Comparing numerical solutions of models with heterogeneous agents (Model A): a simulation - based parameterized expectations algorithm

Lilia Maliar and Serguei Maliar*†

August 21, 2007

Abstract

In this paper, we describe how to solve Model A (finite number of countries - complete markets) of the JEDC project by using a simulation-based Parameterized Expectations Algorithm (PEA).

JEL classification : C6; C63; C68; C88

Key Words : Nonlinear dynamic models; Heterogeneous agents; Parameterized expectations; Monte Carlo simulation; Numerical solutions

*We thank Kenneth Judd and Michel Juillard for several useful suggestions. This research was supported by the Stanford Institute for Theoretical Economics, the Center for Financial Studies in Frankfurt, the Instituto Valenciano de Investigaciones Económicas and the Ministerio de Ciencia y Tecnología de España under the Ramón y Cajal program and BEC 2001-0535.

†Departamento de Fundamentos del Análisis Económico, Universidad de Alicante, Campus San Vicente del Raspeig, Ap. Correos 99, 03080 Alicante, Spain. E-mails: *maliarl@merlin.fae.ua.es* (Lilia Maliar), *maliars@merlin.fae.ua.es* (Serguei Maliar).

1 Introduction

In this paper, we describe how to solve Model A (finite number of countries - complete markets) of the JEDC project by using a simulation-based Parameterized Expectations Algorithm (PEA).[1] We study only variants of the model in which countries are heterogeneous in their fundamentals A5, A7, A8; the variants where countries are identical in their fundamentals A1, A3, A4 are particular cases of the heterogeneous-country ones and can be solved by using the same program.[2]

The plan of the paper is as follows: Section 2 presents the model and derives the optimality conditions. Section 3 presents the algorithm. Section 4 describes the parameter choice. Section 5 discusses the results, and finally, Section 6 concludes.

2 The model

We consider a model with a finite number of countries, N, where each country is populated by a representative consumer. A social planner maximizes a weighted sum of the expected lifetime utilities of the countries' representative consumers subject to the resource constraint, i.e.,

$$\max_{\left\{\left\{c_t^n,l_t^n,k_{t+1}^n\right\}_{n=1}^N\right\}_{t=0}^\infty} E_0 \sum_{n=1}^N \tau^n \left(\sum_{t=0}^\infty \beta^t u^n \left(c_t^n, l_t^n\right) \right) \tag{1}$$

subject to

$$\sum_{n=1}^N c_t^n + \sum_{n=1}^N k_{t+1}^n = (1-\delta) \sum_{n=1}^N k_t^n + \sum_{n=1}^N a_t^n A f^n \left(k_t^n, l_t^n\right) - \frac{\varphi}{2} \sum_{n=1}^N k_t^n \left(\frac{k_{t+1}^n}{k_t^n} - 1 \right)^\xi, \tag{2}$$

where E_t is the operator of conditional expectation; c_t^n, k_t^n, a_t^n, u^n, f^n and τ^n are consumption, capital, technology shock, utility function, production

[1] The computation of equilibrium in some of the studied models can be simplified by using aggregation theory, see Maliar and Maliar (2003a) for a discussion of aggregation results for dynamic models. In the present paper, we intentionally make no use of either aggregation theory or any other simplifying analytical results.

[2] Models A1, A3 and A4, in which countries are identical in their fundamentals, can be solved in a more efficient manner by exploiting their symmetric structure. In subsequent version, we will write separate programs for those models.

function and welfare weight of a country $n \in \{1, ..., N\}$, respectively; β is the discount factor; δ is the depreciation rate; A is the level of technology; and φ and ξ are the adjustment cost parameters. Initial condition $\left(\{k_0^n\}_{n=1}^N, \{a_0^n\}_{n=1}^N\right)$ is given. The process for technology shocks in country n is given by

$$\log a_t^n = \rho \log a_{t-1}^n + \sigma \left(e_t + e_t^n\right), \qquad \text{with } e_t, e_t^n \sim N(0,1), \tag{3}$$

where ρ and σ are the autocorrelation coefficient and the standard deviation of technology shocks, respectively.

We restrict attention to an interior first-order recursive (Markov) equilibrium. If such an equilibrium exists, it satisfies First Order Conditions (FOCs) of the form

$$u_1^n \left(c_t^n, l_t^n\right) \tau^n = u_1^m \left(c_t^m, l_t^m\right) \tau^m, \tag{4}$$

$$u_2^n \left(c_t^n, l_t^n\right) = -u_1^n \left(c_t^n, l_t^n\right) a_t^n A f_2^n \left(k_t^n, l_t^n\right), \tag{5}$$

$$u_1^n \left(c_t^n, l_t^n\right) \omega_t^n = \beta E_t \left\{ u_1^n \left(c_{t+1}^n, l_{t+1}^n\right) \left[\theta_{t+1}^n + a_{t+1}^n A f_1^n \left(k_{t+1}^n, l_{t+1}^n\right)\right] \right\}, \tag{6}$$

where $n, m \in \{1, ..., N\}$ and where ω_t^n and θ_t^n are defined as

$$\omega_t^n \equiv 1 + \frac{\varphi \xi}{2} \left(\frac{k_{t+1}^n}{k_t^n} - 1\right)^{\xi - 1},$$

$$\theta_t^n \equiv 1 - \delta - \frac{\varphi}{2} \left(\frac{k_{t+1}^n}{k_t^n} - 1\right)^{\xi - 1} \left(\frac{k_{t+1}^n}{k_t^n} (1 - \xi) - 1\right).$$

Here, and further on in the text, notation of type h_j stands for the first order partial derivative of a function $h(x_1, ..., x_j, ..., x_J)$ with respect to a variable x_j. FOCs (4), (5), (6) and resource constraint (2) determine the equilibrium uniquely.

3 A simulation-based PEA

To solve the model, we use a version of the simulation-based Parameterized Expectations Algorithm (PEA) by den Haan and Marcet (1990).[3] Under the

[3] An extensive discussion of this version of the PEA and a list of its applications can be found in Marcet and Lorenzoni (1999). For a general discussion of the projection methods, see Judd (1992, 1998).

PEA, one should approximate the conditional expectation in (6) by a parametric function of state variables and search for approximate decision rules by simulation. If $N > 1$, we cannot use the parameterization of the expectation in the Euler equations (6) proposed in den Haan and Marcet's (1990) because we would get N equations identifying linearly-dependent marginal utilities of consumption and we would be left with only one restriction, resource constraint (2), to determine N capital stocks. In other words, consumption and labor would be overidentified and the capital stocks would be underidentified (see Marcet and Lorenzoni, 1999, for a discussion of this issue). To deal with this problem, we re-write the Euler equation (6) as follows:

$$k_{t+1}^n = \frac{\beta E_t \left\{ u_1^n \left(c_{t+1}^n, l_{t+1}^n \right) \left[\theta_{t+1}^n + a_{t+1}^n A f_1^n \left(k_{t+1}^n, l_{t+1}^n \right) \right] \right\} k_{t+1}^n}{u_1^n \left(c_t^n, l_t^n \right) \omega_t^n}, \tag{7}$$

where $n \in \{1, ..., N\}$. The validity of representation (7) requires that in equilibrium, no capital stocks ever reaches zero, which was indeed the case in our simulations. In the Appendix, we provide the Euler equations for Models 5, 7 and 8 in (13), (17) and (23), respectively.

The state space includes $2N$ state variables, which in period t, are N capital stocks $\{k_t^n\}_{n=1}^N$ and N current technology shocks $\{a_t^n\}_{n=1}^N$. We approximate each country's decision rule for the next-period capital stock, k_{t+1}^n, by an exponentiated polynomial of state variables. When the number of countries is large, even low-order polynomial approximations require computing a large number of coefficients and can be computationally costly. For example, if $N = 10$, a first-order polynomial has $(2N + 1) N = 210$ coefficients, and the second-order polynomial has $(1 + 2N + N (2N + 1)) N = 2310$ coefficients. Moreover, the number of polynomial coefficients increases exponentially with the order of a polynomial, which is the so-called curse of dimensionality. We therefore restrict attention to the second-order polynomial approximations in which we exclude the cross-terms. (For example, under $N = 10$, our reduced second-order polynomial has $(1 + 4N) N = 410$ coefficients). Thus, we postulate the following decision rules for the countries' capital stocks:

$$\begin{bmatrix} k_{t+1}^1 \\ ... \\ k_{t+1}^N \end{bmatrix} = \exp\left\{ \begin{bmatrix} v_0^1 \\ ... \\ v_0^N \end{bmatrix} + \begin{bmatrix} v_1^1 & ... & v_N^1 \\ ... & \ddots & ... \\ v_1^N & ... & v_N^N \end{bmatrix} \begin{bmatrix} \log\left(k_t^1\right) \\ ... \\ \log\left(k_t^N\right) \end{bmatrix} + \right.$$
$$+ \begin{bmatrix} v_{N+1}^1 & ... & v_{2N}^1 \\ ... & \ddots & ... \\ v_{N+1}^{2N} & ... & v_{2N}^{2N} \end{bmatrix} \begin{bmatrix} \log\left(a_t^1\right) \\ ... \\ \log\left(a_t^N\right) \end{bmatrix} + \begin{bmatrix} v_{2N+1}^1 & ... & v_{3N}^1 \\ ... & \ddots & ... \\ v_{2N+1}^N & ... & v_{3N}^N \end{bmatrix} \begin{bmatrix} \log^2\left(k_t^1\right) \\ ... \\ \log^2\left(k_t^N\right) \end{bmatrix} +$$
$$\left. + \begin{bmatrix} v_{3N+1}^1 & ... & v_{4N}^1 \\ ... & \ddots & ... \\ v_{3N+1}^N & ... & v_{4N}^N \end{bmatrix} \begin{bmatrix} \log^2\left(a_t^1\right) \\ ... \\ \log^2\left(a_t^N\right) \end{bmatrix} \right\}. \quad (8)$$

If a set of parameters $\mathbf{v} \equiv \left\{ \left\{ v_j^n \right\}_{j=0}^{4N} \right\}_{n=1}^{N}$ is fixed, equations (2), (4), (5), (8) determine uniquely the model's dynamics. Our objective is to find the unknown parameters in approximation (8).

We compute the set of parameters $\mathbf{v}$ by using the following iterative procedure:

- *Step* 1. Fix initial conditions. We choose $k_0^n = k_{ss}$, and $a_0^n = 1$ for $n = 1, ..., N$.[4] Draw and fix for all simulations a random series $\left\{ \left\{ a_t^n \right\}_{n=1}^{N} \right\}_{t=0}^{T}$ by using (3). For an iteration i, fix a set of parameters $\mathbf{v} = \mathbf{v}\left(i\right) \in R^{(4N+1)N}$. Specifically, for an initial iteration, we assume

  $$v_s^n = 1 \quad \text{if} \quad s = n \qquad \text{and} \qquad v_s^n = \varepsilon \quad \text{if} \quad s \neq n,$$

  where $n = 1, ..., N$, $s = 1, ..., 4N$ and ε is a small number (we take $\varepsilon = 10^{-5}$).[5]

- *Step* 2. Given $\mathbf{v}\left(i\right)$, recursively calculate $\left\{ \left\{ k_{t+1}^n \right\}_{n=1}^{N} \right\}_{t=0}^{T}$ from (8).

- *Step* 3. Given $\left\{ \left\{ k_{t+1}^n \right\}_{n=1}^{N} \right\}_{t=0}^{T}$, solve for $\left\{ \left\{ c_t^n \right\}_{n=1}^{N}, \left\{ l_t^n \right\}_{n=1}^{N} \right\}_{t=0}^{T}$ satisfying (2), (4) and (5).

---

[4] Here, and further on in the text, z_{ss} denotes a steady state value of a variable z.

[5] Non-zero initial values of the coefficients are needed in order to initialize the MATLAB subroutine "nlinfit", which we use to perform the nonlinear regression.

- *Step* 4. Construct variables $\left\{\left\{\widetilde{k}_{t+1}^{n}\right\}_{n=1}^{N}\right\}_{t=0}^{T}$, which are the realizations of the variables in the conditional expectation of the Euler equation (7),

$$\widetilde{k}_{t+1}^{n} \equiv \frac{\beta\left\{u_1^n\left(c_{t+1}^n, l_{t+1}^n\right)\left[\theta_{t+1}^n + a_{t+1}^n A f_1^n\left(k_{t+1}^n, l_{t+1}^n\right)\right]\right\} k_{t+1}^n}{u_1^n\left(c_t^n, l_t^n\right)\omega_t^n}. \tag{9}$$

  Run a nonlinear least-square regression of the constructed variables on explanatory function (6) and call the estimated vector of parameters $G\left(\mathbf{v}\left(i\right)\right)$.

- *Step* 5. Compute the vector $\mathbf{v}\left(i+1\right)$ for the next iteration

$$\mathbf{v}\left(i+1\right) = \left(1-\mu_v\right)\mathbf{v}\left(i\right) + \mu_v G\left(\mathbf{v}\left(i\right)\right), \tag{10}$$

  where $\mu_v \in (0,1)$ is the updating parameter.

Iterate on *Steps* $2-5$ until a fixed point is found, $\mathbf{v}^* = G\left(\mathbf{v}^*\right)$.

We shall now describe how to perform *Step* 3. For each t, we have $2N$ conditions, specifically, one resource constraint (2), $N-1$ conditions of type (4) and N conditions of type (5), which allow us to restore $2N$ unknowns, $\{c_t^n, l_t^n\}_{n=1}^N$. In principle, we can solve for the consumption and labor allocations by using a numerical solver. However, this might be computationally costly as we are to find a solution to a $2N$-dimensional system of non-linear equations $T \times I$ times, where I is the number of iterations necessary for convergence (for example, if $T = 10000$ and $I = 1000$, we have $T \times I = 10^7$). We describe two cheap alternatives to the procedure of solving for consumption and labor by a numerical solver on each iteration.

The first alternative is used in Model 5: we reduce the computational cost by calculating the relation between the individual and aggregate consumption outside of the iterative cycle.[6] Specifically, let us consider FOC (4) and construct a grid for values for aggregate consumption $C_t = \sum_{n=1}^N c_t^n$ such that $\{C_1, C_2, ..., C_M\}$. The grid should be chosen so that the value of C_t, which can effectively occur along simulations, is always within the range $[C_1, C_M]$. Define the grid function for individual consumption $c^n\left(C_m\right)$, $n = 1, ..., N$, $m = 1, ..., M$ by computing the solution to FOC (4) for each $C_m \in \{C_1, C_2, ..., C_M\}$. To be specific, from (4), we express consumption of

[6] This approach is also described in Maliar and Maliar (2005).

countries $n = 2, ..., N$ in terms of consumption of country $n = 1$; and from resource constraint (2), we express the aggregate consumption; see condition (14) in the Appendix. Within the iterative cycle, with our grid function, we can compute c_t^1 by interpolation and restore the remaining c_t^n.

The second alternative we describe for Models 7 and 8. It is a simple updating approach, which is similar in spirit to the one we use in *Step* 5, but instead of updating coefficients, we update labor allocations. Specifically, from (5), we solve for consumption in terms of labor for all countries $n = 1, ..., N$ (see conditions (18) and (24) for Model 7 and Model 8, respectively); from (4), we express labor of countries $n = 2, ..., N$ in terms of labor of country $n = 1$ (see conditions (19) and (25) for Model 7 and Model 8, respectively); and finally, from resource constraint (2), we express labor of country 1 in terms of the other variables (see conditions (20) and (26) for Model 7 and Model 8, respectively).

Let $\left\{ \{ l_t^n (j) \}_{n=1}^N \right\}_{t=0}^T$ be the labor series for Models 7 and 8 obtained on iteration j. Use an updating condition like (10) in order to recompute the series and call the new series $\left\{ \{ \mathcal{L} (l_t^n (j)) \}_{n=1}^N \right\}_{t=0}^T$. Find the series for the next iteration by using the updating

$$l_t^n (j+1) = (1 - \mu_l) \, l_t^n (j) + \mu_l \mathcal{L} (l_t^n (j)) , \tag{11}$$

where $\mu_l \in (0, 1)$ is the updating parameter. Iterate until a fixed point $\{ l_t^n \}^* = \mathcal{L} (\{ l_t^n \}^*)$ for all t, n is found with a sufficient degree of accuracy. On the first iteration ($j = 0$), we assume that $l_t^n (0)$ is equal to its steady state value for all t and n.

Let us finally discuss the convergence properties of the PEA. It is well-known that the PEA is not a contraction mapping method, and it does not automatically guarantee finding a solution. The problem arises if the initial guess about the parameters $\mathbf{v}$ is very far from the true solution, in which case the simulated series can become non-stationary and can break down the regression. In order to enhance the convergence properties of the algorithm, we use the approach "moving bounds" described in Maliar and Maliar (2003b). We specifically restrict each capital stock k_{t+1}^n in the model to be within bounds $\left[\underline{k} (i) , \overline{k} (i) \right]$ such that

$$\underline{k} (i) = k_{ss} \exp (-\lambda i) ,$$

$$\overline{k} (i) = k_{ss} (2 - \exp (-\lambda i)) ,$$

where i is the number of iterations performed, and $\lambda > 0$ is the parameter determining the pace at which the bounds are moved. Under this choice, on the first iteration ($i = 0$), the simulated series coincide with the steady state solution, $k_t^n(0) = k_{ss}$ for all t. On the subsequent iterations, the lower and upper bounds gradually move approaching 0 and $2k_{ss}$, respectively. In fact, our updating procedure (11) for computing labor allocations in Models 7 and 8 may also lead to series with undesirable properties on initial iterations, namely, the series can be highly nonstationary or the series can take negative values. To prevent these undesirable outcomes, we impose bounds on simulated series of consumption and labor as well.

4 Parameterization

In this preliminary version, we restrict attention to one particular set of the parameters. The parameters which are common for all studied models are summarized in Table 1.

Table 1. Common parameters.

α	β	δ	ρ	σ	ϕ	ξ
0.36	0.99	0.025	0.95	0.01	10	2

The country-specific parameters are provided in Table 2.

Table 2. Country-specific parameters.

A5	A7	A8
$(\gamma_m, \gamma_M) = (0.25, 1.0)$	$(\gamma_m, \gamma_M) = (0.25, 0.5)$	$(\gamma_m, \gamma_M) = (0.2, 0.4)$ $(\chi_m, \chi_M) = (0.75, 0.9)$ $(\mu_m, \mu_M) = (-0.3, 0.3)$

The remaining model's parameters are chosen so that both capital and labor in the steady state are equal to one. To be specific, we set the level of technology at $A = \frac{1-\beta+\beta\delta}{\alpha\beta}$. In Model 5 with no labor choice, this value of A ensures that the steady state capital stock is equal to one, while in Models 7 and 8 with labor choice, it ensures that the capital to labor ratio in the steady state is equal to one. Furthermore, if capital and labor are equal to one, then $f^n(k_{ss}^n, l_{ss}^n) = f^n(1,1) = 1$ for any of the production functions considered in Models 5, 7 and 8. In autarky, steady state consumption is identical for all countries and is given by $c_{ss}^{n} = Af^{n}\left(k_{ss}^{n}, l_{ss}^{n}\right) - \delta k_{ss}^{n} = A - \delta$. Substituting steady state consumption, capital and labor in the intratemporal FOC (5) allows us to identify the remaining utility-function parameters of Models 7 and 8: in Model 7, the parameter ψ is computed from (21) and in Model 8, the parameters $b_1, ..., b_n$ are computed from (27) (see the Appendix). We next calibrate the welfare weights to match consumption in the steady state. The welfare weights are defined up to a multiplicative constant. Thus, without a loss of generality, we set the welfare weight of the first country equal to one, $\tau^1 = 1$. We restore the remaining welfare weights from condition (4), which in Models 5, 7 and 8, becomes (16), (22) and (28), respectively, (see the Appendix).

We compute a solution under two lengths of simulation, $T_1 = 3000$ and $T_2 = 10000$. We set the updating parameter in the decision rule for capital (10) at $\mu_v = 0.1$. We use the moving-bounds parameter $\lambda = 0.001$, which approximately corresponds to having $\underline{k} = 0.9k_{ss}$ and $\overline{k} = 1.1k_{ss}$ after 100 iterations. The convergence criterion used was that the L^2 distance obtained between the vectors of coefficients in two subsequent iterations is less than 10^{-10}.

As far as consumption and labor allocations are concerned, finding them with a high degree of accuracy on each iteration on $\mathbf{v}$ would imply a high computational cost and is in fact not useful since on the next iteration, we would have to re-compute the consumption and labor allocations for a different vector $\mathbf{v}$. We therefore do not target a high precision in consumption and labor series on each iteration on $\mathbf{v}$ but do 10 updatings of the consumption and labor series according to (11) under $\mu_x = 0.01$. As the solution for capital stock characterized by $\mathbf{v}$ is refined along the iterations, so do the consumption and labor series. Our convergence criterion for consumption and labor is that the L^2 distance obtained between the corresponding time series on two subsequent iterations is less than 10^{-10}.

As far as the initial guess is concerned, we proceed in two steps. First, we compute a solution under a first-order exponentiated polynomial by starting from a non-stochastic steady state. Second, we use the obtained results as an initial guess for our second order polynomial approximation. Our program is written in Matlab, and we run simulations on Pentium 4 PC with 3.33Ghz processor.

5 Results

Our algorithm was successful in finding the solutions to all considered models. For the sake of comparison, we report the solutions under both simulation lengths considered $T_1 = 3000$ and $T_2 = 10000$. In Figures 1, 2 and 3, we illustrate the solution to the Model 8 under $T_2 = 10000$ for $N = 2, 6, 10$ countries.

We now discuss the accuracy issues. Under our computational method, the only relevant source of approximation errors is the Euler equation errors. First, the resource constraint (2) is by construction satisfied exactly. Second, to compute numerically the consumption function in Model 5, we use very fine grid for aggregate consumption, namely, 1000 points, situated between 0.5 and 1.5 level of aggregate consumption in the steady state; the fine grid guarantees a high degree of accuracy. Thirs, in Models 7 and 8, we iterate on labor allocations until the approximation error in each allocation is less than 10^{-10}, which again implies a very high accuracy. Thus, all the approximation errors under our method are a priory negligible compared to the Euler equation errors. We therefore focus only on the last type of errors. Before performing the tests, we removed the first 100 observations to eliminate the effect of initial conditions.

To evaluate the errors of the Euler-equation approximation, we express the Euler equation (4) in the following normalized form:

$$\epsilon_t^n = \frac{\beta E_t \left\{ u_1^n \left(c_{t+1}^n, l_{t+1}^n \right) \left[\theta_{t+1}^n + a_{t+1}^n A f_1^n \left(k_{t+1}^n, l_{t+1}^n \right) \right] \right\}}{u_1^n \left(c_t^n, l_t^n \right) \omega_t^n} - 1. \tag{12}$$

Under $N = 2$, the conditional expectation was computed by using product four-point Gauss-Hermite integration and uder $N = 6$ and $N = 10$, it was computed by using a monomial formula of degree 3, as is described in Judd (1998). In Test 1, we compute the approximation error of the Euler equation across a set of 100 points for capital stocks and shocks, $x = 1, ..., 100$, that are randomly chosen at each of the following radia $r \in \{0.01, 0.1, 0.3\}$ from the deterministic steady state. In Test 2, we compute the errors on 100 points for capital stocks and shocks that are drawn from the simulated series in periods $t = 10, 20, 30, ..., 100$. Finally, in Test 3, we calculate the Denhaan-Marcet statistics, as described in Taylor and Uhlig (1990),

$$m^n = (a^n)' (z'z) \left[z'z (\eta^n)^2 \right]^{-1} (z'z) a^n,$$

where $a^n = (z'z)^{-1} z'\eta^n$ is the ordinary least square estimator in the regression of the Euler equation residual, $\eta^n \equiv [\epsilon_1^n, ..., \epsilon_T^n]'$, with $\epsilon_1^n, ..., \epsilon_T^n$ defined in (12) on a list z which includes a constant, the current state variables, and the second-order monomials of the current state variables. The Denhaan-Marcet statistic has approximately $\chi(11)$ distribution, see den Haan and Marcet (1994). A two-sided test at a significance level of 2.5% at each side is $3.82 < m^n < 21.92$.

For each test implemented, we report computational time (in seconds), which is denoted by T(test 1), T(test 2) and T(test 3) for Tests 1, 2 and 3, respectively. In Test 1, for each radius r, we provide the maximum Euler equation errors $\epsilon^{\max} = \max\limits_{n\in\{1,...,N\},x=1,...,10} \{\|\epsilon_t^n(x)\|\}$, where $\|\cdot\|$ denotes the absolute value. In Test 2, we provide both the mean Euler equation error, $\overline{\epsilon} = \frac{1}{10N}\sum_{x=1}^{10}\sum_{n=1}^{N}\|\epsilon_t^n(x)\|$, and the maximum Euler equation error. Finally, in Test 3, we report the minimum and the maximum of the Denhaan-Marcet statistic, $m^{\min} = \min\limits_{n\in\{1,...,N\}}\{m^n\}$ and $m^{\max} = \max\limits_{n\in\{1,...,N\}}\{m^n\}$, respectively. The computational time and the results of the accuracy tests for Models 5, 7 and 8 are presented in Tables 3, 4, and 5, respectively (notation $z(-p)$ means $z10^{-p}$).

Table 3. The results for Model A5.

	$N = 2$		$N = 6$		$N = 10$	
Sim. length	3000	10000	3000	10000	3000	10000
T (solution)	1m15s	4m24s	27m09s	1h03m	8h15m	14h33m
T(test 1)	0.63s	0.63s	1.76s	1.86s	42.20s	42.78s
$\epsilon^{\max}(r_1)$	3(-4)	2(-4)	4(-4)	4(-4)	8(-4)	5(-4)
$\epsilon^{\max}(r_2)$	4(-3)	4(-4)	5(-3)	2(-3)	5(-3)	2(-3)
$\epsilon^{\max}(r_3)$	2(-2)	5(-3)	3(-2)	9(-3)	3(-2)	1(-2)
T(test 2)	0.23s	0.24s	0.56s	0.63s	14.05s	14.08s
$\overline{\epsilon}$	2(-4)	2(-4)	3(-4)	2(-4)	4(-4)	3(-4)
$\epsilon^{\max}$	1(-3)	5(-4)	2(-3)	1(-3)	2(-3)	2(-3)
T(test 3)	0.16s	0.55s	8.25s	28.95s	1m17s	4m26s
$m^{\min}$	4.75	5.91	71.74	54.21	268.53	182.24
$m^{\max}$	7.91	16.60	89.52	79.95	302.12	236.94

Table 4. The results for Model A7.

	$N = 2$		$N = 6$		$N = 10$	
Sim. length	3000	10000	3000	10000	3000	10000
T (solution)	4m15s	8m56s	44m12s	1h10m	9h12m	13h48m
T(test 1)	15.94s	16.16s	5m32s	5m15s	3h25m	4h05m
$\epsilon^{\max}$ (r_1)	7(-4)	2(-4)	1(-3)	4(-4)	1(-3)	7(-4)
$\epsilon^{\max}$ (r_2)	7(-3)	5(-4)	1(-2)	2(-3)	5(-3)	3(-3)
$\epsilon^{\max}$ (r_3)	5(-2)	5(-3)	8(-2)	1(-2)	3(-2)	2(-2)
T(test 2)	5.39s	5.42s	1m53s	1m46s	1h19m	1h40m
$\bar{\epsilon}$	2(-4)	1(-4)	4(-4)	3(-4)	5(-4)	3(-4)
$\epsilon^{\max}$	1(-3)	5(-4)	2(-3)	1(-3)	3(-3)	2(-3)
T(test 3)	0.13s	0.59s	8.78s	28.99s	1m20s	5m15s
$m^{\min}$	9.97	23.45	73.27	98.74	200.69	211.63
$m^{\max}$	10.71	27.13	98.85	131.16	252.19	266.12

Table 5. The results for Model A8.

	$N = 2$		$N = 6$		$N = 10$	
Sim. length	3000	10000	3000	10000	3000	10000
T (solution)	4.07m	12.03m	24m18s	1h24m	3h21m	4h34m
T(test 1)	30.86s	30.45s	1h21m	2h01m	5h26m	24h17m
$\epsilon^{\max}$ (r_1)	4(-4)	3(-4)	7(-4)	7(-4)	8(-4)	8(-4)
$\epsilon^{\max}$ (r_2)	3(-3)	6(-4)	2(-3)	2(-3)	4(-3)	2(-3)
$\epsilon^{\max}$ (r_3)	2(-2)	7(-3)	2(-2)	1(-2)	4(-2)	1(-2)
T(test 2)	10.36s	10.23s	42m05s	42m41s	1h56m	8h18m
$\bar{\epsilon}$	3(-4)	2(-4)	6(-4)	4(-4)	7(-4)	4(-4)
$\epsilon^{\max}$	1(-3)	7(-4)	3(-3)	2(-3)	4(-3)	2(-3)
T(test 3)	0.13s	0.63s	8.38s	25.45s	1m05s	4m28s
$m^{\min}$	16.23	50.88	88.07	104.38	215.10	222.39
$m^{\max}$	16.44	57.21	112.19	131.45	245.69	262.82

The results in the tables show similar regularities for all the models considered.

- The computational time increases around 2 times when the length of simulations increases from $T_1 = 3000$ to $T_2 = 10000$ and the error can decrease up to several times.

- The model delivers errors in the magnitude of 10(−4) in the ergodic distribution and with the radia which are close to the mean of ergodic distribution, however, the accuracy is sharply reduced if the radia increases. This is because the PEA computes the solution on the ergodic distribution, and thus, does not accurately predict the tails.

- Given the length of simulation, the accuracy is reduced as the number of countries increases. This is because we have the same number of observations to identify a larger number of parameters.

- As far as Test 3 is concerned, for $N = 2$, the Denhaan-Marcet statistics are mainly within the critical interval, but for other cases, they are larger than the upper critical value. However, many algorithms commonly accepted in the literature fail on this statistics (see Taylor and Uhlig, 1990, page 14).

6 Conclusion

The simulation-based PEA, described in this paper, was successful in solving a complete-market neoclassical stochastic growth model with a finite number of countries. In all experiments considered, the algorithm was able to converge to the true solution starting from the non-stochastic steady state. Our algorithm can be applied to models with either homogeneous or heterogeneous countries without modifications, as it does not rely on countries' symmetricity. The PEA we use is not particularly costly: under the length of simulation $T = 10000$, it takes few minutes to solve a two-country model, about 1 hour to solve a six-country model and up to 14 hours to solve a ten-country model. In fact, the speed of computations was not our priority: we set the algorithm's parameters to values, which are common for all models considered and which guarantee convergence even if we start iterations far from the true solution. In most of the cases, computational time could be reduced twice or more if we change the algorithm's parameters depending on a specific model. Overall, we view all the obtained solutions as sufficiently accurate: when the state variables are drawn from the ergodic distribution, the errors do not exceed 10^{-3}, i.e., 0.1%. This magnitude of errors is likely to be not essential for many economically relevant applications. For example, such errors would play a little role in the model's second-moment properties, which are standartly used in the real-business-cycle literature for describing the predictions of the studied models.

References

[1] den Haan, W., and Marcet, A. (1990), "Solving the stochastic growth model by parameterizing expectations," *Journal of Business and Economic Statistics,* 8, 31-34.

[2] den Haan, W., and Marcet, A. (1994), "Accuracy in simulations," *Review of Economic Studies*, 6, 3-17.

[3] Judd, K. (1992), " Projection methods for solving aggregate growth models," *Journal of Economic Theory*, 58, 410-452.

[4] Judd, K. (1998), " Numerical methods in economics," London, England: The MIT Press, Cambridge Massachusetts.

[5] Maliar, L. and Maliar, S. (2003a), The representative consumer in the neoclassical growth model with idiosyncratic shocks, *Review of Economic Dynamics*, 6, 362-380.

[6] Maliar, L., and Maliar, S. (2003b) "Parameterized expectations algorithm and the moving bounds", *Journal of Business and Economic Statistics*, 21/1, 88-92.

[7] Maliar, L. and Maliar, S. (2005) "Parameterized Expectations Algorithm: How to Solve for Labor Easily", *Computational Economics*, 25, 269-274.

[8] Marcet, A., and Lorenzoni, G. (1999), "The parameterized expectation approach: some practical issues," in *Computational Methods for Study of Dynamic Economies,* eds. R. Marimon and A. Scott, New York: Oxford University Press, 143-171.

[9] Taylor, J., and Uhlig, H., (1990) "Solving nonlinear stochastic growth models: a comparison of alternative solution methods", *Journal of Business and Economic Statistics*, 8 (1), 1-17.

7 Appendix

In this section, we provide the optimality conditions (written in the forms used in iterations) as well as conditions used for calibrating the parameters in our three models of interest.

Model A5 There is no labor choice, so that condition (5) is absent. The remaining conditions (7), (2) and (4), respectively, are

$$\widetilde{k}_{t+1}^{n} = \frac{\beta E_t \left\{ \left(c_{t+1}^{n}\right)^{-1/\gamma_n} \left[\theta_{t+1}^{n} + \alpha a_{t+1}^{n} A \left(k_{t+1}^{n}\right)^{\alpha-1}\right]\right\}}{\left(c_t^{n}\right)^{-1/\gamma_n} \omega_t^{n}} k_{t+1}^{n}, \tag{13}$$

$$C_t = \sum_{n=1}^{N} c_t^n = \sum_{n=1}^{N} \left[(1-\delta) k_t^n + a_t^n A \left(k_t^n\right)^{\alpha} - \frac{\varphi}{2} k_t^n \left(\frac{k_{t+1}^n}{k_t^n} - 1\right)^{\xi} - k_{t+1}^n \right], \tag{14}$$

$$c_t^n = \left[\left(c_t^1\right)^{-1/\gamma_1} \tau^1 / \tau^n\right]^{-\gamma_n}. \tag{15}$$

To calibrate the welfare weights of countries $n = 2, ..., N$, we use condition (15) evaluated in the steady state,

$$\tau^n = \left(c_{ss}^1\right)^{-1/\gamma_1} / \left(c_{ss}^n\right)^{-1/\gamma_n}. \tag{16}$$

Model A7 Conditions (7), (5) and (4), respectively, are

$$\widetilde{k}_{t+1}^{n} = \frac{\beta E_t \left\{ \frac{\left[\left(c_{t+1}^{n}\right)^{\psi} \left(L^e - l_{t+1}^{n}\right)^{1-\psi}\right]^{1-1/\gamma_n}}{c_{t+1}^{n}} \left[\theta_{t+1}^{n} + \alpha a_{t+1}^{n} A \left(k_{t+1}^{n}\right)^{\alpha-1} \left(l_{t+1}^{n}\right)^{1-\alpha}\right]\right\}}{\frac{\left[\left(c_t^{n}\right)^{\psi} \left(L^e - l_t^{n}\right)^{1-\psi}\right]^{1-1/\gamma_n}}{c_t^{n}} \omega_t^{n}} k_{t+1}^{n}, \tag{17}$$

$$\widetilde{c}_t^n = \frac{\psi \left(L^e - l_t^n\right)}{1-\psi} (1-\alpha) a_t^n A \left(k_t^n\right)^{\alpha} \left(l_t^n\right)^{-\alpha}, \tag{18}$$

$$\widetilde{l}_t^n = L^e - \left[\left(L^e - l_t^1\right)^{(1-\psi)(1-1/\gamma_1)} \frac{\left(c_t^1\right)^{\psi(1-1/\gamma_1)-1} \tau^1}{\left(c_t^n\right)^{\psi(1-1/\gamma_n)-1} \tau^n}\right]^{\frac{1}{(1-\psi)(1-1/\gamma_n)}}. \tag{19}$$

To re-compute $\widetilde{l}_t^1$, we use resource constraint (2)

$$\widetilde{l}_t^1 = \left[\frac{\sum_{n=1}^{N} \left[c_t^n + k_{t+1}^n - (1-\delta) k_t^n + \frac{\varphi}{2} k_t^n \left(\frac{k_{t+1}^n}{k_t^n} - 1 \right)^{\xi} \right] - \sum_{n=2}^{N} a_t^n A \left(k_t^n\right)^{\alpha} \left(l_t^n\right)^{1-\alpha}}{a_t^1 A \left(k_t^1\right)^{\alpha}} \right]^{\frac{1}{1-\alpha}}. \tag{20}$$

We use condition (18) taken in the steady state to calibrate the parameter ψ,

$$\psi = \frac{c_{ss}^n}{c_{ss}^n + \left(L^e - l_{ss}^n\right)(1-\alpha) A \left(k_{ss}^n\right)^{\alpha} \left(l_{ss}^n\right)^{-\alpha}}, \tag{21}$$

and we calibrate the welfare weights of countries $n = 2, ..., N$ from (19),

$$\tau^n = \frac{\left(c_{ss}^1\right)^{\psi(1-1/\gamma_1)-1} \left(L^e - l_{ss}^1\right)^{(1-\psi)(1-1/\gamma_1)}}{\left(c_{ss}^n\right)^{\psi(1-1/\gamma_n)-1} \left(L^e - l_{ss}^n\right)^{(1-\psi)(1-1/\gamma_n)}}. \tag{22}$$

Model A8 Conditions (7), (5), (4) and (2), respectively, are

$$\widetilde{k}_{t+1}^n = \frac{\beta E_t \left\{ \begin{array}{c} \left[\left(c_{t+1}^n\right)^{1-1/\chi_n} + b_n \left(L^e - l_{t+1}^n\right)^{1-1/\chi_n} \right]^{\frac{1/\chi_n - 1/\gamma_n}{1-1/\chi_n}} \left(c_{t+1}^n\right)^{-1/\chi_n} \times \\ \times \left[\theta_{t+1}^n + \alpha a_{t+1}^n A \left(k_{t+1}^n\right)^{\mu_n - 1} \left(\alpha \left(k_{t+1}^n\right)^{\mu_n} + \alpha \left(l_{t+1}^n\right)^{\mu_n} \right)^{1/\mu_n - 1} \right] \end{array} \right\}}{\left[\left(c_t^n\right)^{1-1/\chi_n} + b_n \left(L^e - l_t^n\right)^{1-1/\chi_n} \right]^{\frac{1/\chi_n - 1/\gamma_n}{1-1/\chi_n}} \left(c_t^n\right)^{-1/\chi_n} \omega_t^n} k_{t+1}^n, \tag{23}$$

$$\widetilde{c}_t^n = \left[\frac{(1-\alpha) a_t^n A \left(l_t^n\right)^{\mu_n - 1} \left(\alpha \left(k_t^n\right)^{\mu_n} + (1-\alpha) \left(l_t^n\right)^{\mu_n} \right)^{1/\mu_n - 1}}{b_n} \right]^{\chi_n} \left(L^e - l_t^n\right), \tag{24}$$

$$\widetilde{l}_t^n = L^e -$$

$$\left(\frac{1}{b_n} \left\{ \frac{\left[\left(c_t^1\right)^{1-1/\chi_1} + b_1 \left(L^e - l_t^1\right)^{1-1/\chi_1} \right]^{\frac{1/\chi_1 - 1/\gamma_1}{1-1/\chi_1}} \left(c_t^1\right)^{-1/\chi_1} \tau^1}{\left(c_t^n\right)^{-1/\chi_n} \tau^n} \right\}^{\frac{1-1/\chi_n}{1/\chi_n - 1/\gamma_n}} - \frac{\left(c_t^n\right)^{1-1/\chi_n}}{b_n} \right)^{\frac{1}{1-1/\chi_n}}. \quad (25$$

$$\widetilde{l}_t^1 = \left(\left[\frac{\left\{ \begin{array}{c} \sum_{n=1}^{N} \left[c_t^n + k_{t+1}^n - (1-\delta) k_t^n + \frac{\varphi}{2} k_t^n \left(\frac{k_{t+1}^n}{k_t^n} - 1 \right)^{\xi} \right] - \\ - \sum_{n=2}^{N} a_{t+1}^n A \left(\alpha \left(k_t^n\right)^{\mu_n} + (1-\alpha) \left(l_t^n\right)^{\mu_n} \right)^{1/\mu_n} \end{array} \right\}}{a_t^1 A (1-\alpha)^{1/\mu_n}} \right]^{\mu_n} - \frac{\alpha \left(k_t^n\right)^{\mu_n}}{1-\alpha} \right)^{\frac{1}{\mu_n}}, \tag{26}$$

We calibrate the parameters $b_1, ..., b_N$ from the steady state version of (24),

$$b^n = \frac{(1-\alpha) A \left(l_{ss}^n\right)^{\mu_n - 1} \left(\alpha \left(k_{ss}^n\right)^{\mu_n} + (1-\alpha) \left(l_{ss}^n\right)^{\mu_n} \right)^{1/\mu_n - 1} \left(L^e - l_{ss}^n\right)}{\left(c_{ss}^n\right)^{1/\chi_n}}, \tag{27}$$

and we calibrate the welfare weights according to (25),

$$\tau^n = \frac{\left[\left(c_{ss}^1\right)^{1-1/\chi_1} + b_1 \left(L^e - l_{ss}^1\right)^{1-1/\chi_1} \right]^{\frac{1/\chi_1 - 1/\gamma_1}{1-1/\chi_1}} \left(c_{ss}^1\right)^{-1/\chi_1}}{\left[\left(c_{ss}^n\right)^{1-1/\chi_n} + b_n \left(L^e - l_{ss}^n\right)^{1-1/\chi_n} \right]^{\frac{1/\chi_n - 1/\gamma_n}{1-1/\chi_n}} \left(c_{ss}^n\right)^{-1/\chi_n}}. \tag{28}$$

Figure 1. Model A8 with 2 countries: time series solution.

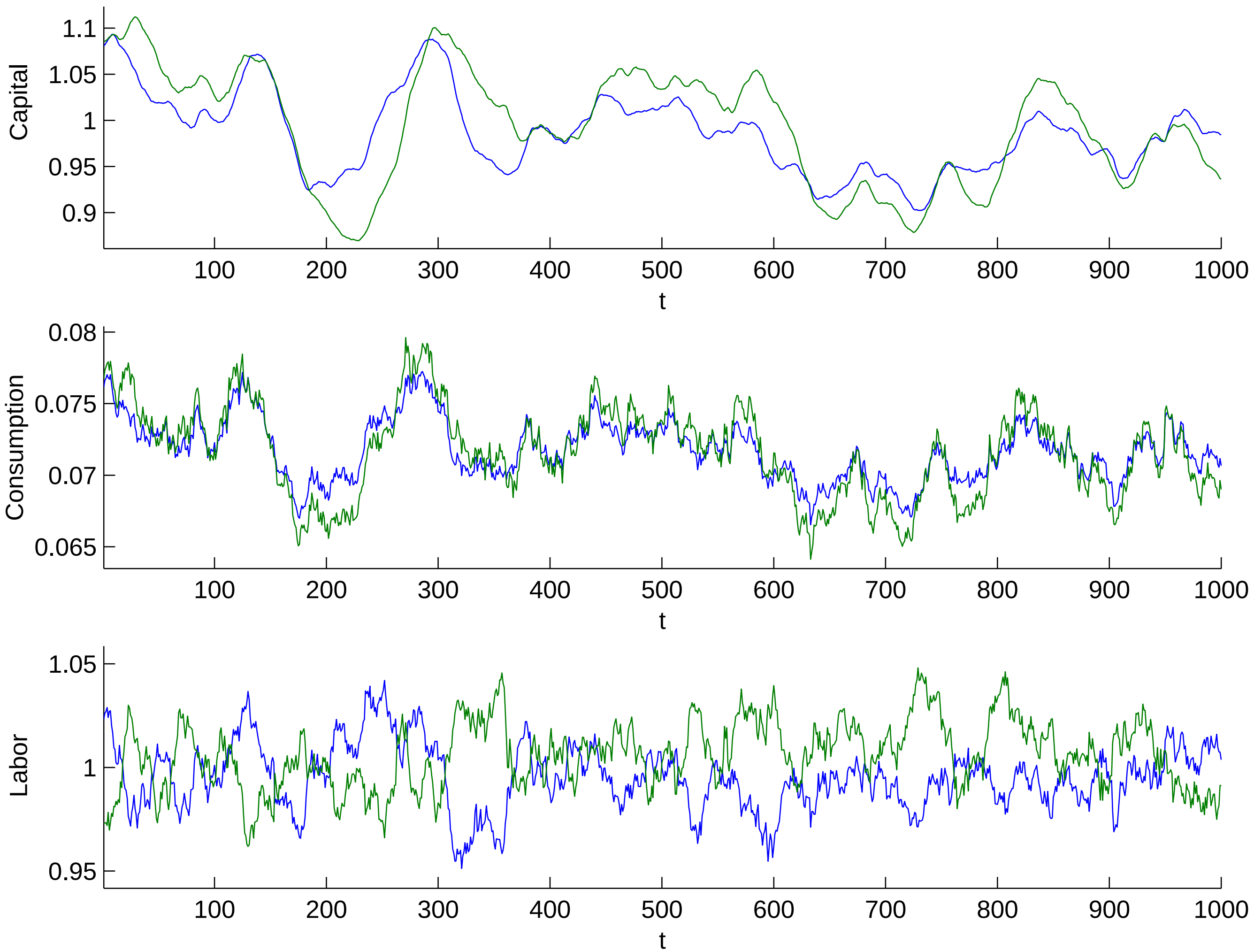

Figure 2. Model A8 with 6 countries: time series solution.

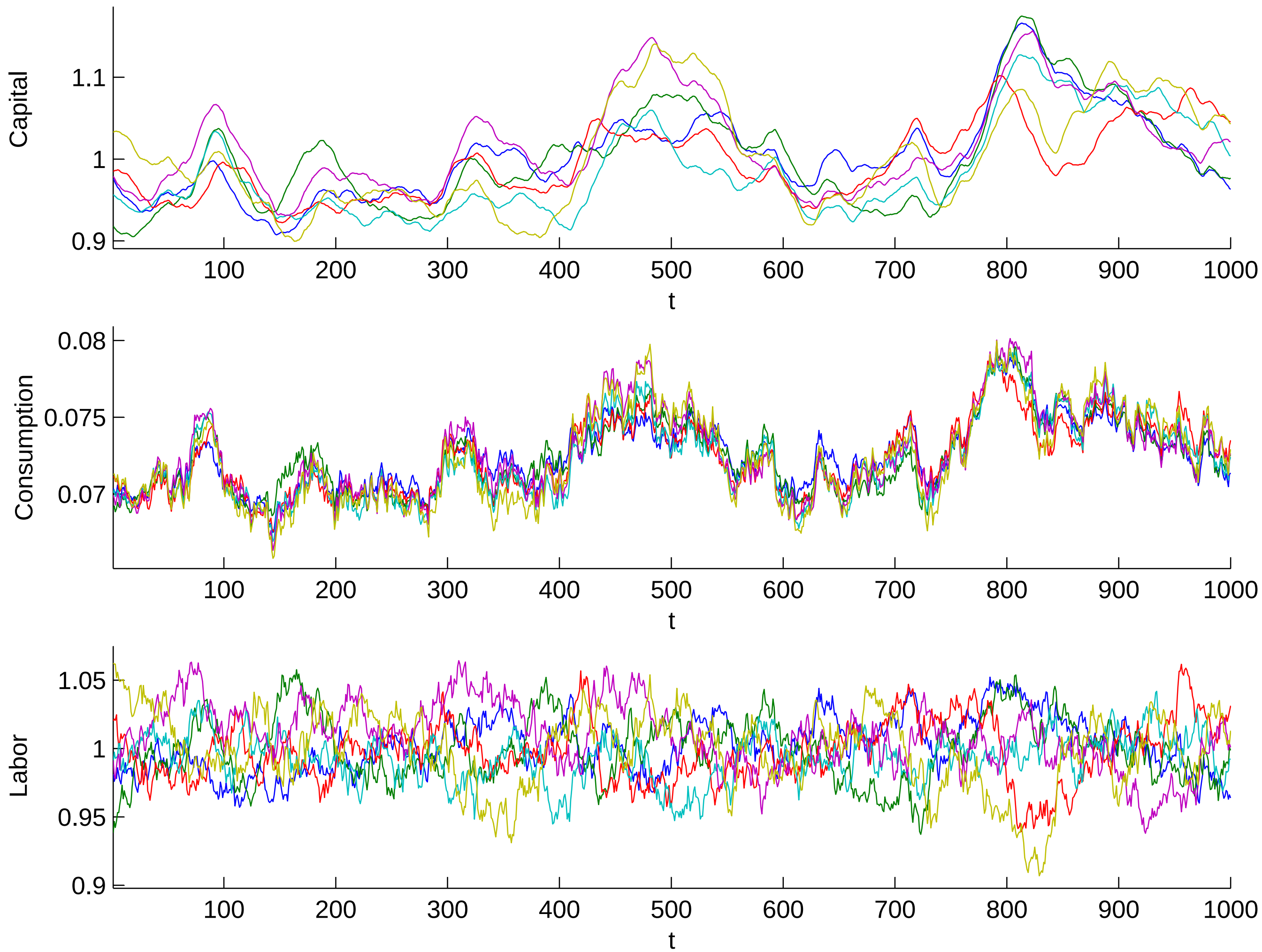

Figure 3. Model A8 with 10 countries: time series solution.