



Contents lists available at ScienceDirect

## Journal of Economic Dynamics &amp; Control

journal homepage: [www.elsevier.com/locate/jedc](http://www.elsevier.com/locate/jedc)

# Comparison of solutions to the multi-country Real Business Cycle model

Robert Kollmann<sup>a,b,c,\*</sup>, Serguei Maliar<sup>d,e</sup>, Benjamin A. Malin<sup>f</sup>, Paul Pichler<sup>g</sup>

<sup>a</sup> ECARES, Université Libre de Bruxelles, CP 114, 50 Av. F. Roosevelt, B-1050 Brussels, Belgium

<sup>b</sup> Faculté de Sciences Economiques, Université Paris-Est, 61 Av. du Gén. de Gaulle, F-94000 Créteil, France

<sup>c</sup> CEPR, 53-56 Gt. Sutton Street, London EC1V 0DG, United Kingdom

<sup>d</sup> Hoover Institution, 434 Galvez Mall, Stanford University, Stanford, CA 94305-6010, USA

<sup>e</sup> Department Fundamentos del Analisis Economico, Universidad de Alicante, Campus San Vicente del Raspeig, Ap. Correos 99, 03080 Alicante, Spain

<sup>f</sup> Federal Reserve Board, Washington, DC 20551, USA

<sup>g</sup> Economic Studies Division, Oesterreichische Nationalbank, P.O. Box 61, A-1011 Vienna, Austria

## ARTICLE INFO

### Article history:

Received 11 March 2010

Accepted 5 July 2010

Available online 1 October 2010

### JEL classification:

C63

### Keywords:

Numerical solutions

Simulations

Approximations

Algorithms

## ABSTRACT

We compare the performance of perturbation, projection, and stochastic simulation algorithms for solving the multi-country RBC model described in Den Haan et al. (this issue). The main challenge of solving this model comes from its large number of continuous-valued state variables, ranging between four and 20 in the specifications we consider. The algorithms differ substantially in terms of speed and accuracy, and a clear trade-off exists between the two. Perturbation methods are very fast but invoke large approximation errors except at points close to the steady state; the projection methods considered are accurate on a large area of the state space but are very slow for specifications with many state variables; stochastic simulation methods have lower accuracy than projection methods, but their computational cost increases only moderately with the state-space dimension. Simulated series generated by different methods can differ noticeably, but only small differences are found in unconditional moments of simulated variables. On the basis of our comparison, we identify the factors that account for differences in accuracy and speed across methods, and we suggest directions for further improvement of some approaches.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

This paper compares six different algorithms for solving the multi-country Real Business Cycle (RBC) model described in Den Haan et al. (this issue). This model does not have a closed-form solution, and the specifications we consider are numerically challenging because they have a large number of continuous-valued state variables (up to 20). Large dynamic stochastic general equilibrium (DSGE) models of the type considered here are the workhorses of modern international macroeconomics and provide the core of macro models used by many policy institutions, including central banks. A wide array of solution methods for DSGE models have been presented in recent years, and it is important for researchers to understand the relative performance of these methods in terms of accuracy and computational intensity. To date,

\* Corresponding author at: ECARES, Université Libre de Bruxelles, CP 114, 50 Av. F. Roosevelt, B-1050 Brussels, Belgium.

E-mail addresses: [robert\\_kollmann@yahoo.com](mailto:robert_kollmann@yahoo.com) (R. Kollmann), [maliaars@stanford.edu](mailto:maliaars@stanford.edu) (S. Maliar), [Benjamin.A.Malin@frb.gov](mailto:Benjamin.A.Malin@frb.gov) (B.A. Malin), [paul.pichler@oenb.at](mailto:paul.pichler@oenb.at) (P. Pichler).

**Table 1**  
Participating algorithms.

Abbreviation	Participants	Programming language
PER1	Robert Kollmann, Jinill Kim, Sunghyun Kim	MATLAB
PER2	Robert Kollmann, Jinill Kim, Sunghyun Kim	MATLAB
MRGAL	Paul Pichler	MATLAB
SMOL	Benjamin Malin, Dirk Krueger, Felix Kubler	FORTRAN
CGA	Serguei Maliar, Lilia Maliar, Kenneth Judd	MATLAB
SSA	Serguei Maliar, Lilia Maliar, Kenneth Judd	MATLAB

systematic comparisons between numerical solution methods have focused on smaller DSGE models,<sup>1</sup> but little research has been done to compare methods for solving much larger models. This project is a step towards filling that gap in the literature.<sup>2</sup>

Specifically, we compare the six solution methods listed in Table 1—some of which are well known, while others are novel and not (yet) widely used. These methods can be roughly categorized into three classes, as distinguished by Judd et al. (2009): perturbation, projection, and stochastic simulation algorithms.<sup>3</sup>

*Perturbation* methods solve for the coefficients of Taylor expansions of the true decision rules around the (non-stochastic) steady state. The first- and second-order methods (PER1 and PER2) discussed in Kollmann et al. (this issue) represent this class.

*Projection* methods construct a grid of points in the state space, define approximating policy functions together with a loss function measuring the fit of the approximation, and solve for the coefficients of the (approximate) policy functions to minimize the loss function. The *Smolyak-collocation* method (SMOL) of Malin et al. (this issue), the *monomial-rule Galerkin* method (MRGAL) of Pichler (this issue), and the *cluster-grid* algorithm (CGA) of Maliar et al. (this issue) belong to this class.

*Stochastic-simulation* algorithms run simulations to simultaneously compute the ergodic distribution of state variables and the approximate policy functions. This class is represented by the (first-order) *stochastic-simulation* algorithm (SSA) of Maliar et al. (this issue).

A thorough discussion of the methods' novel features and their relationship to the literature is mostly left to the individual papers cited above, but here (and in Section 2), we do note that all methods employ strategies that allow them to be feasible for high-dimensional problems. In particular, the projection methods incorporate novel elements that keep the number of function evaluations required to find an approximate solution from growing too rapidly with the number of state variables. These involve evaluating policy functions at small sets of points (rather than a tensor-product grid), using cheap monomial integration formulae to compute conditional expectations (rather than Gaussian product rules), and using efficient strategies to solve for control variables (avoiding numerical non-linear equations solving). For SSA, the simulation length (and thus, the number of points used for both policy function evaluations and Monte Carlo integration) does not vary with the number of state variables, so the computational cost increases only moderately with the dimension of the problem. Finally, perturbation methods are, by construction, well suited for the high-dimensional problems considered here because they only require evaluations at the model's steady state.

The following are the main lessons learned from this comparison project:

- The methods differ substantially with respect to speed and accuracy, and a clear trade-off exists between the two. The projection methods, which are the most accurate over most of the state space, are also the slowest when the number of state variables is large. Compared to projection methods, the stochastic simulation algorithm is less accurate, but its computing time grows less quickly with dimensionality. The perturbation methods are fastest but are also the least accurate (except at points very near the steady state). The trade-off between speed and accuracy is illustrated by the following statistics: on a simulation run of 10,000 periods, the maximum errors across all equilibrium conditions and all model specifications generated by the six methods (in parentheses) are 0.009% (CGA), 0.030% (SMOL), 0.115% (MRGAL), 0.145% (SSA), 1.349% (PER2) and 6.310% (PER1); the maximum computing times have (essentially) the reverse order: 0.4 s (PER1), 6.8 s (PER2), 1 h 32 min (MRGAL), 3 h 18 min (SSA), 12 h 45 min (SMOL), and 43 h 53 min (CGA).
- The relative speed and accuracy of the methods here can be understood in light of differences in: (i) whether the method is local or global; (ii) the functional form chosen for approximating the policy functions; (iii) whether the approximate policy functions are all explicitly parameterized functions of only the state variables, or if some are more flexibly specified functions of other policy functions; (iv) the numerical integration method used to evaluate conditional expectations in the model's Euler equations.

<sup>1</sup> See Taylor and Uhlig (1990), Gaspar and Judd (1997), Marimon and Scott (1999), Aruoba et al. (2006), and Heer and Maussner (2008), among others.

<sup>2</sup> The recent work of Den Haan (2010) is of a similar spirit. It compared solution methods for a model with a *continuum* of agents and state variables. By contrast, the multi-country RBC model that is the focus of this paper has a large, but finite, number of agents and state variables.

<sup>3</sup> Detailed discussions of projection and perturbation approaches are provided in Gaspar and Judd (1997) and Judd (1998); Judd et al. (2009) discuss stochastic-simulation algorithms.

- Linear approximation methods deliver policy functions that may differ noticeably from those generated by higher-order methods—especially at points far from the steady state. Simulated time series can also differ substantially across methods. However, all methods deliver similar unconditional moments of simulated variables. Even the perturbation methods, which are least accurate (i.e., generate the largest approximation errors in model equations), produce moments that are close to those delivered by the (most-accurate) projection methods. Given this, and in light of the trade-off between speed and accuracy, a researcher's choice of solution method should depend on the specific application (or question of interest), rather than simply taking the most-accurate method.<sup>4</sup>

We organize the rest of the paper as follows. Section 2 briefly describes the six algorithms and discusses their strategies for handling high-dimensional state spaces—more detailed discussions can be found in the individual contributions to the JEDC special issue. Section 3 compares the computational speed of the methods, while Section 4 evaluates accuracy. Section 5 compares approximate policy functions, simulated series, and moments of key variables generated by the algorithms. Section 6 concludes.

## 2. Description of the algorithms

To facilitate the description of the algorithms, we first discuss key common features of the 30 model specifications considered here.

### 2.1. The models

The  $N$ -country RBC models have  $N$  predetermined endogenous state variables, namely the capital stocks of the countries. Let  $k_t$  denote the  $(N \times 1)$  vector of beginning-of-period- $t$  capital stocks, and  $a_t$  the  $(N \times 1)$  vector of exogenous log total factor productivities of the  $N$  countries at  $t$ . There are  $3N+1$  non-predetermined variables (controls) chosen in period  $t$ ; let  $z_t \equiv (\lambda_t, c_t, i_t, l_t)$  denote the vector of controls, where  $\lambda_t$  is the (scalar) Lagrange multiplier on the resource constraint, while  $c_t$ ,  $i_t$ , and  $l_t$  are  $(N \times 1)$  vectors of consumption, investment and hours worked in the  $N$  countries.

Each model here is defined by two sets of equilibrium conditions:  $N$  intertemporal Euler equations of the format shown in Eq. (1), and  $3N+1$  other conditions (world resource constraint, laws of motion for capital, and static optimality conditions) shown in Eq. (2):

$$\mathbb{E}_t \mathcal{H}(z_{t+1}, z_t, k_{t+1}, k_t, a_{t+1}, a_t) = 0, \quad (1)$$

$$\mathcal{Q}(z_t, k_{t+1}, k_t, a_t) = 0, \quad (2)$$

where  $\mathbb{E}_t$  is the expectations operator in period  $t$ , while  $\mathcal{H}$  and  $\mathcal{Q}$  are functions.<sup>5</sup> Eq. (2) pins down  $z_t$ , for given values of  $k_{t+1}$ ,  $k_t$  and  $a_t$ , and we refer to (2) as the “intra-temporal” conditions of the model. The law of motion for productivity is given by:  $a_{t+1} = \rho a_t + \varepsilon_{t+1}$ , where  $|\rho| < 1$  is a scalar and  $\varepsilon_{t+1}$  is an  $(N \times 1)$  vector of normally-distributed white noise.

The model solution is given by policy functions  $k_{t+1} = F(k_t, a_t)$  and  $z_t = G(k_t, a_t)$ , that satisfy

$$\mathbb{E}_t \mathcal{H}(G(F(k_t, a_t), \rho a_t + \varepsilon_{t+1}), G(k_t, a_t), F(k_t, a_t), k_t, \rho a_t + \varepsilon_{t+1}, a_t) = 0, \quad (3)$$

$$\mathcal{Q}(G(k_t, a_t), F(k_t, a_t), k_t, a_t) = 0, \quad (4)$$

for all values of  $(k_t, a_t)$ .

The numerical solution methods approximate the true policy functions  $F, G$  by parametric functions  $\hat{F}, \hat{G}$ . The (local) perturbation methods construct these functions as the Taylor series expansions of the true policy functions around the model's non-stochastic steady state. The (global) projection and stochastic simulation methods, on the other hand, determine  $\hat{F}, \hat{G}$  to yield small approximation errors (over some domain):

$$\mathbb{E}_t \mathcal{H}(\hat{G}(\hat{F}(k_t, a_t), \rho a_t + \varepsilon_{t+1}), \hat{G}(k_t, a_t), \hat{F}(k_t, a_t), k_t, \rho a_t + \varepsilon_{t+1}) \approx 0, \quad (5)$$

$$\mathcal{Q}(\hat{G}(k_t, a_t), \hat{F}(k_t, a_t), k_t, a_t) \approx 0. \quad (6)$$

### 2.2. The algorithms

All six methods approximate the policy functions for end-of-period capital holdings  $k_{t+1}$  (i.e.,  $\hat{F}$ ) using polynomials of the state variables  $k_t$  and  $a_t$ . However, the degree of the polynomials differs across methods, as shown in Table 2

<sup>4</sup> For applications in which accuracy on the ergodic distribution is a primary concern, CGA is the most appropriate method. For analyzing economies that start far from their steady state, SMOL, which is accurate for large parts of the state space, is likely a better choice. However, for applications that require repeated computation of model solutions (as in simulation-based inference), the perturbation methods (high speed and adequate unconditional moments) are the best methods to use.

<sup>5</sup>  $\mathcal{H}$  maps  $R^{3N+1} \times R^{3N+1} \times R^N \times R^N \times R^N \times R^N$  into  $R^N$ .  $\mathcal{Q}$  maps  $R^{3N+1} \times R^N \times R^N \times R^N$  into  $R^{3N+1}$ .

**Table 2**  
Implementation of algorithms.

Algorithm	Approx. policies	Functional form	# Basis functions	# Nodes
PER1	$\log k', \log c, \log i, \log \lambda, \log l$	1st-order compl. poly.	$1+x$	–
PER2	$\log k', \log c, \log i, \log \lambda, \log l$	2nd-order compl. poly.	$1+x+\frac{x(x+1)}{2}$	–
MARGAL	$k', C^*, \log \lambda^{**}$	2nd-order compl. Chebyshev poly.	$1+x+\frac{x(x+1)}{2}$	$2x^2+1$
SMOL	$k', c^1$	Subset of 4th-order compl. Chebyshev poly.	$1+4x+4\frac{x(x+1)}{2}$	$1+4x+4\frac{x(x+1)}{2}$
CGA	$k'$	2nd-order compl. poly.	$1+x+\frac{x(x+1)}{2}$	500
SSA	$k'$	1st-order compl. poly.	$1+x$	10,000

Notes: For each algorithm listed in the first column, the second column shows the variables for which the algorithm specifies a parameterized policy function ( $k'$  is end-of-period capital;  $C$  is world consumption;  $c^1$  is country-1 consumption; other variables are defined in the main text); the third column describes the functional form of the parameterized policy functions; the final two columns list the number of basis functions embedded in the functional form and (for the global methods) the number of nodes (grid-points) at which the models' equilibrium conditions are evaluated;  $x$  denotes the number of state variables.

\* Only Model II.

\*\* Only Models III–IV.

(which summarizes some key features of the methods). PER1 and SSA use linear functions, while the other methods use higher-, usually second-, order polynomials.

The methods differ fundamentally regarding the construction of the approximate policy functions ( $\hat{G}$ ) of  $c_t, i_t, l_t, \lambda_t$ , (see Table 2). PER1 and PER2 specify all policy functions ( $\hat{G}$ ) as polynomials of  $k_t$  and  $a_t$ . By contrast, CGA and SSA do not explicitly parameterize  $\hat{G}$ ; instead, these methods solve for  $\hat{G}(k_t, a_t)$  from  $\mathcal{Q}(\hat{G}(k_t, a_t), \hat{F}(k_t, a_t), k_t, a_t) = 0$ , taking  $\hat{F}$  and  $(k_t, a_t)$  as given. SMOL and MARGAL take intermediate approaches, by explicitly parameterizing a subset (usually one) of the control variables as functions of  $(k_t, a_t)$  and generating the remaining policy functions as solutions to a subset of the “intra-temporal” conditions, taking the explicitly parameterized policy functions and the state variables as given.

We show later that the different ways of constructing  $\hat{G}$  matter greatly for accuracy, but first we describe how the different methods solve for the coefficients of their parameterized policy functions.

- The *perturbation* methods (PER1 and PER2) of Kollmann et al. (this issue) compute the coefficients of the approximating policy functions as functions of derivatives of the model's equations, evaluated at the (deterministic) steady state. A second-order perturbation requires computing the first- and second-order derivatives of the functions  $\mathcal{H}$  and  $\mathcal{Q}$ .
- The *nominal-rule Galerkin* approach (MARGAL) of Pichler (this issue) introduces polynomial approximations to the unknown policy functions into the equilibrium conditions, computes weighted sums of residuals by projecting these conditions – Eqs. (5) and (6) – on the basis functions of the approximating polynomials, and finally determines the polynomial coefficients by equating each weighted sum to zero (using a Newton-type solver). Unlike the standard Galerkin method, MARGAL uses a non-product monomial rule for computing weighted residuals, which requires evaluating (5) and (6) only on a relatively small grid of points in the state space (see Table 2).
- The *Smolyak-collocation* method (SMOL) of Malin et al. (this issue) is a *sparse-grid* collocation method. It consists of Smolyak's (1963) algorithm for constructing a sparse grid of collocation points in the state space and an accompanying interpolation formula to determine the coefficients of the approximating functions, given the function values at the collocation points. The values at the collocation points are solved for (using “time iteration”<sup>6</sup>) by requiring the model's equilibrium conditions to hold exactly at these points.
- The *cluster-grid algorithm* (CGA) of Maliar et al. (this issue) operates on a set of cluster points constructed to cover the ergodic distribution of state variables.<sup>7</sup> It solves for the parameterized capital policy functions by using “fixed-point iteration”<sup>8</sup> to determine the capital values at each cluster point (given the previous iteration of the policy functions) and then updates the coefficients of the policy functions by running a least-squares regression of the capital values on the parameterized functions.

<sup>6</sup> As described in Judd (1998, pp. 553–555), time iteration mimics value function iteration for Euler-based methods and has good convergence properties due to the monotonicity properties of the Euler equation.

<sup>7</sup> For the results described in the present paper, the ergodic set is constructed using the SSA solution. But, this does not mean that CGA must rely on other solution methods. Indeed, one could start with an arbitrary set of points, use the CGA solution on this set of points to simulate the model and construct an ergodic set, and then re-solve the model on this ergodic set.

<sup>8</sup> As described in Judd (1998, pp. 555–557), fixed-point iteration is simpler than time-iteration to implement (because it does not require solving nonlinear equations), however, it is less numerically stable.

- The *stochastic simulation algorithm* (SSA) of Maliar et al. (this issue) solves for the ergodic distribution of the state variables and generates policy functions using an iterative approach. It proceeds as follows: initialize a policy function for capital, simulate a time series of all model variables, run a least-square regression of the capital decision variables on polynomial terms of the state variables to update the initial guess, and iterate until convergence.<sup>9</sup>

### 2.3. Dealing with high-dimensional state spaces

The models considered are computationally challenging for several reasons: (i) due to the large number of state variables, the models require the computation of high-dimensional integrals (conditional expectations); (ii) the number of coefficients of the parameterized policy functions is large (and grows with the dimension of the state space); (iii) in addition, a large number of “intra-temporal” conditions has to be solved to determine the control variables  $z_t$  at many different points in the state space. As a result, the computational cost may grow rapidly with the dimension of the state-space, making the problem infeasible for large  $N$ . One of the goals of this comparison project is to evaluate different strategies for addressing these challenges. The methods’ choices of different strategies will explain (as discussed below) why they differ noticeably in terms of speed and accuracy.

#### 2.3.1. Computing conditional expectations

Computing the conditional expectation  $\mathbb{E}_t \mathcal{H}$  is costly when the number of shocks is large. In fact, projection algorithms discussed in the prior literature do this using standard tensor-product Gaussian quadrature methods, which require a number of nodes that grows exponentially in  $N$ , and thus become infeasible for large  $N$ . To overcome this problem, the three projection methods here use monomial integration formulae (Stroud, 1971; Judd, 1998) that require fewer nodes: MRGAL and SMOL use integration formulae where the number of points increases quadratically in  $N$ , while CGA first computes policy rules using a cheap formula with  $2N$  points and then refines the solution using a more accurate formula with  $2^N$  points. The stochastic-simulation algorithm (SSA) uses a Monte-Carlo integration method with the same number of points in all models; thus, the cost of that Monte-Carlo approach does not increase with  $N$  (although accuracy may eventually decrease as  $N$  gets large). (Note that, in contrast to the other algorithms, perturbation methods do not require numerical evaluation of conditional expectations; in a second order perturbation, the intercept of the approximating policy functions depends on the second moments of the forcing variables that are exogenously given.)

#### 2.3.2. Solving for $\hat{F}$

All methods we examine approximate the policy functions for capital holdings,  $\hat{F}$ , by a polynomial. PER1 and SSA employ a first-order complete polynomial; PER2, MRGAL and CGA resort to a second-order complete polynomial; SMOL uses a polynomial approximation based on a *subset* of the monomials that form a fourth-order complete polynomial. To compute the coefficients in these approximating polynomials, PER1 and PER2 solve for the Taylor series expansion around a model’s non-stochastic steady state, while SSA, MRGAL, CGA, and SMOL use their specific weighted residual approaches.

As documented in Table 2, for all methods considered, the number of basis functions in the approximating polynomials (and thus the number of coefficients to be determined when solving the model) grows linearly or quadratically in  $N$ . Importantly, the use of complete polynomials allows the number of coefficients to not grow exponentially in  $N$ .

However, this property alone does not guarantee that projection methods are feasible for high-dimensional applications. In addition, the number of nodes (grid-points) at which the residuals in equilibrium conditions are evaluated must not grow too rapidly with  $N$ . Projection methods, along the lines of Judd (1992), have mostly focused on very accurately solving low-dimensional problems and have thus used tensor-product grids, which are both easy to implement and achieve high accuracy. The number of nodes in such grids, however, grows exponentially in the number of state variables, and as a consequence, standard projection approaches would not be feasible for the high-dimensional models considered in the present project.<sup>10</sup> The projection methods we examine, MRGAL, SMOL, and CGA, employ different strategies to overcome this difficulty. MRGAL uses a non-product monomial integration rule, with the number of nodes growing quadratically in  $N$ , to evaluate projections (that is, to integrate weighted residuals); the Smolyak algorithm carefully selects a sparse grid of collocation points (as many points as there are coefficients in the approximating function); CGA chooses a grid of cluster-points located on the ergodic set (thus reducing the grid by eliminating combinations of the state variables that do not occur in equilibrium) and does not increase the number of points with the dimension of the state space.

Finally, note that SSA also evaluates the policy functions (i.e., equilibrium conditions) on a “grid” of points that cover the ergodic set (i.e., the points visited along the simulation). Since the length of the simulation is fixed, the number of points does not increase with the dimension of the state space.

<sup>9</sup> This algorithm can be viewed as a simulation-based version of the parameterized expectations algorithm by Den Haan and Marcet (1990) in which the approximation step is modified as described in Judd et al. (2009) to enhance numerical stability.

<sup>10</sup> This problem was pointed out by Gaspar and Judd (1997), who also suggested incorporating more efficient multi-dimensional numerical integration techniques into projection approaches but left the implementation of that idea for future research.

### 2.3.3. Solving for $\hat{G}$

All the non-perturbation methods solve for (at least some of) the control-variable policy functions,  $\hat{G}$ , from the “intra-temporal” conditions taking the capital choice as given. Solving these “intra-temporal” conditions with a high degree of accuracy may be costly, however: at each point visited in the state-space, a system of nonlinear equations has to be solved for (the elements of)  $z_t$ .

The methods deal with this challenge in different ways. SMOL explicitly parameterizes the consumption of one country, which allows for a closed-form expression of the other  $\hat{G}$  policy functions for Models I and II and a smaller system of nonlinear equations for Models III and IV. MRGAL builds upon the “intra-temporal” choice approach of Maliar and Maliar (2005). The key idea of this approach is to reduce the computational cost of solving for the control variables by approximating the function  $G$  on a grid of points outside of the iterative circle, and using this approximation to compute the control variables inside the circle without a numerical solver. Within the current project, this idea was first implemented for SSA (and Model I) in an earlier version of Maliar et al. (this issue); their approximation to  $G$  was based on a grid-function and interpolation. MRGAL applies this idea for all Models I–IV, using a high-order Chebyshev polynomial approximation to  $G$  rather than interpolation. Finally, CGA and SSA do not specify parametric functions for the control variables, but instead, solve for them directly as the values that satisfy Eqs. (1) and (2), taking as given the state variables and capital decision rule  $k_{t+1} = \hat{F}(k_t, a_t)$ . In principle, this approach can be implemented using any numerical equation solver, but in practice, the computational cost quickly becomes prohibitive. To keep the cost low, CGA and SSA use a vectorized “iteration-on-allocation” solver that computes the control variables simultaneously for an entire simulated series.<sup>11</sup>

## 3. Computational speed

Table 3 reports running times for each method to solve each of the 30 models. These times provide a rough idea of the computational cost of the different approaches. A precise comparison is made difficult by differences in the hardware and software used by the contributors. Nevertheless, several lessons can be learned from Table 3.

First, perturbation methods are faster than global methods by several orders of magnitude. A linear solution can be obtained in less than a second for all models, while a quadratic perturbation solution can be obtained in less than 10 s. Global methods are substantially slower, especially in asymmetric model specifications and when the number of countries is large.

Looking more closely at the global methods, MRGAL and SMOL are faster than SSA for small  $N$ , but the cost of MRGAL and SMOL grows more quickly with  $N$ . For example, MRGAL and SMOL solve Model I with  $N=2$  countries in less than a second, but when  $N=10$ , take over 1 h and 10 h, respectively. By contrast, the time cost of SSA increases less than proportionally with  $N$ , and thus SSA is faster than MRGAL and SMOL when  $N$  is large enough (as can be seen, for example, from Model I with  $N=10$ ). These patterns can be explained by noting that SSA considers a grid of 10,000 points regardless of the size of  $N$ , while the number of grid points at which MRGAL and SMOL must evaluate the models’ equilibrium conditions grows quadratically in  $N$ . Furthermore, the number of coefficients of the (linear) SSA capital policy function does not grow as quickly in  $N$  as the number of coefficients under the higher-order policy functions used by MRGAL and SMOL.

Of the three projection methods, MRGAL is fastest, followed by SMOL and CGA. For small  $N$ , the relative difference between CGA and the other projection methods is particularly large, which most likely reflects the choice of grid points. CGA uses 500 grid points regardless of the value of  $N$ , while for  $N=2$ , MRGAL and SMOL use 33 and 57 points, respectively. As  $N$  increases, the running times of all methods increase substantially, but for different reasons. For MRGAL and SMOL, the number of grid points and cost of evaluating integrals increases quadratically in  $N$ . For CGA, the number of grid points does not increase, but the cost of the monomial integration formula used by CGA increases exponentially with  $N$ .<sup>12</sup>

## 4. Accuracy results

We next examine the accuracy of the solutions produced by each method. The accuracy measures are based on errors in the equilibrium conditions (i.e., Eqs. (5) and (6)). Specifically, we use the definitions/normalizations of errors presented by Juillard and Villemot (this issue); all methods are evaluated using the “test bench” constructed by these authors. The test bench provides a uniform framework for computing errors (i.e., all methods use the exact same exogenous shock processes, error specifications, etc.), and this allows for precise comparison of the methods. We first discuss maximum errors across all equilibrium conditions, for points in the state space that are at a fixed radius from the steady state, and then for points visited along a simulation run. The following discussion focuses on accuracy measures for the 15 asymmetric model specifications (i.e., in which preferences and/or technologies differ across countries)—see Tables 4 and 5. The results for the symmetric models are relegated to a web appendix available for download at the journal’s web site <http://www.sciencedirect.com/science/journal/01651889>, as these results differ little from the asymmetric models.

<sup>11</sup> See Maliar et al. (this issue) for an explanation of the “iteration-on-allocation” solver.

<sup>12</sup> Maliar et al. (this issue) experiment with different monomial formulae to evaluate the trade-off between accuracy and computational cost.

**Table 3**  
Running times (in seconds).

Model	N	PER1	SSA	PER2	MRGAL	CGA	SMOL
<i>Symmetric specifications</i>							
I	2	0.3	149.6	0.5	0.9	117.7	0.9
	4	0.1	210.8	0.5	7.0	263.7	58.7
	6	0.1	271.8	1.4	76.1	639.3	916.3
	8	0.2	340.4	3.4	666.9	5670.4	7312.7
	10	0.4	411.4	6.8	4472.3	157952.4	38149.7
II	2	0.1	891.3	0.3	1.2	1392.7	2.7
	4	0.1	1730.8	0.5	15.4	2752.6	128.6
	6	0.2	2607.0	1.8	178.1	5263.8	1772.4
	8	0.2	3573.0	4.3	3102.8	16378.4	13051.6
III	2	0.1	1743.9	0.2	9.5	1648.1	4.9
	4	0.1	3640.5	0.7	107.3	4099.2	195.0
	6	0.1	5279.3	2.3	1228.2	13588.6	2835.1
IV	2	0.1	2557.7	0.3	10.3	2438.2	7.6
	4	0.1	5767.1	1.3	96.3	6806.4	285.2
	6	0.1	8652.5	4.4	1431.2	41761.4	3890.6
<i>Asymmetric specifications</i>							
I	2	0.1	214.0	0.2	0.9	99.4	0.9
	4	0.1	277.3	0.5	8.5	223.4	71.8
	6	0.1	397.6	1.7	127.4	599.9	1199.5
	8	0.2	516.8	4.0	1059.6	4143.9	9090.3
	10	0.3	601.3	7.7	5499.7	123950.4	45900.5
II	2	0.1	1118.1	0.2	1.4	1602.1	2.2
	4	0.1	2248.0	0.5	16.0	3293.4	117.1
	6	0.1	3360.4	1.8	197.3	7155.7	1674.9
	8	0.2	4833.4	4.3	2378.8	27785.8	12126.8
III	2	0.1	1841.6	0.2	9.1	1985.8	74.6
	4	0.1	3318.2	0.7	107.5	3843.4	2207.2
	6	0.1	4781.7	2.3	1243.5	10332.2	19194.7
IV	2	0.1	3144.2	0.3	10.4	2581.7	134.5
	4	0.2	7267.9	1.4	100.3	6804.8	3391.3
	6	0.1	10679.1	4.4	1489.0	18321.8	28184.4

Notes: The running times for the global solution methods depend to some extent on the initial guesses for the coefficients of policy functions. SSA uses constant policy functions (steady state values) as initial guesses. MRGAL and SMOL use log-linear perturbation solutions as initial guesses, while CGA uses the solutions of SSA as initial guesses. "Symmetric" model specifications assume that preferences and technologies are identical across countries. In "asymmetric" specifications, preferences/technologies differ across countries. The following soft- and hardware was employed: MATLAB 7 on an Intel (R) Pentium(R) 4 CPU, 3.06 GHz, 960 MB RAM (PER1 and PER2); MATLAB 7 on an Intel(R) Core(TM) 2 CPU, 3 GHz, 4 GB RAM (MRGAL); Intel Fortran 8.0 on an Intel (R) Xeon (TM) 2.80 GHz CPU, 12 GB RAM (SMOL); MATLAB 7 on a Quad Intel(R) Core(TM) i7 CPU920, 2.67 GHz, 6 GB RAM (SSA and CGA).

#### 4.1. Accuracy on a sphere

The first accuracy test involves randomly drawing 1000 vectors of states  $(k_t, a_t)$  lying on spheres of fixed distance from the steady state. We consider spheres with radii of  $r_1=0.01$ ,  $r_2=0.1$ , and  $r_3=0.3$ , respectively (see Juillard and Villemot, [this issue](#)). For each model, the error measures presented in [Table 4](#) correspond to the maximum absolute errors across all equilibrium conditions and all 1000 draws. The errors are reported in log10 form.

Close to the steady state ( $r_1=0.01$ ), all solution methods are highly accurate: with the exception of PER1, maximum errors (across all models) are typically well below 0.01%; and even for PER1, maximum errors are below 0.1%. The most accurate method is CGA, with maximum errors never exceeding 0.00020% at radius  $r_1$ , followed by SMOL, PER2 and MRGAL, whose maximum errors are 0.0035%, 0.0062% and 0.01%, respectively. The ranking of the latter three methods depends on the model under consideration; for example, MRGAL dominates PER2 and SMOL for Model I but is dominated by PER2 for Models III and IV. Finally, notice that all "second-order" methods (PER2, MRGAL, SMOL, CGA) dominate the "first-order" methods (PER1, SSA) in terms of accuracy, at radius  $r_1$ .<sup>13</sup>

When one moves further away from the steady state, accuracy deteriorates quickly for the perturbation methods. At radius  $r_2=0.1$ , the error measures associated with PER1 and PER2 are, respectively, two and three orders of magnitude larger than at radius  $r_1=0.01$ ; at  $r_3=0.3$ , they are three and four orders of magnitude larger. The maximum error associated with PER1 is 12% at  $r_2$  and 65% at  $r_3$  (corresponding maximum errors for PER2: 1.90% and 50%).

<sup>13</sup> Note that only PER1 and PER2 are true first- and second-order approximations, respectively. As described in Section 2, the other methods use first- or second-order polynomials to approximate capital but solve for (at least some) non-predetermined variables (consumption, labor, etc.) in a more flexible way.

**Table 4**  
Accuracy on a sphere: asymmetric specifications.

Model	N	PER1	SSA	PER2	MARGAL	CGA	SMOL
<i>Radius r=0.01</i>							
I	2	-3.21	-4.53	-5.61	-6.35	-6.12	-5.06
	4	-3.52	-4.16	-5.24	-6.21	-6.47	-4.82
	6	-3.72	-4.28	-5.40	-6.09	-6.35	-4.66
	8	-3.81	-4.19	-5.58	-5.99	-6.33	-4.55
	10	-3.97	-4.10	-5.70	-5.89	-6.30	-4.45
II	2	-3.04	-4.21	-5.01	-5.00	-6.02	-5.00
	4	-3.26	-4.00	-4.92	-5.11	-6.26	-4.78
	6	-3.50	-4.12	-5.10	-4.95	-6.13	-4.63
	8	-3.65	-4.02	-5.35	-4.98	-6.03	-4.52
III	2	-2.85	-4.22	-4.21	-4.00	-5.79	-4.99
	4	-3.14	-4.06	-4.35	-4.23	-5.90	-4.77
	6	-3.36	-4.20	-4.61	-4.43	-6.01	-4.62
IV	2	-2.94	-4.09	-4.88	-4.14	-5.61	-5.06
	4	-3.25	-3.92	-4.84	-4.50	-5.71	-4.82
	6	-3.46	-3.99	-5.06	-4.25	-5.74	-4.66
<i>Radius r=0.10</i>							
I	2	-1.23	-3.01	-2.63	-4.56	-4.39	-4.62
	4	-1.51	-3.03	-2.22	-4.82	-4.58	-4.68
	6	-1.72	-3.14	-2.44	-5.02	-4.85	-4.60
	8	-1.82	-3.14	-2.53	-5.36	-4.92	-4.51
	10	-1.96	-3.18	-2.73	-5.42	-4.81	-4.44
II	2	-1.07	-2.97	-2.29	-4.19	-4.28	-4.40
	4	-1.27	-3.03	-1.93	-4.40	-4.58	-4.53
	6	-1.48	-3.14	-2.22	-4.43	-4.88	-4.52
	8	-1.65	-3.15	-2.41	-4.52	-4.96	-4.45
III	2	-0.92	-2.92	-1.96	-3.35	-4.29	-4.40
	4	-1.15	-3.03	-1.72	-3.57	-4.50	-4.44
	6	-1.35	-3.15	-2.02	-3.69	-4.84	-4.47
IV	2	-0.98	-2.76	-2.15	-3.49	-4.02	-4.43
	4	-1.23	-2.85	-1.83	-3.70	-4.32	-4.52
	6	-1.46	-2.95	-2.13	-3.72	-4.55	-4.52
<i>Radius r=0.30</i>							
I	2	-0.40	-2.05	-1.01	-2.94	-2.91	-3.38
	4	-0.57	-2.20	-0.77	-3.22	-3.17	-3.85
	6	-0.75	-2.44	-0.99	-3.43	-3.44	-4.31
	8	-0.87	-2.47	-1.12	-3.71	-3.66	-4.34
	10	-0.99	-2.54	-1.33	-3.72	-3.54	-4.31
II	2	-0.29	-2.03	-0.68	-2.50	-2.82	-3.29
	4	-0.41	-2.20	-0.53	-2.91	-3.12	-3.73
	6	-0.54	-2.43	-0.79	-3.23	-3.43	-3.86
	8	-0.69	-2.41	-1.02	-3.57	-3.71	-4.07
III	2	-0.19	-1.97	-0.34	-2.56	-2.87	-3.20
	4	-0.30	-2.16	-0.30	-3.07	-3.19	-3.67
	6	-0.43	-2.38	-0.58	-3.34	-3.46	-4.00
IV	2	-0.23	-1.80	-0.52	-2.63	-2.63	-3.31
	4	-0.36	-1.99	-0.41	-3.18	-3.18	-3.79
	6	-0.53	-2.21	-0.69	-3.36	-3.27	-4.06

Notes: The numbers reported, in log10 units, are maximum absolute errors across all equilibrium conditions and 1000 draws of state variables located on spheres in the state space (centered at steady state) with radii 0.01, 0.10, and 0.30, respectively. This table pertains to "asymmetric" model specifications (in which preference/technology parameters differ across countries). See Juillard and Villemot (this issue) for definition of model errors.

As expected, the accuracy of global methods does not deteriorate as quickly when moving away from the steady state. This is particularly true for the SMOL algorithm, for which the error measures at radius  $r_3=0.3$  are on average only one order of magnitude larger than at radius  $r_1=0.01$ . The maximum error associated with SMOL at radius  $r_3$  never exceeds 0.07%, while the maximum errors at radius  $r_3$  associated with CGA, MARGAL, and SSA are close to 0.23%, 0.32% and 1.58%, respectively. Notice also that, while CGA dominates SMOL at radius  $r_1$ , the reverse is true at radius  $r_3$ .

#### 4.2. Accuracy on a stochastic simulation

We next examine accuracy at values of the state variables visited along a stochastic simulation of 10,000 periods. To give a sense of how this exercise compares with the previous one, note that, for Model IV, the average distance from steady state ranges between 0.11 ( $N=2$ ) and 0.18 ( $N=6$ ). Table 5 reports average and maximum absolute errors across all



**Table 5**  
Accuracy on a simulation: asymmetric specifications.

Model	N	PER1	SSA	PER2	MRGAL	CGA	SMOL
<i>Average errors</i>							
I	2	-3.69	-4.79	-5.13	-6.33	-6.39	-5.80
	4	-3.87	-4.63	-5.19	-6.25	-6.44	-5.86
	6	-3.99	-4.65	-5.29	-6.28	-6.48	-5.89
	8	-4.06	-4.60	-5.35	-6.17	-6.41	-5.85
	10	-4.12	-4.54	-5.39	-6.11	-6.38	-5.83
II	2	-3.53	-4.79	-4.84	-5.69	-6.38	-5.78
	4	-3.67	-4.69	-4.82	-5.92	-6.42	-5.93
	6	-3.78	-4.71	-4.92	-6.05	-6.45	-5.99
	8	-3.86	-4.65	-4.99	-6.09	-6.40	-5.96
III	2	-3.05	-4.80	-4.21	-4.78	-6.15	-5.70
	4	-3.13	-4.69	-4.21	-5.14	-6.19	-5.89
	6	-3.18	-4.72	-4.26	-5.24	-6.22	-5.99
IV	2	-3.11	-4.62	-4.35	-4.85	-5.98	-5.77
	4	-3.16	-4.50	-4.31	-5.18	-6.03	-5.91
	6	-3.20	-4.52	-4.33	-5.28	-6.09	-5.97
<i>Maximum errors</i>							
I	2	-1.70	-3.20	-2.60	-4.49	-4.53	-3.81
	4	-1.84	-3.18	-2.89	-4.57	-4.51	-3.72
	6	-1.95	-3.08	-2.84	-4.60	-4.61	-3.92
	8	-2.00	-3.20	-2.91	-4.57	-4.69	-3.89
	10	-2.00	-3.21	-2.89	-4.44	-4.68	-3.74
II	2	-1.45	-3.12	-2.30	-3.50	-4.50	-3.64
	4	-1.44	-3.17	-2.32	-3.80	-4.48	-3.63
	6	-1.55	-3.05	-2.16	-3.85	-4.39	-3.83
	8	-1.67	-3.14	-2.38	-3.91	-4.54	-3.77
III	2	-1.20	-3.08	-1.90	-2.95	-4.19	-3.60
	4	-1.25	-3.14	-2.06	-3.38	-4.17	-3.53
	6	-1.38	-3.04	-1.87	-3.36	-4.32	-3.66
IV	2	-1.25	-2.90	-2.09	-2.94	-4.07	-3.73
	4	-1.43	-2.96	-2.25	-3.37	-4.06	-3.61
	6	-1.45	-2.84	-2.06	-3.41	-4.11	-3.81

Notes: The numbers reported, in log10 units, are averages (top panel) and maxima (bottom panel) of absolute values of model errors, where the averages/maxima are taken across all equilibrium conditions and all dates for a stochastic simulation run of 10,000 periods. This table pertains to “asymmetric” model specifications (in which preference/technology parameters differ across countries). See [Juillard and Villemot \(this issue\)](#) for definition of model errors.

periods and equilibrium conditions. *Average* errors are below 0.1% for all methods and all models. The largest average errors are associated with PER1; they typically range between 0.01% and 0.1%. SSA produces noticeably smaller errors. In fact, for Models III and IV, the maximum error under SSA is smaller than that in the second-order perturbation solution (PER2). This reflects the fact that although SSA only uses a linear approximation for the capital policy function, it uses a more accurate solution of the “intra-temporal” conditions (see Section 4.3).

The projection methods (MRGAL, CGA and SMOL) all dominate PER1, SSA, and PER2 in terms of accuracy. CGA delivers the lowest average errors, followed by SMOL and MRGAL. Interestingly, MRGAL seems to have a small advantage over SMOL for Models I and II, while producing less accurate approximations for Models III and IV. Finally, for any given method, accuracy tends to improve slightly as the number of countries grows.

The inspection of *maximum* errors confirms the general picture provided by the average errors. PER1 produces the largest maximum errors (up to 6.31%) followed by PER2 (maximum errors up to 1.35%). SSA (maximum errors below 0.15%) performs noticeably better than PER1 and PER2. The maximum CGA, SMOL and MRGAL errors (across all equations) are 0.009%, 0.030%, and 0.115%, respectively. The ranking of projection methods is again dependant on the model, in the same way as for the average error measures.

#### 4.3. Accuracy on individual equations

To better understand the above results, it is useful to separately examine the equilibrium conditions. Due to space constraints, we focus on Models I and IV with  $N=2$  countries (asymmetric specifications). [Table 6](#) displays maximum errors (across periods) for each equation in those two models, for the same stochastic simulation of 10,000 periods considered in [Table 5](#).

[Table 6](#) reveals that *all* solution methods generate non-negligible approximation errors in Euler equations, but not all of them generate such errors in the “intra-temporal” conditions. Noticeably, for SSA and CGA, the Euler equations constitute

**Table 6**  
Accuracy by equation.

Equation	PER1	SSA	PER2	MRGAL	CGA	SMOL
<i>Model I with N=2 (asymmetric specification)</i>						
Euler <sub>1</sub>	-3.89	-3.20	-4.73	-4.53	-4.55	-4.84
Euler <sub>2</sub>	-3.89	-3.22	-4.62	-4.49	-4.53	-4.87
WorResConst	-1.70	-	-2.60	-	-	-3.81
MargUtilCons <sub>1</sub>	-9.28	-	-8.62	-	-	-
MargUtilCons <sub>2</sub>	-9.07	-	-8.64	-	-	-
LawMotionCap <sub>1</sub>	-3.09	-	-3.96	-	-	-
LawMotionCap <sub>2</sub>	-2.91	-	-3.80	-	-	-
<i>Model IV with N=2 (asymmetric specification)</i>						
Euler <sub>1</sub>	-3.64	-2.93	-4.13	-4.27	-4.07	-4.85
Euler <sub>2</sub>	-3.61	-2.90	-4.12	-4.23	-4.09	-4.86
WorResConst	-1.25	-	-2.09	-2.94	-	-3.73
MargUtilCons <sub>1</sub>	-1.79	-	-2.97	-7.77	-	-
MargUtilCons <sub>2</sub>	-2.03	-	-2.92	-6.56	-	-7.85
MargUtilLab <sub>1</sub>	-1.59	-	-2.72	-7.77	-	-
MargUtilLab <sub>2</sub>	-1.63	-	-2.51	-6.56	-	-7.58
LawMotionCap <sub>1</sub>	-2.73	-	-3.29	-	-	-
LawMotionCap <sub>2</sub>	-2.49	-	-3.20	-	-	-

Notes: This table considers Models I and IV with  $N=2$  countries (asymmetric specifications). For each model equation (listed in 1st Column) the table reports maximum absolute errors, in  $\log_{10}$  units, across all periods of a simulation run of 10,000 periods. For better readability, we use an entry “-” if the accuracy measure is below  $-10$  (such that the numerical error is below  $10^{-10}$ ), as we view such errors as negligible. The definition of model errors follows Juillard and Villemot (this issue).

Euler<sub>*i*</sub>: country-*i* Euler equation. WorResConst: world resource constraint. MargUtilCons<sub>*i*</sub>: risk sharing condition that equates the (scaled) marginal utility of consumption in country *i* to the Lagrange multiplier of the world resource constraint. MargUtilLab<sub>*i*</sub>: condition that equates marginal rate of substitution between leisure and consumption to marginal product of labor in country *i*. LawMotionCap<sub>*i*</sub>: law of motion of country *i* capital.

the only source of (non-negligible) errors, which is due to the “iteration-on-allocation” strategy that allows SSA and CGA to solve the “intra-temporal” conditions, Eq. (6), very accurately. On the other hand, PER1 and PER2 display non-negligible errors for all equilibrium conditions (because, as discussed above, PER1 and PER2 parameterize all decision rules). For both MRGAL and SMOL, the emergence of non-negligible errors in the “intra-temporal” conditions differs between the two models. For Model I, MRGAL solves the “intra-temporal” conditions very accurately, while SMOL generates a noticeable error in the world resource constraint.<sup>14</sup> For Model IV, both MRGAL and SMOL generate error measures larger than  $10^{-10}$  in several intra-temporal equations, including the world resource constraint.

Finally, notice that, if a method generates an approximation error in the world resource constraint, then this error is larger than the error in any other equation.<sup>15</sup> Table 6 thus highlights the important role that accurately solving the “intra-temporal” conditions has for driving the maximum errors across all equations (see Table 5).

## 5. Further comparisons

In this section, we compare the solutions along other dimensions that are of interest to economists. We discuss plots of the approximate policy functions and of simulated time series, and we compare unconditional moments of simulated variables. Due to space constraints, this section only reports results for the “asymmetric” specification of Model IV with  $N=2$  countries.

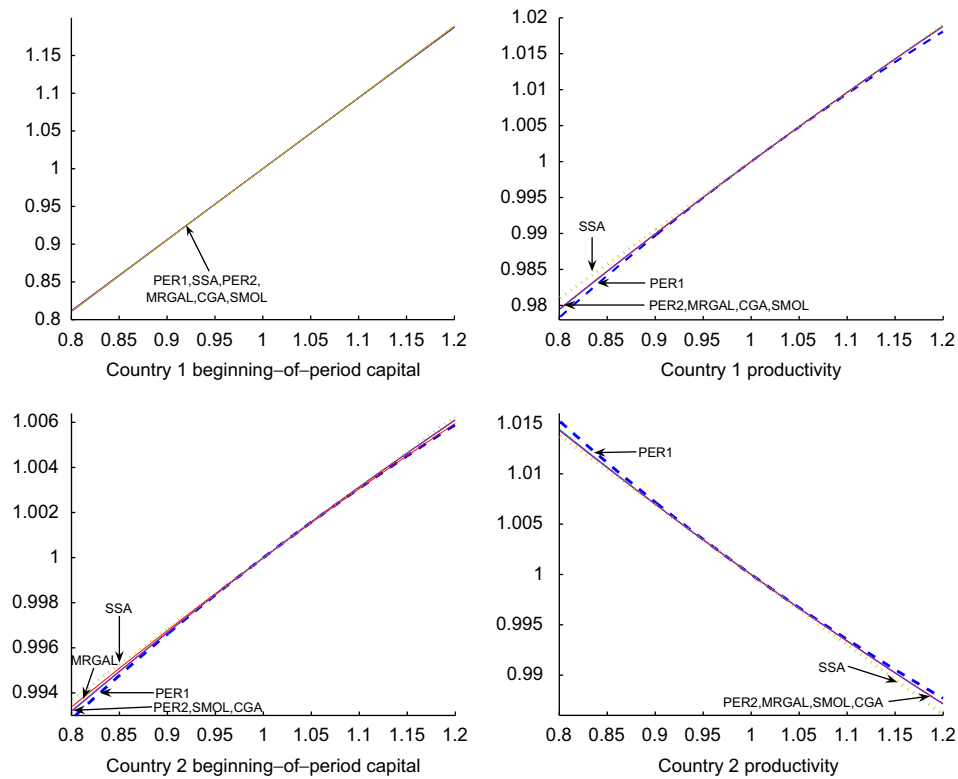
### 5.1. Policy functions

Fig. 1 displays the policy function for country-1 capital,  $k_{t+1}^1$ . One state variable is varied at a time, ranging from 80% to 120% of its steady-state value, while holding the other state variables at their steady-state values. In this range, the policy functions generated by the higher-order approximations PER2, MRGAL, CGA and SMOL are very similar, while those generated by PER1 and SSA differ more noticeably.

Table 7 reports the maximum over absolute relative differences (in percentage points) between the country-1 capital policy functions generated by any pair of solution methods, over the part of the state space considered in Fig. 1

<sup>14</sup> The intuition behind this difference is that, while MRGAL parameterizes only the capital decision rules for Model I, SMOL parameterizes one more decision rule (consumption of country 1) which leads to an approximation error in one more equilibrium condition (the world resource constraint).

<sup>15</sup> The error in the world resource constraint is especially large under PER1, as that constraint is highly non-linear in logged states and controls (PER1 uses a log-linear approximation). By contrast, in Model I, the international risk sharing conditions are linear in logged consumption and the logged Lagrange multiplier of the world resource constraint—which explains why PER1 generates negligible errors in these conditions.



**Fig. 1.** Capital policy function  $k_{t+1}^1(k_t^1, a_t^1, k_t^2, a_t^2)$ .

Notes: For Model IV with  $N=2$  (asymmetric) countries, the figure shows the end-of-period capital stock of country 1,  $k_{t+1}^1$ , as a function of  $k_t^1$  (top-left),  $a_t^1$  (top-right),  $k_t^2$  (bottom-left), and  $a_t^2$  (bottom-right), respectively. In each Panel, one state variable is varied at a time, holding the other state variables constant (at steady-state values).

**Table 7**

Maximum differences for country-1 capital policy function (%).

	PER1	SSA	PER2	MRGAL	CGA
<i><math>k_{t+1}^1</math> choice as a function of <math>k_t^1</math></i>					
SSA	0.148				
PER2	0.150	0.102			
MRGAL	0.141	0.093	0.009		
CGA	0.158	0.110	0.009	0.017	
SMOL	0.152	0.104	0.002	0.011	0.006
<i><math>k_{t+1}^1</math> choice as a function of <math>a_t^1</math></i>					
SSA	0.268				
PER2	0.111	0.158			
MRGAL	0.119	0.150	0.008		
CGA	0.119	0.149	0.008	0.003	
SMOL	0.108	0.160	0.003	0.011	0.011
<i><math>k_{t+1}^1</math> choice as a function of <math>k_t^2</math></i>					
SSA	0.084				
PER2	0.033	0.050			
MRGAL	0.054	0.033	0.020		
CGA	0.034	0.049	0.002	0.019	
SMOL	0.037	0.047	0.004	0.017	0.003
<i><math>k_{t+1}^1</math> choice as a function of <math>a_t^2</math></i>					
SSA	0.0163				
PER2	0.082	0.107			
MRGAL	0.087	0.108	0.005		
CGA	0.094	0.110	0.011	0.007	
SMOL	0.079	0.105	0.003	0.008	0.014

Notes: For Model IV with  $N=2$  countries (asymmetric specification), the table reports maximum absolute values of relative differences between country-1 policy functions for end-of-period capital  $k_{t+1}^1$ , generated by pairs of algorithms. We vary one state variable at a time, in a range from 80% to 120% of its steady-state value, holding the other state variables constant (at steady-state values).  $k_t^i$ ,  $a_t^i$ : beginning-of-period capital and productivity of country  $i$ .

**Table 8**  
Maximum differences across simulated series (%).

Variable	PER1	SSA	PER2	MRGAL	CGA
<i>Consumption of country 1</i>					
SSA	0.81				
PER2	0.55	0.68			
MRGAL	0.54	0.63	0.09		
CGA	0.53	0.65	0.04	0.08	
SMOL	0.55	0.64	0.04	0.09	0.02
<i>Labor supply of country 1</i>					
SSA	1.26				
PER2	1.02	0.66			
MRGAL	1.04	0.61	0.09		
CGA	1.04	0.66	0.07	0.08	
SMOL	1.02	0.64	0.06	0.07	0.06
<i>Investment of country 1</i>					
SSA	9.72				
PER2	7.64	7.05			
MRGAL	7.44	6.45	1.93		
CGA	7.21	6.73	1.72	0.65	
SMOL	7.10	6.47	1.62	0.79	0.46
<i>End-of-period capital of country 1</i>					
SSA	1.93				
PER2	1.27	1.46			
MRGAL	1.34	1.34	0.34		
CGA	1.23	1.38	0.15	0.25	
SMOL	1.27	1.39	0.09	0.37	0.16
<i>Consumption of country 2</i>					
SSA	1.47				
PER2	0.90	1.24			
MRGAL	0.86	1.17	0.12		
CGA	0.88	1.19	0.07	0.11	
SMOL	0.90	1.17	0.06	0.10	0.04
<i>Labor supply of country 2</i>					
SSA	1.79				
PER2	1.90	0.83			
MRGAL	1.73	0.83	0.18		
CGA	1.73	0.83	0.19	0.14	
SMOL	1.73	0.81	0.17	0.15	0.03
<i>Investment of country 2</i>					
SSA	9.56				
PER2	8.56	5.34			
MRGAL	11.27	5.39	3.34		
CGA	10.84	5.32	2.94	0.58	
SMOL	11.00	5.02	3.09	0.62	0.39
<i>End-of-period capital of country 2</i>					
SSA	3.52				
PER2	1.71	3.62			
MRGAL	1.60	3.65	0.26		
CGA	1.64	3.60	0.14	0.24	
SMOL	1.67	3.46	0.16	0.23	0.14

Notes: For Model IV with  $N=2$  countries (asymmetric specification), the table reports maximum absolute values of relative differences between simulated series of listed variables generated by the algorithms (pairwise comparisons). Based on a simulation run of 10,000 periods.

(e.g., the entry 'PER1-SSA' reports the maximum relative % difference between the PER1 policy function and the SSA policy function).<sup>16</sup> The maximum pairwise relative differences between the policy functions generated by PER2, MRGAL, CGA and SMOL do not exceed 0.02%. By contrast, the differences relative to PER1 and SSA are larger and reach up to 0.16%. The largest differences appear between the PER1 and SSA methods (up to 0.27%).<sup>17</sup>

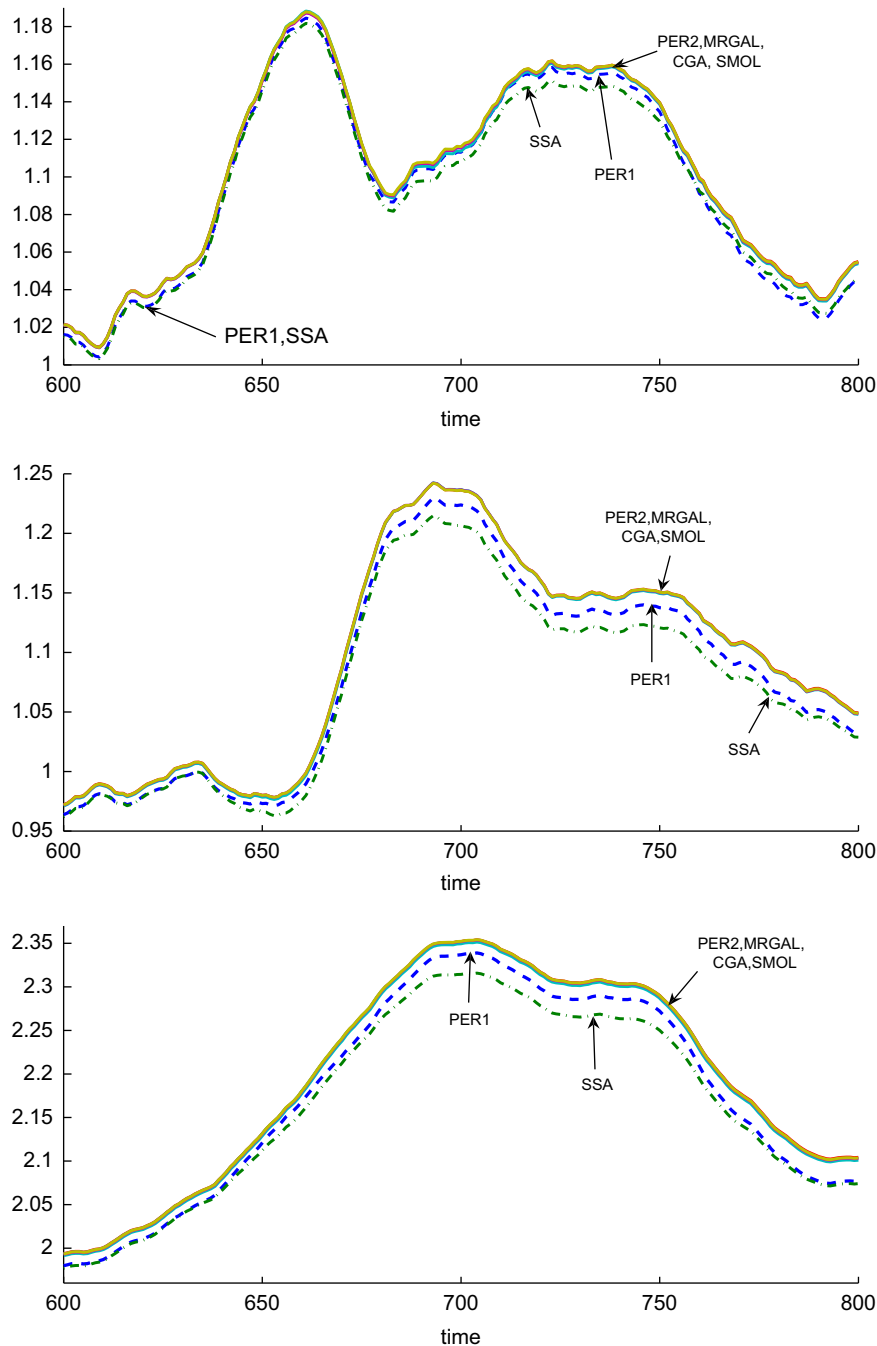
<sup>16</sup> The relative difference between two quantities  $a$  and  $b$  is defined as  $(a-b)/(0.5 \cdot (|a|+|b|))$ .

<sup>17</sup> The differences between the PER1 and SSA policy functions for capital reflect the fact that the PER1 rule is a log-linear approximation, while the SSA rule is based on a linear approximation.

## 5.2. Comparing simulated time series

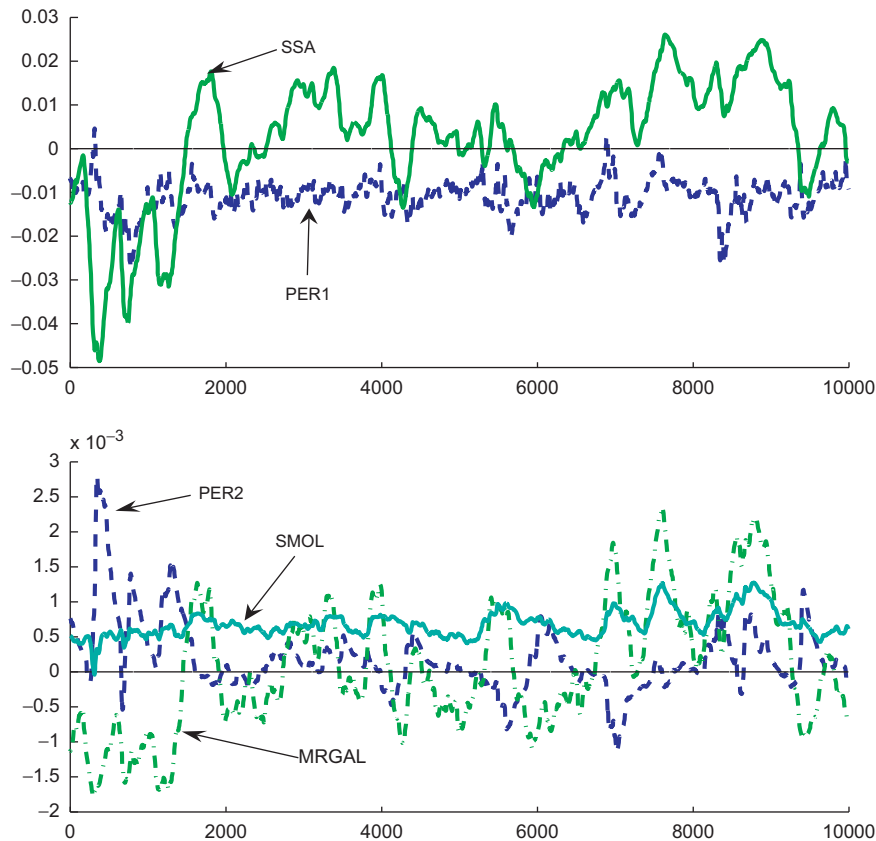
Table 8 compares simulated decision variables (consumption, hours worked, investment and capital) of countries 1 and 2, based on the same 10,000 period simulation described in Section 4. For each pair of solution methods, we report maximum absolute relative differences (between the simulated time series).

The PER2, MRGAL, CGA and SMOL methods generate simulated consumption and hours series that are quite similar (maximum relative deviations below 0.20%); those consumption and hours series differ more noticeably from the ones



**Fig. 2.** Simulated capital stocks.

Notes: For Model IV with  $N=2$  countries (asymmetric specification), the figure shows simulated time series for the capital stock of country 1 (top panel) and of country 2 (middle panel), and for the world capital stock (bottom panel), over a sample of 200 periods. (The sample exhibits the largest model errors in a simulation run of 10,000 periods.)



**Fig. 3.** Differences between simulated world capital stocks.

Notes: For Model IV with  $N=2$  countries (asymmetric specification), the figure shows the difference between simulated world capital stock series generated by SSA, PER1, PER2, MRGAL, SMOL and the series generated by CGA in a simulation run of 10,000 periods.

generated by the PER1 and SSA methods (maximum relative deviations of up to 1.9%). Relative differences across methods are larger for simulated capital, and especially for investment. The maximum relative difference between investment series generated by PER1 and SSA is 9.7%; the investment series generated by PER2 deviate by up to 3.4% from the series generated by MRGAL, CGA and SMOL.

Fig. 2 shows the simulated capital stocks during a 200-period sub-sample ranging from  $t=600$  to 800. We choose this sub-sample because it is the portion of the 10,000-period simulation in which the approximation errors associated with the six numerical solutions attain their largest values. We observe that the simulated series generated by all six methods are highly correlated. In particular, the series generated by the non-linear methods (PER2, MRGAL, CGA, SMOL) are basically indistinguishable. However, there are noticeable differences between the series generated by these four methods and the simulated series generated by the linear methods, PER1 and SSA. Specifically, in the sub-sample considered, the capital stocks generated by PER1 and SSA are always below the capital stocks generated by the higher-order methods.

Fig. 3 displays the differences, for the whole 10,000 period simulation, between the world capital stock series generated by PER1, SSA, PER2, MRGAL and SMOL compared to the series generated by CGA.<sup>18</sup> Note that all differences are stationary (i.e., the simulated series do not diverge). Moreover the differences between the solution methods are relatively small. This contrasts with Den Haan (2010) who finds that one of the solution algorithms<sup>19</sup> generates capital series that are a multiple of the capital series generated by the other algorithms. However, Fig. 3 does show that there are persistent differences between solutions. The top panel of Fig. 3 shows that the world capital stock generated by PER1 is systematically (i.e., in 99.5% of the observations) below the capital stock generated by CGA. This might reflect the certainty equivalence property of the PER1 solution that rules out precautionary savings. The capital stock generated by SSA, on the other hand, is smaller than CGA in certain sub-samples (e.g., the sub-sample illustrated in Fig. 2) but larger in others. The bottom panel of Fig. 3

<sup>18</sup> CGA is chosen as the reference method because it is the most accurate method on the stochastic simulation, as indicated by Table 5.

<sup>19</sup> Namely the one that is based on first-order perturbation methods and replaces the inequality constraint with a penalty function.

**Table 9**  
Unconditional moments of key variables.

	$c_1$	$c_2$	$l_1$	$l_2$	$i_1$	$i_2$	$y_1$	$y_2$
<i>PER1</i>								
mean	0.02796	0.02792	0.99966	1.00258	0.02497	0.02494	0.02798	0.02803
std	0.00077	0.00113	0.02639	0.03456	0.00296	0.00335	0.00212	0.00216
$corr(x, x_{-1})$	0.99	0.99	0.97	0.96	0.91	0.93	0.97	0.96
$corr(x, y_1)$	0.81	0.53	0.94	-0.16	0.75	0.12	1.00	0.21
$corr(x, y_2)$	0.43	0.67	0.00	0.73	0.00	0.78	0.21	1.00
<i>SSA</i>								
mean	0.02803	0.02804	0.99845	0.99954	0.02496	0.02500	0.02801	0.02806
std	0.00076	0.00111	0.02596	0.03466	0.00290	0.00328	0.00209	0.00216
$corr(x, x_{-1})$	0.99	0.99	0.97	0.96	0.91	0.92	0.97	0.96
$corr(x, y_1)$	0.82	0.54	0.94	-0.19	0.74	0.08	1.00	0.18
$corr(x, y_2)$	0.40	0.66	-0.03	0.75	-0.06	0.79	0.18	1.00
<i>PER2</i>								
mean	0.02803	0.02803	0.99838	0.99971	0.02493	0.02498	0.02800	0.02805
std	0.00077	0.00114	0.02625	0.03442	0.00291	0.00331	0.00211	0.00216
$corr(x, x_{-1})$	0.99	0.99	0.97	0.96	0.91	0.93	0.97	0.96
$corr(x, y_1)$	0.81	0.53	0.93	-0.16	0.75	0.12	1.00	0.21
$corr(x, y_2)$	0.43	0.67	0.00	0.72	0.00	0.78	0.21	1.00
<i>MARGAL</i>								
mean	0.02803	0.02803	0.99839	0.99972	0.02493	0.02497	0.02800	0.02805
std	0.00077	0.00114	0.02635	0.03451	0.00294	0.00334	0.00211	0.00216
$corr(x, x_{-1})$	0.99	0.99	0.97	0.96	0.91	0.93	0.97	0.96
$corr(x, y_1)$	0.81	0.53	0.93	-0.16	0.75	0.11	1.00	0.21
$corr(x, y_2)$	0.43	0.67	-0.01	0.72	-0.01	0.78	0.21	1.00
<i>CGA</i>								
mean	0.02803	0.02803	0.99838	0.99973	0.02493	0.02497	0.02800	0.02805
std	0.00077	0.00114	0.02629	0.03449	0.00293	0.00333	0.00211	0.00216
$corr(x, x_{-1})$	0.99	0.99	0.97	0.96	0.91	0.93	0.97	0.96
$corr(x, y_1)$	0.81	0.53	0.93	-0.16	0.75	0.12	1.00	0.21
$corr(x, y_2)$	0.43	0.67	0.00	0.72	0.00	0.78	0.21	1.00
<i>SMOL</i>								
mean	0.02803	0.02803	0.99839	0.99968	0.02494	0.02498	0.02800	0.02806
std	0.00077	0.00114	0.02629	0.03449	0.00293	0.00333	0.00211	0.00216
$corr(x, x_{-1})$	0.99	0.99	0.97	0.96	0.91	0.93	0.97	0.96
$corr(x, y_1)$	0.81	0.53	0.93	-0.16	0.75	0.12	1.00	0.21
$corr(x, y_2)$	0.43	0.67	0.00	0.72	0.00	0.78	0.21	1.00

Notes: For Model IV with  $N=2$  countries (asymmetric specification), the table shows moments of key variables as listed in the first row. By  $c_i$ ,  $l_i$ ,  $i_i$ ,  $y_i$  we denote consumption, hours worked, investment, and GDP in country  $i=1,2$ . The moments were computed from a simulated time series of 10,000 periods.

shows that the world capital stock generated by SMOL is slightly but systematically (in 99.9% of the observations) above the capital stock generated by CGA, while there are no systematic differences between PER2, MARGAL, and CGA.

### 5.3. Simulated moments

Finally, we compare moments of key variables constructed on the 10,000 period simulation. This is an interesting exercise because such moments constitute a primary output of DSGE models. Indeed, one criterion for assessing such models is how far these statistics are from their empirical counterparts. Therefore, it is important to know whether the choice of approximation method can lead to significant differences in these statistics. For each method, Table 9 reports 40 moments (means, standard deviations, and (auto-)correlations) of consumption, hours worked, investment and output. Table 10 reports average and maximum relative differences of the 40 moments across methods.<sup>20</sup>

CGA and SMOL produce moments that are extremely similar, while the difference between these two methods and PER2 or MARGAL are slightly larger but still only in the third decimal place. SSA has the largest differences with other methods, but from Table 9, we see that even these differences only consist of a few hundredths for some correlations. Even the biggest differences (around 5%) would not matter when comparing model and data moments, because sampling error in empirical data moments is typically much larger.

<sup>20</sup> For correlations, we report simple differences ( $corr_1 - corr_2$ ) because some correlations are close to zero.

**Table 10**  
Average and maximum relative differences in moments (%).

Variable	PER1	SSA	PER2	MRGAL	CGA
<i>Average pairwise relative differences</i>					
SSA	1.12				
PER2	0.20	1.05			
MRGAL	0.23	0.99	0.22		
CGA	0.15	1.07	0.07	0.16	
SMOL	0.14	1.06	0.08	0.15	0.01
<i>Maximum pairwise relative differences</i>					
SSA	5.26				
PER2	1.61	5.41			
MRGAL	1.08	4.24	1.17		
CGA	0.81	5.18	0.81	0.95	
SMOL	0.78	5.14	0.84	0.90	0.05

Notes: For each pair of algorithms, the table reports average and maximum absolute values (in %) of relative differences across 40 moments. The moments are computed from a simulated time series of 10,000 periods. The table pertains to Model IV with  $N=2$  countries (asymmetric specification).

## 6. Conclusion

We conclude by discussing two possible directions for future research. First, our results suggest that combining certain aspects of the solution algorithms might lead to hybrid methods with superior speed and accuracy. Future research might, for example, incorporate Maliar et al.'s (this issue) fast and accurate iteration-on-allocation technique for solving intra-temporal conditions into the Smolyak collocation method. Furthermore, Maliar et al. (this issue) propose a hybrid method which uses perturbation to solve for the state variables (capital stocks) and uses more accurate numerical methods (e.g., iteration-on-allocation) to solve for other variables. They test the hybrid perturbation method using the models of the project and find that it delivers solutions that are far more accurate (more than an order of magnitude) than the ones delivered by the original log-linear decision rules.

Second, as discussed in the Introduction, the key challenges posed by the models studied here is the large number of state variables. However, these models are “easy” in the sense that their policy functions are smooth. An open question is how well the algorithms here would solve models with stronger non-linearities. To address this question, future research might examine the numerical algorithms in an environment with occasionally binding inequality constraints as present, for example, in the comparison conducted by Den Haan (2010).

## Acknowledgements

We are grateful to Wouter Den Haan, Ken Judd, and Michel Juillard for advice and encouragement and thank the other participants of the project for comments and suggestions. We are particularly grateful to Michel Juillard and Sebastien Villemot for providing us with computer code allowing for a uniform comparison of numerical methods; all accuracy measures and simulations reported in this paper were computed with that *test bench*. The views in this paper are solely the responsibility of the authors and should not be interpreted as reflecting the views of the Board of Governors of the Federal Reserve System or of the Oesterreichische Nationalbank or any other person associated with the Federal Reserve System or with the Oesterreichische Nationalbank. R. Kollmann thanks the National Bank of Belgium and the EU Commission for financial support. S. Maliar acknowledges financial support from the Hoover Institution at Stanford University, the Ivie, the Generalitat Valenciana under the grant BEST/2010/141 and the Ministerio de Ciencia e Innovacion de Espana and FEDER funds under the project SEJ-2007-62656. P. Pichler gratefully acknowledges financial support from the Austrian Science Fund (FWF) under project no. P-19686.

## References

- Aruoba, S.B., Fernandez-Villaverde, J., Rubio-Ramirez, J.F., 2006. Comparing solution methods of dynamic equilibrium economies. *Journal of Economic Dynamics and Control* 30, 2477–2508.
- Den Haan, W.J., 2010. Comparison of solutions to the incomplete markets model with aggregate uncertainty. *Journal of Economic Dynamics and Control* 34, 4–27.
- Den Haan, W.J., Judd, K.L., Juillard, M., this issue. Computational Suite of Models with Heterogenous Agents: Multi-Country Real Business Cycle Models. *Journal of Economic Dynamics and Control*, doi:10.1016/j.jedc.2010.09.010.
- Den Haan, W.J., Marcat, A., 1990. Solving the stochastic growth model by parameterized expectations. *Journal of Business and Economic Statistics* 8, 31–34.
- Gaspar, J., Judd, K.L., 1997. Solving large-scale rational-expectations models. *Macroeconomic Dynamics* 1, 45–75.
- Heer, B., Maussner, A., 2008. Computation of business cycle models: a comparison of numerical methods. *Macroeconomic Dynamics* 12, 641–663.
- Judd, K.L., 1992. Projection methods for solving aggregate growth models. *Journal of Economic Theory* 58, 410–452.
- Judd, K.L., 1998. *Numerical Methods in Economics*. MIT Press, Cambridge, MA.



- Judd, K.L., Maliar, L., Maliar, S., 2009. Numerically stable stochastic simulation approaches for solving dynamic models. NBER Working Paper 15296.
- Juillard, M., Villemot, S., this issue. Multi-country Real Business Cycle models: accuracy tests and testing bench. *Journal of Economic Dynamics and Control*, doi:10.1016/j.jedc.2010.09.011.
- Kollmann, R., Kim, J., Kim, S.H., this issue. Solving the multi-country Real Business Cycle model using a perturbation method. *Journal of Economic Dynamics and Control*, doi:10.1016/j.jedc.2010.09.012.
- Maliar, L., Maliar, S., 2005. Parameterized expectations algorithm: how to solve for labor easily. *Computational Economics* 25, 269–274.
- Maliar, S., Maliar, L., Judd, K.L., this issue. Solving the multi-country Real Business Cycle model using ergodic set methods. *Journal of Economic Dynamics and Control*, doi:10.1016/j.jedc.2010.09.014.
- Malin, B.A., Krueger, D., Kubler, F., this issue. Solving the multi-country Real Business Cycle model using a Smolyak collocation method. *Journal of Economic Dynamics and Control*, doi:10.1016/j.jedc.2010.09.015.
- Marimon, R., Scott, A., 1999. *Computational Methods for the Study of Dynamic Economies*. Oxford University Press, Oxford.
- Pichler, P., this issue. Solving the multi-country Real Business Cycle model using a monomial rule Galerkin method. *Journal of Economic Dynamics and Control*, doi:10.1016/j.jedc.2010.09.009.
- Smolyak, S., 1963. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics, Doklady* 4, 240–243.
- Stroud, A., 1971. *Approximate Calculation of Multiple Integrals*. Prentice Hall, Englewood Cliffs, NJ.
- Taylor, J.B., Uhlig, H., 1990. Solving nonlinear stochastic growth models: a comparison of alternative solution methods. *Journal of Business and Economic Statistics* 8, 1–17.