966_{-2}	3.0_{-3}	2.085_{-2}
τ	5.00_{+1}	5.719_{+1}	1.9_{+1}	8.012_{+1}	5.403_{+1}	2.0_{+1}	4.873_{+1}
σ_a	3.50_{-2}	3.357_{-2}	2.7_{-3}	3.303_{-2}	3.385_{-2}	2.4_{-3}	3.422_{-2}
σ_y	8.75_{-3}	1.680_{-2}	2.1_{-3}	1.597_{-2}	1.696_{-2}	2.1_{-3}	1.745_{-2}
σ_i	2.33_{-3}	4.969_{-3}	2.5_{-3}	5.671_{-3}	4.726_{-3}	2.4_{-3}	5.044_{-3}
σ_l	1.56_{-3}	2.873_{-3}	2.1_{-4}	2.700_{-3}	2.882_{-3}	2.1_{-4}	2.888_{-3}

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 at minimum, and 853 at maximum. For the Smolyak particle filter, these numbers are 851, 841, and 852. Again the measurement shocks are hardly identified and the parameter τ is badly estimated as well.

Smolyak Sum Filter. The last nonlinear estimation we report in Table S.XIV tests whether increasing the number of summands in the Smolyak sum filter improves the estimation. Both estimations are run with solution level 3 and filter level 2. 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: the only difference is again the maximum likelihood estimates of τ, 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, and 853: for 20 summands they are 851, 837, and 853.

Nonidentification of τ. We finally investigate whether the problem we encounter with the estimation of τ is due to the errors of the approximations or a property of the model. Table S.XV shows the first approach to this issue 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 τ 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.

TABLE S.XIV
SMOLYAK SUM FILTERS, LARGE ERRORS

	True	Mean	SD	ML	Mean	SD	ML
θ	3.57_{-1}	3.570_{-1}	5.9_{-3}	3.553_{-1}	3.558_{-1}	6.6_{-3}	3.551_{-1}
β	9.90_{-1}	9.885_{-1}	7.4_{-3}	9.845_{-1}	9.873_{-1}	6.8_{-3}	9.813_{-1}
α	4.00_{-1}	3.998_{-1}	1.5_{-2}	3.954_{-1}	3.966_{-1}	1.7_{-2}	3.947_{-1}
ρ	9.50_{-1}	9.450_{-1}	8.5_{-3}	9.450_{-1}	9.451_{-1}	9.3_{-3}	9.436_{-1}
δ	2.00_{-2}	2.023_{-2}	2.9_{-3}	1.945_{-2}	2.014_{-2}	3.1_{-3}	2.060_{-2}
τ	5.00_{+1}	6.060_{+1}	2.0_{+1}	7.046_{+1}	5.777_{+1}	1.9_{+1}	6.039_{+1}
σ_a	3.50_{-2}	3.364_{-2}	2.3_{-3}	3.345_{-2}	3.373_{-2}	2.5_{-3}	3.283_{-2}
σ_y	8.75_{-3}	1.671_{-2}	2.2_{-3}	1.742_{-2}	1.738_{-2}	2.0_{-3}	1.739_{-2}
σ_i	2.33_{-3}	5.027_{-3}	2.6_{-3}	3.332_{-3}	4.393_{-3}	2.5_{-3}	4.414_{-3}
σ_l	1.56_{-3}	2.878_{-3}	2.2_{-4}	2.951_{-3}	2.872_{-3}	2.1_{-4}	2.866_{-3}

TABLE S.XV

KALMAN FILTER, LINEAR MODEL, SMALL ERRORS

	True	Mean	SD	ML	Mean	SD	ML
θ	3.57_{-1}	3.615_{-1}	3.0_{-3}	3.583_{-1}	3.615_{-1}	3.0_{-3}	3.583_{-1}
β	9.90_{-1}	9.885_{-1}	1.5_{-3}	9.903_{-1}	9.885_{-1}	1.5_{-3}	9.903_{-1}
α	4.00_{-1}	4.125_{-1}	7.8_{-3}	4.040_{-1}	4.125_{-1}	7.8_{-3}	4.040_{-1}
ρ	9.50_{-1}	9.451_{-1}	5.0_{-3}	9.509_{-1}	9.451_{-1}	5.0_{-3}	9.509_{-1}
δ	2.00_{-2}	2.267_{-2}	1.8_{-3}	2.063_{-2}	2.267_{-2}	1.8_{-3}	2.063_{-2}
τ	5.00_{+1}	6.086_{+1}	2.0_{+1}	3.786_{+1}	6.086_{+1}	2.0_{+1}	3.786_{+1}
σ_a	3.50_{-2}	3.260_{-2}	2.2_{-3}	3.155_{-2}	3.260_{-2}	2.2_{-3}	3.155_{-2}
σ_y	1.58_{-4}	8.753_{-4}	4.1_{-4}	2.369_{-4}	8.753_{-4}	4.1_{-4}	2.369_{-4}
σ_i	8.66_{-4}	5.518_{-4}	2.8_{-4}	8.196_{-4}	5.518_{-4}	2.8_{-4}	8.196_{-4}
σ_l	1.10_{-3}	1.145_{-3}	8.5_{-5}	1.095_{-3}	1.145_{-3}	8.5_{-5}	1.095_{-3}

A more convincing point is made in Table S.XVI. It also demonstrates that peaked likelihood slices are not necessarily informative with regard to the standard deviation of a parameter estimate, since the likelihood slices seem to be properly peaked for τ at the true parameter values in the left plot of Figure S.1. The left column "Smallest τ" shows the figures with the smallest τ 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 τ, where we use the smallest value of 31.88, the log likelihood collapses. If, in addition to τ, we change

TABLE S.XVI

POOR IDENTIFICATION OF τ

	Smallest τ	True Values	Changes
θ	3.581_{-1}	3.570_{-1}	3.570_{-1}
β	9.964_{-1}	9.900_{-1}	$9.9\mathbf{64}_{-1}$
α	4.028_{-1}	4.000_{-1}	4.000_{-1}
ρ	9.552_{-1}	9.500_{-1}	$9.5\mathbf{52}_{-1}$
δ	1.911_{-2}	2.000_{-2}	$\mathbf{1.911}_{-2}$
τ	3.188_{+1}	3.188_{+1}	3.188_{+1}
σ_a	3.919_{-2}	3.500_{-2}	3.500_{-2}
σ_y	1.149_{-3}	1.580_{-4}	1.580_{-4}
σ_i	3.173_{-4}	8.660_{-4}	8.660_{-4}
σ_l	1.260_{-3}	1.100_{-3}	1.100_{-3}
		Log Likelihood	
Smolyak Kalman	1193	−8167	1164
Particle	1189	−10761	1159

the parameters β, ρ, and δ toward those we obtained together with the smallest τ, the likelihood value recovers. This is done in column "Change," where bold faced numbers show the changed parameters compared to the true values. Further changes of the parameters toward the "Smallest τ" parameterization bring us back to the likelihood of 1193 and 1189.

We conclude that the poor estimates of τ 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.

Nonidentification of σ_y. A substantial change of the parameter σ_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 S.XVII we finally look at the likelihood slice for σ_y at the true parameters. The σ_y values are between the minimal and maximal values of the posterior sample. The variations between 1167 and 1174 for the Smolyak Kalman filter and 1169 and 1177 for the particle filter indicate a flat likelihood, in the sense that this variation is close to the variation of the log likelihood we usually see during Metropolis–Hastings sampling. We ascribe the problem with measurement errors to lack of proper identification of this model and to the ad hoc solution of adding some measurement errors. The identification problem was also discussed by Fernández-Villaverde and Rubio-Ramírez (2007), who obtained more accurate estimates, however.

Marginal Likelihood. The last calculation is the marginal likelihood in Table S.XVIII. The upper part shows the results of the estimations based on 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 and other filters. A positive number favors the Smolyak Kalman filter; a negative number indicates superiority of other filters. The upper three

TABLE S.XVII

LIKELIHOOD SLICE AT σ_y

σ_y	Particle	Smolyak Kalman
3.78_{-04}	1169	1170
5.67_{-04}	1172	1171
7.56_{-04}	1174	1172
9.45_{-04}	1177	1174
1.13_{-03}	1175	1172
1.32_{-03}	1177	1171
1.51_{-03}	1175	1169
1.70_{-03}	1171	1167

TABLE S.XVIII

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

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. We see a further improvement in the marginal likelihood compared to the Smolyak Kalman filter.

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

REFERENCES

ALSBACH, D., AND H. SORENSON (1972): "Nonlinear Estimation Using Gaussian Sum Approximation," *IEEE Transactions on Automated Control*, 17, 439–448.

AMISANO, G., AND O. TRISTANI (2007): "Euro Area Inflation Persistence in an Estimated Nonlinear DSGE Model," Discussion Paper 6373, CEPR.

ANDERSON, B. D. O., AND J. B. MOORE (1979): *Optimal Filtering*. Mineola, NY: Dover Publications.

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, 174–188.

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, 1403–1412.

BROOKS, S. P., AND A. GELMAN (1998): "General Methods for Monitoring Convergence of Iterative Simulations," *Journal of Computational and Graphical Statistics*, 7, 434–455.

CHIB, S., AND E. GREENBERG (1995): "Understanding the Metropolis–Hastings Algorithm," *The American Statistician*, 49, 327–335.

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.

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

GELFAND, A., AND D. DEY (1994): "Bayesian Model Choice: Asymptotics and Exact Calculations," *Journal of the Royal Statistical Society, Ser. B*, 56, 501–514.

GELMAN, A., AND D. B. RUBIN (1992): "Inference From Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7, 457–472.

GENZ, A., AND C. KEISTER (1996): "Fully Symmetric Interpolatory Rules for Multiple Intergrals 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.

HEISS, F., AND V. WINSCHEL (2008): "Likelihood Approximation by Numerical Integration on Sparse Grids," *Journal of Econometrics*, 144, 62–80.

JUDD, K. L. (1992): "Projection Methods for Solving Aggregate Growth Models," *Journal of Economic Theory*, 58, 410–452.

JULIER, S. J., AND J. K. UHLMANN (1997): "A New Extension of the Kalman Filter to Nonlinear Systems," in *Proceedings of Aerosense: 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, FL.

KALMAN, R. E. (1960): "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME—Journal of Basic Engineering*, 82, 35–45.

KLEIN, P. (2000): "Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model," *Journal of Economic Dynamics and Control*, 24, 1405–1423.

KOTECHA, J., AND P. DJURIĆ (2003): "Gaussian Sum Particle Filter," *IEEE Transactions on Signal Processing*, 51, 2602–2612.

KÜBLER, F., AND D. KRÜGER (2004): "Computing Equilibrium in OLG Models With Stochastic Production," *Journal of Economic Dynamics and Control*, 28, 1411–1436.

STORN, R., AND K. PRICE (1997): "Differential Evolution—A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," *Journal of Global Optimisation*, 11, 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, 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.

Dept. of Economics, L7, 3-5, University of Mannheim, 68131 Mannheim, Germany; winschel@rumms.uni-mannheim.de

and

School of Business and Economics, Institute of Statistics and Econometrics, Humboldt-Universität zu Berlin, 10099 Berlin, Germany; mk@mk-home.de.

Manuscript received February, 2006; final revision received January, 2009.