

Envelope Condition Method with an Application to Default Risk Models

Cristina Arellano, Lilia Maliar, Serguei Maliar and Viktor Tsyrennikov*

July 22, 2014

Abstract

We develop an envelope condition method (ECM) for dynamic programming problems – a tractable alternative to expensive conventional value function iteration. ECM has two novel features: First, to reduce the cost, ECM replaces expensive backward iteration on Bellman equation with relatively cheap forward iteration on an envelope condition. Second, to increase the accuracy of solutions, ECM solves for derivatives of a value function jointly with a value function itself. We complement ECM with other computational techniques that are suitable for high-dimensional problems, such as simulation-based grids, monomial integration rules and derivative-free solvers. The resulting value-iterative ECM method can accurately solve models with at least up to 20 state variables and can successfully compete in accuracy and speed with state-of-the-art Euler equation methods. We also use ECM to solve a challenging default risk model with a kink in value and policy functions, and we find it to be fast, accurate and reliable.

EL classification : C6, C61, C63, C68

Key Words : Dynamic programming; Value function iteration; Bellman equation; Endogenous grid; Envelope condition; Curse of dimensionality; Large scale; Sovereign debt; Default risk

*We are indebted to Editor Chris Otrok, two anonymous referees and Kenneth L. Judd for many useful comments and suggestions. Errors are ours. Lilia Maliar and Serguei Maliar acknowledge support from the Hoover Institution and Department of Economics at Stanford University, Santa Clara University, University of Alicante, Ivie and MECED under the grant ECO2012-36719. Corresponding author: Lilia Maliar, Office 249, Department of Economics, Stanford, CA 94305-6072, USA; email: maliarl@stanford.edu; tel: 6507259069.

1 Introduction

We develop an *envelope condition method* (ECM) for dynamic programming problems – a tractable alternative to expensive conventional value function iteration (VFI). ECM has two novel features: First, to reduce the cost, ECM replaces expensive backward iteration on Bellman equation with relatively cheap forward iteration on an envelope condition. Second, to increase the accuracy of solutions, ECM also solves for derivatives of a value function jointly with a value function itself. We complement ECM with other computational techniques that are suitable for high-dimensional problems, such as simulation-based grids, monomial integration rules and derivative-free solvers. The resulting value-iterative ECM method can accurately solve models with at least up to 20 state variables and can successfully compete in accuracy and speed with state-of-the-art Euler equation methods. We also use ECM to solve a challenging default risk model with a kink in value and policy functions, and we find it to be fast, accurate and reliable.

First, as an illustration, we use ECM to solve the standard one-agent optimal growth model with inelastic labor supply. In such a model, either conventional VFI performs numerical optimization (to find a maximum of the right side of the Bellman equation) or it performs numerical rootfinding (to find a solution to a FOC). In contrast, ECM leads us to an alternative recursion that requires only direct calculations and that delivers a solution to the Bellman equation without the need of either numerical optimization or numerical solvers. We establish the following formal results about the ECM recursion: First, the ECM operator has the same fixed point solution as the regular Bellman operator. Second, an appropriately constructed ECM operator with damping is a contraction mapping like the regular Bellman operator. In our numerical experiments, the ECM method has rapid convergence and produces very accurate solutions.

We also develop a version of ECM that approximates derivatives of value function (possibly, jointly with value function) instead of value function itself. This version of ECM produces more accurate solutions than an otherwise identical ECM that solves exclusively for a value function. This is because solving accurately for value function does not necessarily leads to sufficiently accurate approximations of its derivatives. For example, if a value function is approximated with a polynomial of degree n , then its derivatives are effectively approximated with a polynomial of degree $n - 1$, i.e., we "lose" one polynomial degree when differentiating a value function. In contrast, by approximating derivatives of value function directly, we focus on the object that identifies policy functions and hence, obtain more accurate solutions.

Second, we complement ECM with computational techniques that are tractable in

high-dimensional applications, including stochastic simulation, non-product monomial integration rules, and derivative-free solvers, to solve a multicountry growth model with up to 10 countries (20 state variables).¹ This model is the one studied in the February 2011's special issue of the *Journal of Economic Dynamics and Control* (henceforth, JEDC project) which compares the performance of six state-of-the-art solution methods.² We show that the ECM methods is tractable and reliable in this setting and able to successfully compete with state-of-the-art Euler equation methods in the high-dimensional applications which were part of the JEDC project. For our most accurate third-degree polynomial solutions, maximum unit-free residuals in the model's equations are always smaller than 0.002% on a stochastic simulation of 10,000 observations.

The third application is a default risk model. We apply the ECM to a sovereign default model that is a variant of the Arellano (2008) model. Default models are challenging computationally because value and policy functions have kinks and the price function of debt depends on the level of debt reflecting default probabilities. We show that the ECM methods are reliable and fast in computing this model. Relative to the expensive VFI method, ECM speeds up the computation time by more than 50 times. In this application, we solve a benchmark default risk model. Nevertheless, we think that ECM can be useful for many other applications with default risk. In fact, a substantial hurdle for the growing literature on sovereign default is the computational cost; see, e.g., Aguiar and Gopinath (2006), Chatterjee, Corbae, Nakajima and Ríos-Rull (2007), Maliar, Maliar and Pérez-Sebastián (2008), Chatterjee and Eyigungor (2011), Arellano, Bai, and Kehoe (2013) and Tsyrennikov (2013); see Aguiar and Amador (2013) for a review of the literature on sovereign debt. The ECM methods can facilitate the development of this literature by expanding the types of problems that can be efficiently solved.

There is a large body of literature that focuses on solving DP problems including methods based on discretization of state space (e.g., Rust (1996, 1997)), stochastic simulation methods (e.g., Smith (1991, 1993), Maliar and Maliar (2005)), learning methods (e.g., Bertsekas and Tsitsiklis (1996)), perturbation methods (e.g., Judd (1998)), approximate DP methods (e.g., Powell (2011)), polyhedral approximations

¹See Maliar and Maliar (2014) for a survey of such methods.

²The objectives of the JEDC project are described in Den Haan, Judd and Juillard (2011); the methodology of the numerical analysis is outlined in Juillard and Villemot (2011); the results of the comparison analysis are provided in Kollmann, Maliar, Malin and Pichler (2011). The six participating methods are first- and second-order perturbation methods of Kollmann, Kim and Kim (2011), stochastic simulation and cluster-grid algorithms of Maliar, Maliar and Judd (2011), monomial rule Galerkin method of Pichler (2011) and Smolyak's collocation method of Malin, Krueger and Kubler (2011).

(e.g., Judd, Yeltekin and Conklin (2003), Fukushima and Waki (2011)); also see Rust (2008), Judd (1998), Santos (1999), Stachursky (2009) for literature reviews. From one side, many methods, which are accurate and reliable in problems with low dimensionality, are not tractable in problems with high dimensionality. This is in particular true for projection-style methods that rely on tensor product rules in the construction of either grid points or integration nodes. From the other side, methods that are tractable in high-dimensional problems may be insufficiently accurate. One example is perturbation methods whose accuracy deteriorates rapidly away from the steady state. Another example is simulation-based methods, including approximate DP methods and learning methods, whose accuracy is limited by a low (square-root) rate of convergence of Monte Carlo simulation, see Judd, Maliar and Maliar (2011).

Endogenous grid method (EGM) of Carroll (2005) reduces the cost of conventional VFI; see also Barillas and Fernández-Villaverde (2007), Ishakov, Rust and Schjerning (2012). In a companion paper, Maliar and Maliar (2013) compare the ECM and EGM methods in the context of a one-agent model with elastic labor supply and find that the two methods are very similar both in terms of accuracy of solutions and computational expense. However, in more complex applications, one method may have advantages over the other. Constructing a grid on future state variables under EGM is complicated in problems with kinks in future state variables (due to, e.g., occasionally binding inequality constraints or default) because it is not known at the stage of initialization if an inequality constraint binds or if a default occurs in a given grid point. This typically requires to nest the EGM method within another iterative procedure; see, e.g., Villemot (2012), Fella (2014). In contrast, ECM methods use the conventional current state variables and may be easier to implement.

Other papers that solve high-dimensional problems are multicountry models with 20-60 state variables in Judd, Maliar and Maliar (2011, 2012, 2013); medium scale new Keynesian models in Judd, Maliar and Maliar (2012), Fernández-Villaverde, Gordon, Guerrón-Quintana, and Rubio-Ramírez (2012), Aruoba and Schorfheide (2013), Lopez-Salido (2014), etc.; and large-scale OLG models with 80 state variables in Hasanhodzic and Kotlikoff (2013). All the methods that solve high-dimensional problems, including those participating in the JEDC project, build on Euler equations, and none of these papers uses value iterative approaches even when the studied models admit a dynamic programming representation. However, value iterative approaches that build on ECM can successfully compete with state-of-the-art Euler equation methods in high-dimensional applications. In particular, we are able to compute polynomial approximations up to third degree, while the Euler equation methods participating in the comparison analysis of Kollmann, Maliar, Malin and

Pichler (2011b) are limited to second-degree polynomials (due to their high computational expense).

The rest of the paper is organized as follows. In Section 2, we illustrate the ECM methods in the context of the standard one-agent neoclassical growth model. In Section 3, we apply the ECM methods to solve the multicountry growth models studied in the JEDC project. In Section 4, we apply the ECM method to solve a default risk model. In Section 5, we conclude.

2 ECM in the one-agent growth model

We illustrate the envelope condition method (ECM) in the context of the standard one-agent neoclassical growth model.

2.1 The model

We consider a dynamic programming (DP) problem of finding a value function, V , that solves the Bellman equation,

$$V(k, z) = \max_{c, k'} \{u(c) + \beta E[V(k', z')]\} \quad (1)$$

$$\text{s.t. } k' = (1 - \delta)k + zf(k) - c, \quad (2)$$

$$\ln z' = \rho \ln z + \varepsilon', \quad \varepsilon' \sim \mathcal{N}(0, \sigma^2), \quad (3)$$

where k , c and z are capital, consumption and productivity level, respectively; $\beta \in (0, 1)$; $\delta \in (0, 1]$; $\rho \in (-1, 1)$; $\sigma \geq 0$; the utility and production functions, u and f , respectively, are strictly increasing, continuously differentiable and concave; the primes on variables denote next-period values, and $E[V(k', z')]$ is an expectation conditional on state (k, z) .

We assume that a solution to Bellman equation (1)–(3) is interior and that V is differentiable. Therefore, the optimal quantities satisfy a first-order condition (FOC) with respect to consumption and an envelope condition, which, respectively, are

$$u_c(c) = \beta E[V_k(k', z')], \quad (4)$$

$$V_k(k, z) = u_c(c) [1 - \delta + zf_k(k)]. \quad (5)$$

Here and further in the paper, $F_x(\dots, x, \dots)$ denotes a first-order partial derivative of function $F(\dots, x, \dots)$ with respect to variable x .

2.2 Envelope condition methods

A distinctive feature of the ECM solution methods, advocated in the present paper, is that they build on envelope condition (5) instead of FOC (4). To illustrate the idea of the ECM method, let us recall how envelope condition (5) is constructed for the model (1)–(3). We write $V_k(k, z)$ as

$$V_k(k, z) = u_c(C(k, z)) C_k(k, z) + \beta E[V_k(K(k, z), z')] K_k(k, z), \quad (6)$$

where $c = C(k, z)$ and $k' = K(k, z)$ are the optimal policy functions. According to (2), we have $K_k(k, z) = 1 - \delta + zf(k) - C_k(k, z)$, which allows us to re-write (6) as

$$\underbrace{V_k(k, z) = u_c(C(k, z)) [1 - \delta + zf_k(k)]}_{\text{Envelope condition}} + \underbrace{\{\beta E[V_k(k', z')] - u_c(C(k, z))\}}_{\text{FOC}} K_k(k, z). \quad (7)$$

Conventional value function iteration (VFI) solves for $C(k, z)$ by setting FOC (4) equal to zero and hence, it enforces the envelope condition (5) indirectly. In turn, ECM studied here goes the other way around: it solves for $C(k, z)$ using envelope condition (5), and hence, it enforces FOC (4) indirectly. Both ways of iteration must lead to the same solution since the converged policy function $C(k, z)$ must satisfy both FOC (4) and envelope condition (5).

The above difference in the construction of policy functions turned out to be important for both computational expense and convergence properties of value iterative methods. The key advantage of ECM over conventional VFI is that solving for policy functions from an envelope condition is generally easier than solving for it from FOCs. The shortcoming of ECM is that its convergence requires additional and more stringent restrictions such as iteration with damping. In the next section, we describe two variants of ECM, one is ECM-VF that solves for a value function and the other is ECM-DVF that solves for a derivative of value function.

2.2.1 ECM-VF

ECM-VF finds a solution to (1)–(3) by solving the following fixed-point problem:

ECM-VF

Given V , for each point (k, z) , define the following recursion:

- i). $c = u'^{-1} \left[\frac{V_k(k, z)}{1 - \delta + z f'(k)} \right]$.
- ii). $k' = (1 - \delta) k + z f(k) - c$.
- iii). $\widehat{V}(k, z) = u(c) + \beta E[V(k', z')]$.

The optimal value function satisfies $\widehat{V} = V$.

Formulas i) and ii) are envelope condition (5) and budget constraint (2), respectively, and formula iii) is Bellman equation (1) evaluated under optimal policy functions (which eliminates the maximization sign). We observe that under ECM, neither numerical maximization nor numerical solver is necessary for iteration on the Bellman equation: we take some V , find c in i), find k' in ii) and compute a new \widehat{V} in iii): these all are direct calculations.³

Two related methods in the literature are conventional value function iteration (VFI) and endogenous grid method (EGM) of Carroll (2005). Both methods perform time iteration on FOC (4), namely, they guess value function at $t + 1$ and use the Bellman equation to compute value function at t . FOC (4) combined with budget constraint (2) becomes

$$u'(c) = \beta E[V_k((1 - \delta)k + zf(k) - c, z')]. \quad (8)$$

Conventional VFI solves for c that satisfies (8) given (k, z) .

Conventional VFI

Given V , for each point (k, z) , define the following recursion:

- i). Find c satisfying $u'(c) = \beta E[V_k((1 - \delta)k + zf(k) - c, z')]$.
- ii). $k' = (1 - \delta)k + zf(k) - c$.
- iii). $\widehat{V}(k, z) = u(c) + \beta E[V(k', z')]$.

The optimal value function satisfies $\widehat{V} = V$.

Conventional VFI is expensive because Step i) requires us to numerically find a root to (8) for each (k, z) by interpolating V_k to new values (k', z') and by approximating conditional expectation – this all must be done inside an iterative cycle; see Aruoba, Fernández-Villaverde and Rubio-Ramírez (2006) for an example of cost assessment

³We also studied a variant of ECM-VF that performs policy function iteration instead of value function iteration. Such a method guesses policy functions for consumption $c = C(k, z)$ and/or capital $k = K(k, z)$, computes the corresponding V and uses it to recompute the policy functions, iterating until convergence; see Rust (2008) and Stachursky (2009) for a general discussion about value and policy function iterations. An example of ECM iterating on policy function is shown in Section 3.

of conventional VFI. (Alternatively, we can find k' by maximizing the right side of Bellman equation (1) directly without using FOCs, however, this is also expensive).

EGM of Carroll (2005) exploits the fact that it is easier to solve (8) with respect to c given (k', z) than to solve it with respect to c given (k, z) . EGM constructs a grid on (k', z) by fixing the future endogenous state variable k' and by treating the current endogenous state variable k as unknown. Since k' is fixed, EGM computes $E[V_k(k', z')]$ up-front and thus can avoid costly interpolation and approximation of expectation in a rootfinding procedure.

EGM of Carroll (2005)

Given V , for each point (k', z) , define the following recursion:

- i). $c = u'^{-1} \{\beta E[V_k(k', z')]\}$.
- ii). Find k satisfying $k' = (1 - \delta)k + zf(k) - c$.
- iii). $\widehat{V}(k, z) = u(c) + \beta E[V(k', z')]$.

The optimal value function satisfies $\widehat{V} = V$.

In Step ii) of EGM, we still need to find k numerically. However, for the studied model, Carroll (2005) shows a change of variables that makes it possible to avoid finding k on each iteration (except of the very last iteration).

Our ECM-VF differs from the above two methods in that it iterates forward rather than backward. To be specific, it makes a guess on the current value function instead of the future value function, and it solves for the model's variables using envelope condition (5) instead of FOC (4). As we had seen, ECM-VF avoids the rootfinding completely in studied model (even without a change of variables). Thus, it attains the same result as the EGM of Carroll (2005) via a different mechanism. In more complicated models, for example, an optimal growth model with elastic labor supply, neither EGM nor ECM avoid the rootfinding completely but still simplify it considerably compared to conventional VFI; see Barillas and Fernández-Villaverde (2007) for an extension of EGM to a model with elastic labor supply. In a companion paper, Maliar and Maliar (2013) show that ECM-VF and EGM perform very similarly in terms of their accuracy and speed in the context of a model with elastic labor supply.

EGM of Carroll (2005) is non-trivial to implement for models in which future state variables have kinks, for example, for models with occasionally binding inequality constraints or with a risk of default. Since EGM needs to construct a grid on future state variables, we must specify whether an inequality constraint binds or whether a default occurs in each grid point. This is generally not possible to do before the model is solved. To deal with this complication, the literature nests EGM within another iterative procedure; see, e.g., Villemot (2012), Fella (2014). The proposed

ECM methods may have advantages over EGM for this kind of problems as they construct grids on present state variables whose values are naturally known. In Section 4, we apply an ECM method to solve a default risk model.

2.2.2 ECM-DVF

ECM-DVF is similar to ECM-VF however it solves for a derivative of value function instead of value function itself. Combining (4) and (5) gives the following useful recursion for a derivative of the value function

$$V_k(k, z) = \beta [1 - \delta + z f'(k)] E [V_k(k', z')]. \quad (9)$$

Recursion (9) can be viewed as an analogue of the Bellman equation written for a derivative of value function.

For the model (1)–(3), the ECM-DVF fixed-point problem is as follows:

ECM-DVF

Given V_k for each point (k, z) , define the following recursion:

- i). $c = u'^{-1} \left[\frac{V_k(k, z)}{1 - \delta + z f'(k)} \right]$.
- ii). $k' = (1 - \delta) k + z f(k) - c$.
- iii). $\widehat{V}_k(k, z) = \beta [1 - \delta + z f'(k)] E [V_k(k', z')]$.

The derivative of optimal value function satisfies $\widehat{V}_k = V_k$.

Given a converged V_k , find V satisfying $\widehat{V}(k, z) = u(c) + \beta E [V(k', z')]$.

The difference of ECM-DVF from the previously studied ECM-VF consists in that we iterate on V_k without computing V on each iteration. We only compute V at the very end, when both V_k and the optimal policy functions are constructed.

Again, neither numerical maximization nor a numerical solver is necessary under ECM-DVF. Instead, only direct calculations are needed: we take some V_k , find c in i), find k' in ii) and compute a new \widehat{V}_k in iii): these are all direct calculations. We also studied a variant of ECM-DVF that performs policy function iteration instead of value function iteration.

In our numerical experiments, a version of ECM-DVF that solves for a derivative of value function produces more accurate solutions than an otherwise identical ECM-VF that solves exclusively for a value function. This is because solving accurately for value function does not lead to sufficiently accurate approximations of its derivatives. To see the point, assume that a value function is approximated with polynomial of degree n . Then, its derivatives are effectively approximated with a polynomial of degree $n - 1$ since we "lose" one polynomial degree when differentiating a value

function. In contrast, when approximating derivatives of value function directly, we focus on the object that identifies policy functions and as a result, we obtain more accurate solutions.

ECM-DVF has similarity to Euler equation methods; see Judd (1998) for a general discussion of such methods. The "usual" Euler equation follows from optimality conditions (4), (5): we update (5) to obtain $V_k(k', z')$ and we substitute the result into (4) to eliminate the unknown derivative of the value function,

$$u'(c) = \beta E [u'(c') (1 - \delta + z' f'(k'))]. \quad (10)$$

Euler equation methods approximate policy functions for consumption $c = C(k, z)$, capital $k' = K(k, z)$ (or other policy functions) to satisfy (2), (3) and (10). Similarly to ECM, Euler equation methods do not solve for a value function but only for decision (policy) functions. One of possible decision functions is a derivative of value function. Thus, recursion (9) can be also viewed as an Euler equation written in terms of the derivative of value function.

2.3 Convergence properties of the ECM methods

We now study the convergence of the ECM methods. For the expositional convenience, our analysis is limited to the optimal growth model (1)–(3), however, it can be readily extended to other models that satisfy the standard assumptions such as compactness and convexity of the budget set, smoothness and strong concavity of the utility function and the interiority of solutions; see Stokey and Lucas (1989) and Santos (1999, 2000) for a discussion of these assumptions, and see Stachursky (2009) for weaker assumptions.

2.3.1 ECM-VF

We first consider an ECM-VF method described in Section 2.2.1. We define the regular Bellman operator T for the model (1)–(3) as:

$$TV(k, z) \equiv u(c_{V_T}(k, z)) + \beta E [V(zf(k) - c_{V_T}(k, z), z')], \quad (11)$$

$$c_{V_T}(k, z) : u'(c) = \beta E [V_k(zf(k) - c, z')], \quad (12)$$

where $c_{V_T}(k, z)$ is the consumption function defined implicitly by FOC (4). (To simplify the notations, we assume $\delta = 1$ in this section).

We next introduce an operator Q that corresponds to the ECM-VF recursion

$$QV(k, z) \equiv u(c_{V_Q}(k, z)) + \beta E [V(zf(k) - c_{V_Q}(k, z), z')], \quad (13)$$

$$c_{V_Q}(k, z) : V_k(k, z) = u'(c) z f'(k), \quad (14)$$

where $c_{V_Q}(k, z)$ is the consumption function defined implicitly by envelope condition (5).

The operators T and Q are different but any fixed point of T is also a fixed point of Q and vice versa. (This fact is well illustrated by formula (7) that shows that imposing FOC enforces the envelope condition and vice versa).

Proposition 1 $V^* = TV^*$ iff $V^* = QV^*$.

Proof. If V^* is a fixed point of T , then assuming interiority, both FOC (4) and envelope condition (5) hold for the same V^* , in which case these two equations identify the same policy function, i.e., $c_{V_T}(k, z)$ and $c_{V_Q}(k, z)$ are equal for all k, z . Comparing T and Q , we have $TV^* = QV^*$. But $V^* = TV^*$, so $V^* = QV^*$ also holds. These results together with the fact that V^* is a fixed point of T yield that V^* is a fixed point of Q and establish the claim of Proposition 1. ■

It is well known that the regular Bellman operator T is a contraction mapping and thus, it guarantees a convergence to a fixed point V^* starting from an arbitrary initial guess V , i.e., $T^n V \rightarrow V^*$ as $n \rightarrow \infty$. Let us recall the proof.

Let V and W be two bounded continuous functions. Then, we have

$$|TV(k, z) - TW(k, z)| = \left| \max_c \{u(c) + \beta E[V(zf(k) - c, z')]\} - \max_c \{u(c) + \beta E[W(zf(k) - c, z')]\} \right|.$$

Using the property of a maximum operator, we obtain

$$|TV(k, z) - TW(k, z)| \leq \max_c |\{u(c) + \beta E[V(zf(k) - c, z')]\} - \{u(c) + \beta E[W(zf(k) - c, z')]\}|.$$

Taking the supremum on the left-hand side shows that T is a contraction mapping with a modulus β ,

$$\|TV - TW\| \leq \beta \|V - W\|, \tag{15}$$

where here and further in the text $\|\cdot\|$ is used to denote the supremum L_∞ norm.

We next ask: Is the ECM-VF operator Q a contraction mapping like T , or in other words, does Q guarantee a convergence to a fixed point $Q^n V \rightarrow V^*$ as $n \rightarrow \infty$?

To answer this question, we repeat the above argument with the ECM-VF operator:

$$\begin{aligned}
|QV(k, z) - QW(k, z)| = & \\
& |u(c_V(k, z)) + \beta E[V(zf(k) - c_V(k, z), z')] \\
& - u(c_W(k, z)) - \beta E[W(zf(k) - c_W(k, z), z')]|, \quad (16)
\end{aligned}$$

where $c_V(k, z)$ and $c_W(k, z)$ are consumption functions generated by the envelope conditions of V and W , respectively,

$$\begin{aligned}
c_V(k, z) & : V_k(k, z) = u'(c)zf'(k), \\
c_W(k, z) & : W_k(k, z) = u'(c)zf'(k). \quad (17)
\end{aligned}$$

By using a triangular inequality and by taking a supremum of (16), we arrive at

$$\|QV - QW\| \leq \|u(c_V) - u(c_W)\| + \beta \|V - W\|. \quad (18)$$

Formula (18) for the ECM operator Q has a new term $\|u(c_V) - u(c_W)\|$ compared to a similar formula (15) for the regular Bellman operator T . This term can be potentially large if the derivatives of V and W are very different, and hence, the ECM operator Q is not necessarily a contraction mapping.

Why do the Bellman and ECM operators have different convergence properties? A solution to envelope condition (5) does not maximize the right side of the Bellman equation for any V that occurs in iteration but only for a limiting fixed-point solution V^* . Since ECM does not produce a maximum of the Bellman equation in each iteration, it is not possible to cancel $\|u(c_V) - u(c_W)\|$ as is possible to do under the regular Bellman operator. (Similarly, the regular Bellman operator, enforces FOC (4) in each iteration but does not enforces envelope condition (5); it only enforces such a condition in the limiting fixed point V^*).

Formula (18) shows that when the derivative V_k is fixed, the ECM-VF operator has the same convergence properties as the regular Bellman operator including a linear convergence rate β . In particular, if we know the derivative V_k^* that corresponds to the true solution V^* (for example, we solve for V_k^* using the ECM-DVF method described in Section 2.2.2), then we have that $V_k = W_k = V_k^*$, the term $\|u(c_V) - u(c_W)\|$ disappears, and (18) is identical to (15). If we fix some arbitrary V_k , the ECM operator still converges as long as we maintain V_k fixed. However, in the latter case, ECM does not converge to the true value function V^* but to some other function that corresponds to fixed V_k . This reasoning suggests that one can implement the ECM-VF method (18) using two nested loops: in the inner loop, we solve for V holding V_k fixed, and in the outer loop, we gradually update V_k to match the derivative V constructed in the inner loop, iterating until convergence.

In fact, to guarantee a contraction-mapping property of the ECM operator, the term $\|u(c_V) - u(c_W)\|$ does not need to be zero but just needs to be sufficiently small. Hence, another possible implementation of the ECM-VF method is update the derivative V_k slowly enough to keep $\|u(c_V) - u(c_W)\| < \beta \|V - W\|$ along iterations (instead of the outer and inner loops). We attain this by introducing iteration with damping. To be specific, let us define a version of the ECM operator Q_ξ such that

$$Q_\xi V(k, z) \equiv u\left(c_{V_{Q_\xi}}(k, z)\right) + \beta E \left[V^{(i)}\left(zf(k) - c_{V_{Q_\xi}}(k, z), z'\right) \right], \quad (19)$$

$$c_{V_{Q_\xi}}(k, z) : (1 - \xi) V_k^{(i-1)}(k, z) + \xi V_k^{(i)}(k, z) = u'(c) z f'(k), \quad (20)$$

where $V^{(i-1)}$ and $V^{(i)}$ are value functions obtained on iterations $i - 1$ and i , respectively, $i \geq 2$, and $\xi \in (0, 1)$ is a damping parameter. That is, on each iteration, we update V entirely but we update V_k only partially (the algorithm ECM-VF described in Section 2.2.1 must be modified respectively).

The following proposition shows that the ECM-VF method with damping is a contraction mapping if ξ is sufficiently small.

Proposition 2 *There exists $\xi \in (0, 1)$ such that*

$$\|Q_\xi V - Q_\xi W\| < \|V - W\|. \quad (21)$$

Proof. Let us fix $V^{(i-1)}$ and hence $V_k^{(i-1)}$, obtained on iteration $i - 1$, and consider $\left\| u\left(c_{V_{Q_\xi}}\right) - u\left(c_{W_{Q_\xi}}\right) \right\|$, where $c_{V_{Q_\xi}}$ is given by (20) and $c_{W_{Q_\xi}}$ is defined as follows

$$c_{W_{Q_\xi}}(k, z) : (1 - \xi) V_k^{(i-1)}(k, z) + \xi W_k^{(i)}(k, z) = u'(c) z f'(k). \quad (22)$$

By construction, this term can be made arbitrarily small by reducing the value of ξ (in the limit, $\xi = 0$, this term is zero since $c_{V_{Q_\xi}} = c_{W_{Q_\xi}}$). In particular, for some $\xi \in (0, 1)$, we have $\left\| u\left(c_{V_{Q_\xi}}\right) - u\left(c_{W_{Q_\xi}}\right) \right\| < (1 - \beta) \|V - W\|$. Then, formula (18) implies

$$\begin{aligned} \|QV - QW\| &\leq \left\| u\left(c_{V_{Q_\xi}}\right) - u\left(c_{W_{Q_\xi}}\right) \right\| + \beta \|V - W\| \\ &< (1 - \beta) \|V - W\| + \beta \|V - W\|. \end{aligned}$$

This last results establishes the claim of Proposition 2. ■

Proposition 2 implies that the convergence rate of the ECM-VF operator with damping Q_ξ in particular depends on damping parameter ξ : the smaller is ξ , the slower is the convergence rate. Hence, we must use the largest possible value of ξ that is consistent with the condition $\|u(c_V) - u(c_W)\| < \beta \|V - W\|$. Some experimentation may be needed for finding such a value of ξ . The convergence rate of the ECM-VF with damping can be considerably smaller than the convergence rate of the regular Bellman operator T . Still, one iteration of Q_ξ is far less expensive than one iteration of T , and significant saving in cost may be possible, especially, for problems with high dimensionality.

2.3.2 ECM-DVF

We next turn to our second method ECM-DVF described in Section 2.2.2. We introduce the operator D that corresponds to the ECM-DVF recursion as follows:

$$DV_k(k, z) \equiv z f'(k) \beta E [V_k(z f(k) - c_V(k, z), z')] \quad (23)$$

$$c_V(k, z) : V_k(k, z) = u'(c) z f'(k). \quad (24)$$

We investigate the convergence properties of the ECM-DVF operator as we did for ECM-VF. Let V_k and W_k be two bounded continuous functions. Then, we have

$$\begin{aligned} |DV_k(k, \theta) - DW_k(k, \theta)| = \\ z f'(k) \beta |EV_k[(z f(k) - c_V(k, z), z')] - EW_k[(z f(k) - c_W(k, z), z')]| \end{aligned}$$

By taking a supremum, we obtain

$$\|DV_k - DW_k\| \leq z f'(k) \beta \|V_k - W_k\|. \quad (25)$$

To have a contraction mapping, we need the term $z f'(k) \beta$ that premultiplies $\|V_k - W_k\|$ to be smaller than 1 for all (k, z) . However, this is not the case: this term is equal to 1 in the steady state, and it is either smaller or larger than 1 depending on a specific state (k, z) considered.

Another possible way to prove that a mapping is a contraction is to show that it satisfies the Blackwell sufficiency conditions; see Santos (1999) and Stachursky (2009) for a discussion of these conditions. We find that the operator ECM-DVF possesses the property of monotonicity but not discounting which agrees with the result (25). (Curiously, for the previously considered operator ECM-VF (13), (14), we find exactly the opposite, namely, it possesses the property of discounting but fails to satisfy monotonicity).

Hence, our theoretical analysis indicates that ECM-DVF (23) is not a contraction mapping. Furthermore, in our numerical experiments, the ECM-DVF method was numerically unstable. To stabilize the ECM-DCF method, we again use iteration with damping. Namely, instead of a complete updating of V_k on each iteration, we introduce an operator D_ξ with a partial updating

$$D_\xi V_k(k, z) \equiv (1 - \xi) V_k(k, z) + \xi z f'(k) \beta E [V_k(z f(k) - c_V(k, z), z')], \quad (26)$$

where $\xi \in (0, 1)$. (In the algorithm ECM-DVF described in Section 2.2.2, we need to modify Step iii), respectively). With values of $\xi = 0.1$ and lower, operator D_ξ converges in all our numerical experiments. See Maliar, Maliar and Judd (2011) for a graphical illustration of how damping stabilizes explosive iteration.

2.3.3 Discussion

ECM-VF can be classified as a DP method. An important advantage of the DP class of methods is that under the appropriate assumptions, their properties can be characterized analytically including their convergence rates, error bounds, numerical stability and computational complexity; see Stokey and Lucas (1989), Santos (1999) and Stachursky (2009) for reviews of formal results for such methods. We establish that ECM-VF with damping is a contraction mapping like the regular Bellman operator.

In turn, ECM-DVF is similar to an Euler-equation class of methods. For such methods, formal results are harder to obtain and even their convergence is in general not guaranteed. Effectively, we need to find a numerical solution to a system of non-linear equations. There are three approaches in the literature that are used to solve non-linear systems of equations, namely, fixed-point iteration, time iteration, and Newton-style solvers; see Maliar and Maliar (2014) for a discussion. Time iteration is a special kind of fixed-point iteration that mimics the Bellman operator: given a guess about decision functions for future variables, it finds the values of the current variables to update the guess, iterating until convergence. Time iteration is more numerically stable than other fixed-point iteration schemes, however, it is also more expensive; see Judd (1998, Ch. 16) for a discussion. Moreira and Maldonado (2003) show a variant of time-iteration method for deterministic problems that is a contraction mapping. The idea is to construct a sequence of subiterations on the Euler equation by exploiting a local saddle-path stability of the system. Another paper that shows convergence results for their Euler equation method is Feng, Miao, Peralta-Alva and Santos (2009).

Our ECM-DVF method is not related to a specific iterative procedure for finding a fixed point and is compatible with all the procedures discussed above. In our nu-

merical experiments, we use fixed-point iteration with damping of type (26) because it is simple, inexpensive and reliable, however, we could have used time iteration or Newton-style solvers instead. For a version of ECM-DVF method based on time iteration, we can possibly show (local) convergence by using a construction similar to the one in Moreira and Maldonado (2003). However, the latter paper is limited to deterministic settings. Generalizing their analysis to a stochastic case is a non-trivial task and it goes beyond the scope of the present paper. We leave this extension for further research.

2.4 Numerical analysis

We now present the results of numerical experiments for the one-agent model.

2.4.1 Computational choices

We parameterize the model (1)–(3) using the Cobb-Douglas utility and production functions $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$ and $f(k) = k^\alpha$, and we calibrate the parameters to the standard values: $\alpha = 1/3$, $\beta = 0.99$, $\delta = 0.025$, $\rho = 0.95$ and $\sigma = 0.01$. In the benchmark case, we use $\gamma = 1$, and we perform the sensitivity experiments with respect to this parameter. As a solution domain, we use a rectangular, uniformly spaced grid of 10×10 points for capital and productivity within an ergodic range (to determine such a range we solve and simulate the model several times). We use a 10-node Gauss-Hermite quadrature rule for approximating integrals. We parameterize value function (ECM-VF) or a derivative of value function (ECM-DVF) with complete ordinary polynomials of degrees up to 5. As an initial guess, we use a linear approximation to the capital policy function. To solve for the polynomial coefficients, we use fixed-point iteration. We use MATLAB software, version 7.6.0.324 (R2012a) and a desktop computer ASUS with Intel(R) Core(TM)2 Quad CPU Q9400 (2.66 GHz), RAM 4MB. A detailed description of the algorithms is provided in Appendix A.

2.4.2 Results

In Table 1, we show the results for the two algorithms, ECM-VF and ECM-DVF. As a measure of accuracy, we report the average and maximum absolute unit-free residuals in Euler equation (10). The main result is that both ECM-VF and ECM-DVF deliver high accuracy levels.⁴ The accuracy increases with a degree of approximating

⁴We also tried out versions of ECM that approximate value function jointly with its derivatives. The results are nearly identical to those obtained when we approximate just the derivatives of value function.

Table 1: Accuracy and speed of ECM-VF and ECM-DVF in the one-country model.^a

Polynomial degree	ECM-VF			ECM-DVF		
	L ₁	L _∞	CPU	L ₁	L _∞	CPU
1st	-	-	-	-3.39	-3.24	7.5
2nd	-3.65	-3.42	0.7	-4.64	-4.21	5.0
3rd	-4.83	-4.39	0.5	-5.68	-5.19	2.9
4th	-5.96	-5.36	0.3	-6.83	-6.18	3.0
5th	-7.12	-6.43	0.3	-8.01	-7.32	1.6

^a Notes: L₁ and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in seconds).

polynomial. ECM-VF is less accurate than ECM-DVF given the same degree of approximating polynomial. This is because if we approximate V with a polynomial of some degree, we effectively approximate V_k with a polynomial of one degree less, i.e., we "lose" one polynomial degree. When γ increases (decreases), the accuracy of solutions decreases (increases); these cases are not reported. For example, under $\gamma = 3$ ($\gamma = 1/3$), the residuals for ECM-VF vary with the polynomial degree from -3.2 to -6.04 (from -3.83 to -7.51). For ECM-DVF, the corresponding residuals vary from -2.98 to -6.63 (-3.59 to -8.44).

Finally, as we see from the table, the convergence of ECM-VF is faster than that of ECM-DVF. The observed difference in costs represents the difference in the number of iterations necessary for convergence. The reason why ECM-DVF needed more iterations to converge because it was less numerically stable than ECM-VF and we stabilized it using damping (26) with an updating rate of 10% per iteration (we borrow this technique from the Euler equation class of methods). In contrast, ECM-VF was stable without damping with an updating rate of 100% per iteration.

3 ECM in the multicountry model

We consider the model studied in the February 2011's Journal of Economic Dynamics and Control special issue on a comparison of solution methods. This is a stylized stochastic growth model with N heterogeneous agents (interpreted as countries). Each country is characterized by a capital stock and a productivity level, so that there are $2N$ state variables. In addition to a potentially large number of state variables, the model features endogenous labor-leisure choice, heterogeneity in fundamentals and adjustment cost for capital.

We view this model as a convenient framework for testing the performance of solution methods in problems with high dimensionality. Namely, by varying N , we can expand the size of the problem and assess how the running time and accuracy are affected by the number of state variables. Also, this model was solved by various computational methods which provides a useful benchmark for comparison.

3.1 The model

Each country $h = 1, \dots, N$ is populated by one (representative) consumer. A social planner maximizes a weighted sum of expected lifetime utilities of the consumers by solving the following problem

$$V(\mathbf{k}, \mathbf{z}) = \max_{\{c^h, \ell^h, (k^h)'\}_{h=1, \dots, N}} \left\{ \sum_{h=1}^N \tau^h u^h(c^h, \ell^h) + \beta E[V(\mathbf{k}', \mathbf{z}')] \right\} \quad (27)$$

$$\text{s.t. } \sum_{h=1}^N c^h = \sum_{h=1}^N \left[z^h f^h(k^h, \ell^h) - \frac{\phi}{2} k^h \left(\frac{(k^h)'}{k^h} - 1 \right)^2 + k^h - (k^h)' \right], \quad (28)$$

$$\ln(z^h)' = \rho \ln z^h + \sigma (\varepsilon^h)', \quad (29)$$

where E is the operator of conditional expectation; $c^h, \ell^h, k^h, z^h, u^h, f^h$ and τ^h are consumption, labor, capital, productivity level, utility function, production function and welfare weight of a country $h \in \{1, \dots, N\}$, respectively; $c^h, \ell^h, (k^h)' \geq 0$; $\beta \in [0, 1)$ is the discount factor; ϕ is the adjustment-cost parameter. In the process for productivity (29), $\rho \in (-1, 1)$ is the autocorrelation coefficient of the productivity level; $\sigma > 0$ determines the standard deviation of the productivity level; and $((\varepsilon^1)', \dots, (\varepsilon^N)')^\top \sim \mathcal{N}(0_N, \Sigma)$ is a vector of productivity shocks with $0_N \in \mathbb{R}^N$ being a vector of zero means and $\Sigma \in \mathbb{R}^{N \times N}$ being a variance-covariance matrix. Thus, we allow for the case when productivity shocks of different countries are correlated. Initial condition, $\mathbf{k} \equiv (k^1, \dots, k^N)$ and $\mathbf{z} \equiv (z^1, \dots, z^N)$, is given, and a prime on variables means their future values.

Again, we assume that the solution to DP problem (27)–(29) is interior and that value function V is differentiable. Hence, the planner's choices satisfy the FOCs and envelope condition, given, respectively, by

$$\beta E[V_{k^h}(\mathbf{k}', \mathbf{z}')] = \lambda \left[1 + \phi \cdot \left(\frac{(k^h)'}{k^h} - 1 \right) \right], \quad (30)$$

$$\tau^h u_c^h(c^h, \ell^h) = \lambda, \quad (31)$$

$$u_\ell^h(c^h, \ell^h) \tau^h = -\lambda z^h f_\ell^h(k^h, \ell^h), \quad (32)$$

$$V_{k^h}(\mathbf{k}, \mathbf{z}) = \lambda \left[1 + z^h f_k^h(k^h, \ell^h) + \frac{\phi}{2} \left(\left(\frac{(k^h)'}{k^h} \right)^2 - 1 \right) \right], \quad (33)$$

where λ is the Lagrange multiplier.

3.2 Envelope condition method

In the multicountry case, we implement versions of the ECM methods that perform policy function iteration instead of value function iteration. This is because, operationally, it is easier to solve for a value function given policy function than to solve for policy functions given value function.

We first eliminate λ by combining FOC (30) and envelope condition (33),

$$1 = \frac{\beta E[V_{k^h}(\mathbf{k}', \mathbf{z}')] [\pi^h + z^h f_k^h(k^h, \ell^h)]}{V_{k^h}(\mathbf{k}, \mathbf{z}) o^h}, \quad (34)$$

where o^h and π^h are given by

$$o^h \equiv 1 + \phi \left(\frac{(k^h)'}{k^h} - 1 \right) \quad \text{and} \quad \pi^h = 1 + \frac{\phi}{2} \left(\left(\frac{(k^h)'}{k^h} \right)^2 - 1 \right).$$

Condition (34) relates today's and tomorrow's derivatives of the value function. We next use (34) to parameterize capital policy functions, namely, we premultiply both sides of (34) with $(k^h)'$ to obtain

$$(k^h)' = \frac{\beta E[V_{k^h}(\mathbf{k}', \mathbf{z}')] [\pi^h + z^h f_k^h(k^h, \ell^h)]}{V_{k^h}(\mathbf{k}, \mathbf{z}) o^h} (k^h)'. \quad (35)$$

The optimal capital policy functions, $(k^h)' = K^h(\mathbf{k}, \mathbf{z})$, $h = 1, \dots, N$, must satisfy a fixed point property: if we substitute such functions in the right side of (35), we must obtain the same functions in the left side. Conditions (35) for $h = 1, \dots, N$ provide us with a way to implement fixed-point iteration on capital policy functions. Namely, we guess some policy functions $K^h(\mathbf{k}, \mathbf{z})$, $h = 1, \dots, N$, substitute them in the right side of (35), recompute $(k^h)'$ in the left side and iterate on these steps until convergence.

Parameterization (35) is analogous to the one used in Maliar, Maliar and Judd (2011) to reparameterize the Euler equations in model (27)–(29),

$$(k^h)' = E \left\{ \frac{\beta u_c^h \left((c^h)', (\ell^h)' \right) \left[(\pi^h)' + (z^h)' f_k^h \left((k^h)', (\ell^h)' \right) \right]}{u_c^h (c^h, \ell^h)} \frac{1}{o^h} \right\} (k^h)'. \quad (36)$$

This kind of representation of Euler equations was originally used in the context of Monte Carlo based solution methods in which parameterizing expectation functions in canonical Euler equations do not identify all model's variables: see Den Haan (1990) and Marcet and Lorenzoni (1999) for related examples. The identification of variables is not an issue for solution methods like ECM that builds on deterministic integration techniques. However, solving nonlinear systems of equations (34) can be a non-trivial and costly task, especially, when the dimensionality is large. In contrast, fixed point iteration schemes like (35) and (36) are straightforward to implement; again, only direct calculations are needed.

3.2.1 ECM-VF

The fixed-point problem for the ECM-VF method in the multicountry model is similar to that in the one-country case except that we use policy function iteration instead of value function iteration.

ECM-VF

Given $K^h(\mathbf{k}, \mathbf{z})$, $h = 1, \dots, N$ for each (\mathbf{k}, \mathbf{z}) :

- i). Compute $(k^h)' = K^h(\mathbf{k}, \mathbf{z})$, $h = 1, \dots, N$.
- ii). Find (\mathbf{c}, ℓ) satisfying (28), (31) and (32) for each given $(\mathbf{k}, \mathbf{z}, \mathbf{k}')$.
- iii). Find V satisfying $V(\mathbf{k}, \mathbf{z}) = \sum_{h=1}^N \tau^h u^h(c^h, \ell^h) + \beta E[V(\mathbf{k}', \mathbf{z}')]$.
- iv). Use V to find $V_{k^h}(\mathbf{k}, \mathbf{z})$ and to infer future values $V_{k^h}(\mathbf{k}', \mathbf{z}')$, $h = 1, \dots, N$.
- v). Compute $(\widehat{k}^h)' = \frac{\beta E[V_{k^h}(\mathbf{k}, \mathbf{z})]}{V_{k^h}(\mathbf{k}, \mathbf{z})} \frac{[\pi^h + z^h f_k^h(k^h, \ell^h)]}{o^h} (k^h)'$, $h = 1, \dots, N$.

The optimal policy functions satisfy $(\widehat{k}^h)' = (k^h)'$, $h = 1, \dots, N$.

In step ii), we need to compute (\mathbf{c}, ℓ) satisfying (28), (31) and (32) given $(\mathbf{k}, \mathbf{z}, \mathbf{k}')$. This requires us to solve a system of $2N + 1$ equations with $2N + 1$ unknowns (\mathbf{c}, ℓ) and λ . This system can be solved with a standard Newton's style numerical solver but the cost of such a solver may become prohibitive when the dimensionality of the problem increases. Maliar et al. (2011) show a derivative-free iteration-on-allocation solver that can be used in this context and that can be vectorized for speed.

3.2.2 ECM-DVF

The fixed-point problem of the ECM-DVF method for the multicountry case also relies on policy function iteration.

ECM-DVF

Given $K^h(\mathbf{k}, \mathbf{z})$, $h = 1, \dots, N$ for each (\mathbf{k}, \mathbf{z}) :

- i). Compute $(k^h)' = K^h(\mathbf{k}, \mathbf{z})$, $h = 1, \dots, N$.
- ii). Find (\mathbf{c}, ℓ) satisfying (28), (31) and (32) for each given $(\mathbf{k}, \mathbf{z}, \mathbf{k}')$.
- iii). Find $V_{k^h}(\mathbf{k}, \mathbf{z}) = u_c^h(c^h, \ell^h) [\pi^h + z^h f_k^h(k^h, \ell^h)]$, $h = 1, \dots, N$.
- iv). Use V_{k^h} to infer future values $V_{k^h}(\mathbf{k}', \mathbf{z}')$, $h = 1, \dots, N$.
- v). Compute $(\widehat{k}^h)' = \frac{\beta E[V_{k^h}(\mathbf{k}, \mathbf{z})]}{V_{k^h}(\mathbf{k}, \mathbf{z})} \frac{[\pi^h + z^h f_k^h(k^h, \ell^h)]}{o^h} (k^h)'$, $h = 1, \dots, N$.

The optimal policy functions satisfy $(\widehat{k}^h)' = (k^h)'$, $h = 1, \dots, N$.

Again, in step iii), we need to solve the same system of equation as under ECM-VF, i.e., to compute (\mathbf{c}, ℓ) satisfying (31), (32) and (28) given $(\mathbf{k}, \mathbf{z}, \mathbf{k}')$.

3.2.3 Making ECM tractable in high-dimensional problems

The ECM approaches focus on one specific issue, namely, on how to reduce the computational cost of solving for value function and its derivatives using the optimality conditions. However, to build a solution method, we need to specify other computational choices such as a grid for finding a solution, a function for approximations, an integration method, a fitting method, etc. Recent literature distinguished techniques that are tractable in high-dimensional applications in the context of Euler equation methods; these are non-product grids, low-cost accurate monomial integration rules, derivative-free solvers; see Krueger and Kubler (2004), Malin, Krueger and Kubler (2011), Pichler (2011), Maliar et al. (2011), Judd, Maliar and Maliar (2011, 2012, 2013); see also Maliar and Maliar (2014) for a review. The ECM methods are fully compatible with all these techniques.

We choose to implement ECM-VF and ECM-DVF following the design of *generalized stochastic simulation algorithm* (GSSA) method by Judd et. al. (2011). GSSA uses a set of points produced by stochastic simulation as a grid for finding a solution. In this sense, it is similar to simulation-based Euler equation and value function iteration methods introduced in Marcet (1988) and Maliar and Maliar (2005), respectively.⁵ However, GSSA differs from the latter methods in two respects: first,

⁵Marcet's (1988) method is developed in Den Haan and Marcet (1990) and Marcet and Lorenzoni (1999).

to insure numerical stability, it uses fitting methods that are suitable for dealing with ill-conditioned problems and second, to attain high accuracy of solutions, it uses non-stochastic (monomial and quadrature) integration rules. As a result, GSSA delivers accuracy levels that are comparable to the best accuracy attained in the related literature and that are infeasible for purely simulation methods; see Judd et al. (2011) for a discussion and numerical examples.

3.3 Numerical analysis

We now present the results of numerical experiments for the multicountry model.

3.3.1 Computational choices

We apply the ECM methods to solve Model II with an asymmetric specification; see the comparison analysis of Kollmann et al. (2011). We chose this model among others because it represents all challenges posed in the comparison analysis, namely, a large number of state variables, elastic labor supply, heterogeneity in fundamentals and the absence of closed-form expressions for next-period state and control variables.⁶ The utility and production functions are given by

$$u^h(c_t^h, \ell_t^h) = \frac{(c_t^h)^{1-1/\gamma^h}}{1-1/\gamma^h} - B^h \frac{(\ell_t^h)^{1+1/\eta^h}}{1+1/\eta^h}, \quad z f^h(k_t^h, \ell_t^h) = z^h A (k_t^h)^\alpha (\ell_t^h)^{1-\alpha} - \delta k^h, \quad (37)$$

where $\{\gamma^h, B^h, \eta^h\}$ are the utility-function parameters; α is the capital share in production; A is the normalizing constant in output; $\delta \in (0, 1]$ is the depreciation rate. We calibrate the model as in Kollmann et al. (2011b). We use the following values of common-for-all-countries parameters: $\alpha = 0.36$, $\beta = 0.99$, $\delta = 0.025$, $\sigma = 0.01$, $\rho = 0.95$, $\phi = 0.5$, and we assume that the country-specific utility-function parameters γ^h and η^h are uniformly distributed in the intervals $[0.25, 1]$ and $[0.1, 1]$ across countries $h = 1, \dots, N$, respectively. The steady state level of productivity is normalized to one, $\bar{z}^h = 1$. We also normalize the steady state levels of capital and labor to one, $\bar{k}^h = 1$, $\bar{\ell}^h = 1$, which implies $\bar{c}^h = A$, $\bar{\lambda} = 1$ and leads to $A = \frac{1-\beta}{\alpha\beta}$, $\tau^h = u^h(A, 1)$ and $B^h = (1-\alpha)A^{1-1/\gamma^h}$. We consider $N = 2, 4, 6$ and 8 .

We parameterize value function (ECM-VF) and the derivative of value function (ECM-DVF) with complete ordinary polynomials of degrees 2, 3 and 1, 2, 3, respec-

⁶Model I has a degenerate labor-leisure choice, and Models III and IV are identical to Model II up to specific assumptions about preferences and technologies. Juillard and Villemot (2011) provides a description of all models studied in the comparison analysis of Kollmann et al. (2011b).

tively. As an initial guess, we use a linear approximation to capital policy function. To solve for the polynomial coefficients, we use fixed-point iteration. To solve for consumption and labor satisfying (28), (31) and (32), we use an iteration-on-allocation solver developed in Maliar et al. (2011). To approximate integrals, we use a monomial integration rule $M1$ with $2N$ nodes, and to fit the value and policy functions to simulated data, we use a least-squares truncated QR factorization method; see Judd et al. (2011) for a description of these techniques. We use the same software and hardware as that used to solve the one-country model. We provide a detailed description of the studied ECM methods in Appendix B.

3.3.2 Results

In Table 2, we present the results produced by two versions of the ECM method, ECM-VF that solves for value function and ECM-DVF that solves for derivative of value function. Below, we report accuracy measures on a stochastic simulation, and in Appendix C, we also provide such measures on spheres of different radii used in the JEDC project; see Juillard and Villemot (2011). Our main finding is that the

Table 2: Accuracy and speed of ECM-VF and ECM-DVF in the one-country model.^a

Default rule	ECM-VF			ECM-DVF			VFI		
	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
$\bar{y}(b) = 0.65 - b$	-3.85	-3.32	5.41	-3.00	-2.92	5.32	-3.85	-2.83	412.24
$\bar{y}(b) = 0.75 - b$	-3.86	-3.34	5.48	-3.00	-2.92	5.34	-3.86	-3.34	282.24
$\bar{y}(b) = 0.95 - b$	-3.85	-3.40	5.39	-2.99	-2.93	5.29	-3.48	-3.00	295.13

^a Notes: L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in seconds).

ECM methods are tractable in the context of the given multidimensional problem. Moreover, the ECM methods are able to produce not only the second-degree but also far more expensive third-degree polynomial approximations. All Euler equation methods studied in Kollmann et al. (2011b) are limited to second-degree polynomial approximation. The ECM methods have an advantage over Euler equation methods in that they solve for control variables only at present and do not need to find such variables in all integration nodes. This advantage can be especially important in high-dimensional problems as the number of integration nodes grows with dimensionality.

As far as the accuracy is concerned, ECM-VF is considerably less accurate than ECM-DVF. Our results suggest that in high-dimensional problems, approximating

value function with a polynomial on a grid does not produce accurate approximations for derivatives of the value function. This is the same effect that we observed in Section 2.4.2 for the one-agent model, namely, if we approximate V with a polynomial, we effectively approximate V_k with a polynomial of one degree less, i.e., we "lose" one polynomial degree.

In turn, the ECM-DVF method is very accurate. It delivers accuracy frontier attained in the comparison analysis of Kollmann et al. (2011b). In particular, in an accuracy check on a stochastic simulation, our third-degree solutions are more accurate than second-degree polynomial solutions reported in Kollmann et al. (2011b) although our second-degree polynomial solutions are somewhat less accurate than their most-accurate solutions. For example, for a model with $N = 8$ countries, the second- and third-degree ECM-DVF polynomial solutions have maximum residuals across the optimality conditions of orders $10^{-4.16}$ and $10^{-4.71}$, respectively. For comparison, the most accurate second-degree method in Kollmann et al. (2011) produces maximum residual of order $10^{-4.50}$. Thus, we conclude that ECM value iteration methods can successfully compete with the state-of-the-art Euler equation methods.⁷

4 ECM for default risk models

Default risk models focus on borrowing-lending arrangements in which debt is unsecured and a borrower can default on debt. Examples of situations with default include sovereign default (e.g, Greek default of 2012, Argentinian default of 2001), consumer bankruptcy (defaults on loans and mortgages), firm bankruptcy (defaults on financial or contractual obligations), local government defaults (e.g., Detroit in 2013), etc.

The recent financial and sovereign debt crises worldwide has sparked a growing literature on quantitative models of defaultable debt. Arellano (2008) studied quantitatively the implications of the seminal paper by Eaton and Gersovitz (1983) and showed that it was useful for understanding sovereign default in emerging markets. Aguiar and Gopinath (2006) showed the importance of shocks to trend for output in emerging economies in the context of a sovereign default model. Chatterjee, Corbae, Nakajima and Ríos-Rull (2007) provided a framework to study consumer

⁷The only method (apart from ours) that has produced third-degree polynomial solutions to a similar model is a perturbation-based hybrid Euler equation method of Maliar et al. (2011). This method computes some policy functions locally (using perturbation) and computes the remaining policy functions globally (using analytical formulas and numerical solvers). In the given model with $N = 8$ countries, this method delivers maximum residuals of order $10^{-4.69}$.

bankruptcy in the United States. Their model can rationalize the cross section distribution of bankruptcies across households of different characteristics. Maliar, Maliar and Pérez-Sebastián (2008) construct a default risk model of FDI and capital controls; they argue that scarce capital flow from rich to poor nations can be explained by a risk of expropriation. Arellano, Bai, and Kehoe (2013) studied the implications of firm default for business cycles and for the Great Recession in the United States. Tsyrennikov (2013) analyzed optimal fiscal and default policy in default risk models. See Aguiar and Amador (2013) for a review of the literature on sovereign debt.

A main challenge for this literature, however, is the computational burden of models with default. Computational limitations constitute a substantial obstacle to analyze richer models. Here, we apply the ECM methods to a simple default model and show that it is reliable and efficient.

4.1 A default risk model

We study a variant of the default risk model of Arellano (2008). A country borrower may decide to default when the debt is getting too large and or when facing large negative shock.

A borrower's problem. A country-borrower is populated by a representative household with preferences $E_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$ where u is strictly increasing, continuously differentiable and concave and $\beta \in (0, 1)$. The borrower receives exogenous stochastic income y_t which follows an AR1 process

$$\log(y_t) = \rho \log(y_{t-1}) + \varepsilon_t, \quad (38)$$

with $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, $\rho \in (-1, 1)$, and $\sigma \geq 0$.

The borrower trades one period bonds with international lenders and can default on the bonds. When the borrower has bonds b_t , income y_t , and does not default, it can choose new bond b_{t+1} at price $q(b_{t+1}, y_t)$. Consumption in this case is

$$c_t = y_t + b_t - q(b_{t+1}, y_t)b_{t+1}$$

A negative value of b means that the country issues bonds to borrow; $q(b_{t+1}, y_t)$ is the price that a borrower will pay for a unit bond depending on the quantity of bonds issued by the country b_{t+1} and its current state y_t . These variables determine the probability of default in the next period. The borrower takes as given the bond price function.

Default decision. The borrower can default at any time on the debt b_t it owes and pay 0. We assume that the borrower's default decision is exogenously given by a function $\bar{y}(b_t)$ such that an agent with the debt b_t will default whenever the random income y_t falls below the threshold level $y_t < \bar{y}(b_t)$.

By using (38), we can compute the probability of default $\delta(b_{t+1}, y_t)$ at $t + 1$

$$\delta(b_{t+1}, y_t) = \text{prob}(\underbrace{y_t^\rho \exp(\varepsilon_t)}_{y_{t+1}} < \bar{y}(b_{t+1})) = F \left[\ln \left(\frac{\bar{y}(b_{t+1})}{y_t^\rho} \right) \right] \quad (39)$$

where F is a cumulative distribution function on a normal distribution.

Lenders' problem and bond-price function. Lenders are risk neutral and perfectly competitive. Their problem is

$$\max_{b_{t+1}} \left\{ q_t b_{t+1} - \frac{1 - \delta_t}{1 + r} b_{t+1} \right\}, \quad (40)$$

where r is a risk-free interest rate, and δ_t is a probability of a borrower to default. A zero-profit condition implies that $q_t = \frac{1 - \delta_t}{1 + r}$. If $\delta_t = 0$ (i.e., a borrower never defaults) then $q_t = \frac{1}{1 + r}$ (risk-free interest rate) and if $\delta_t = 1$ (i.e., a borrower always defaults) then $q_t = 0$ (bonds are worthless). If a borrower defaults with some probability $\delta_t \in (0, 1)$, then we have $q_t \in (0, \frac{1}{1 + r})$. Using (39), we can represent the price function $q(b_{t+1}, y_t)$ by

$$q(b_{t+1}, y_t) = \frac{1}{1 + r} \left(1 - F \left[\ln \left(\frac{\bar{y}(b_{t+1})}{y_t^\rho} \right) \right] \right). \quad (41)$$

The probability of default $q(b_{t+1}, y_t)$ increases with the amount of debt b_{t+1} , and it decreases with income y_t .

A recursive formulation. To formulate a Bellman equation for the consumer's problem, we introduce two value functions $V^d(y)$ and $V(b, y)$ that correspond to default and no-default states. To decide whether to default or not, an agent compare these two possibilities and chooses the one that implies higher welfare,

$$V^o(b, y) \equiv \max \{ V(b, y), V^d(y) \}. \quad (42)$$

However, by assumption the borrower chooses default if $V(b, y) < V(b, \bar{y}(b)) \equiv V^d(y)$. If a borrower does not default, his value function V satisfies

$$\begin{aligned} V(b, y) &= \max_{b'} \left\{ u(c) + \beta \int \max [V(b', y'), V(b', \bar{y}(b'))] dF(y') \right\} \\ \text{s.t. } c &= y + b - q(b', y) b', \end{aligned} \quad (43)$$

where F is a distribution function of y' .

Relation to Arellano's (2008) model. In the model of Arellano (2008), a condition for default is defined implicitly by (42): a borrower defaults for those y for which $V(b, y) < V^d(y)$ given b . If a borrower defaults, he goes to autarky and gets

$$V^d(y) = u(y^{def}) + \beta E [\theta V^o(0, y') + (1 - \theta) V^d(y')], \quad (44)$$

where z is probability of re-incorporating in the world economy after default, and $y^{def} \leq y$ are direct output costs from defaulting. By changing z and y^{def} , we can affect $V^d(y)$ and hence, the borrower's incentives to default.

For the case of i.i.d. shocks and when the cost of default is limited to exclusion from the borrowing market, Arellano (2008) showed that the default decision is a cutoff rule of type $\bar{y}(b)$. In quantitative simulations for more general shock process and default costs that paper also contains default decisions that are cutoff rules. Hence, our rule for default $\bar{y}(b)$ induces the corresponding value function condition $V(b, y) < V^d(y)$ and vice versa. In particular, if we take the cutoff rule $\bar{y}(b)$ that is implied by Arellano's (2008) analysis, we will get the same solution as in that paper.

The difference of our analysis from Arellano's (2008) is that we define $\bar{y}(b)$ explicitly and directly. That is, we take the cutoff rule $\bar{y}(b)$ as a primitive and solve for the corresponding value function condition (42), whereas Arellano (2008) takes (42) as a primitive and solves for the corresponding cutoff rule. For those applications in which default decision rules do not have the form of well-defined cutoff rules, we need to distinguish regions in which $V(b, y) \geq V^d(y)$ and construct the corresponding default rules iteratively.

4.2 Envelope condition method

Assuming differentiability of the price and value functions, in those states in which default does not occur $y \geq \bar{y}(b)$, the quantity of issued bonds b' satisfies the following first-order condition:

$$u'(c)[q_b(b', y)b' + q(b', y)] = \beta E [V_b^o(b', y')]. \quad (45)$$

The envelope condition, $V_b(b, y) = u'(c)$, in turn implies the following ECM-DVF recursion:

$$V_b(b, y) = \frac{\beta E [V_b^o(b', y')]}{q_b(b', y)b' + q(b', y)} = \frac{\beta \int V_b(b', y') \mathbf{1}(y' > \bar{y}(b)) dF(y')}{q_b(b', y)b' + q(b', y)}. \quad (46)$$

where $1(X)$ is an indicator of an event X , and F is a distribution function of y' . The term corresponding to the indicator function $1(y' \leq \bar{y}(b'))$ does not appear in (46) because for $y' \leq \bar{y}(b')$, we have $V_b^o(b', y') = \frac{\partial V^d(y')}{\partial b'} = 0$.

4.2.1 ECM-VF

We first show an ECM-VF fixed-point problem for the default risk model (42)-(43).

ECM-VF

Fix $\bar{y}(b)$ and choose a set of points (b, y) such that $y > \bar{y}(b)$.

Compute $q(b', y)$ using (41).

Define $\mathcal{L}(b) \equiv q(b)b$ and precompute its inverse \mathcal{L}^{-1} .

Given $V(b, y)$, for each point (b, y) define:

i). $c = u'^{-1}[V_b(b, y)]$.

ii). $b' = \mathcal{L}^{-1}(b + y - c)$.

iii). $\widehat{V}(b, y) = u(c) + \beta E[\max\{V(b', y'), V(b', \bar{y}(b'))\}]$.

The optimal value function satisfies $\widehat{V} = V$.

While we can solve for b' satisfying (43) for each point (b, y) within the main iterative cycle, doing so would be costly because we need to use a nonlinear solver a large number of times. Precomputation – constructing a part of a numerical solution outside the main iterative cycle – can speed up computation greatly; see Maliar and Maliar (2014) for review of precomputation techniques for dynamic economic models.

4.2.2 ECM-DVF

We now show an ECM-DVF fixed-point problem for the default risk model (42)-(43).

ECM-DVF

Fix $\bar{y}(b)$ and choose a set of points (b, y) such that $y > \bar{y}(b)$.

Compute $q(b', y)$ using (41).

Define $\mathcal{L}(b) \equiv q(b)b$ and precompute its inverse \mathcal{L}^{-1} .

Given $V_b(b, y)$, for each point (b, y) , define:

i). $c = u'^{-1}[V_b(b, y)]$.

ii). $b' = \mathcal{L}^{-1}(b + y - c)$.

iii). $\widehat{V}_b(b, y) = \frac{\beta E[V_b^o(b', y')]}{q_b(b', y)b' + q(b', y)} = \frac{\beta \int V_b(b', y')1(y' > \bar{y}(b'))dF(y')}{q_b(b', y)b' + q(b', y)}$.

The optimal value function satisfies $\widehat{V}_b = V_b$.

When implementing ECM-DVF, one needs to be careful not to include grid points for which $q_b(b', y)b' + q(b', y) < 0$. As was shown in Arellano (2008), the amount

of resources that a country can borrow follows a Laffer curve. Initially, the loan $\mathcal{L}(b) \equiv q(b)b$ increases with b , then it reaches its maximum and finally, it decreases to zero because an increased risk of default quickly drives the bond price $q(b)$ to zero which dominates the product $q(b)b$; see Figure 1 for an example of the Laffer curve. A borrower can never be on a negatively-sloping portion of that Laffer curve.

4.3 Numerical analysis

We now construct numerical solutions for the default risk model.

4.3.1 Computational choices

As an example, we consider a simple default rule $y_t \leq \bar{y}(b_t) \equiv \underline{n} - b_t$, where \underline{n} is an exogenous lower bound on the borrower's net worth. The borrower with the mean income level $y_t = 1$ would default when its debt rises above $1 - \underline{n}$. We consider three values of $\underline{n} = \{0.65, 0.75, 0.95\}$ which imply a defaults roughly the borrower's debt rises above 0.35, 0.25 and 0.05 of an average period income respectively. We assume that the income in (38) is i.i.d, $y \sim N(0, 0.05)$. Then, the probability of default (39) is given by

$$\delta(b_{t+1}, y_t) = \text{prob}(y_{t+1} \leq \bar{y}(b_{t+1})) = F(\underline{n} - b_{t+1}).$$

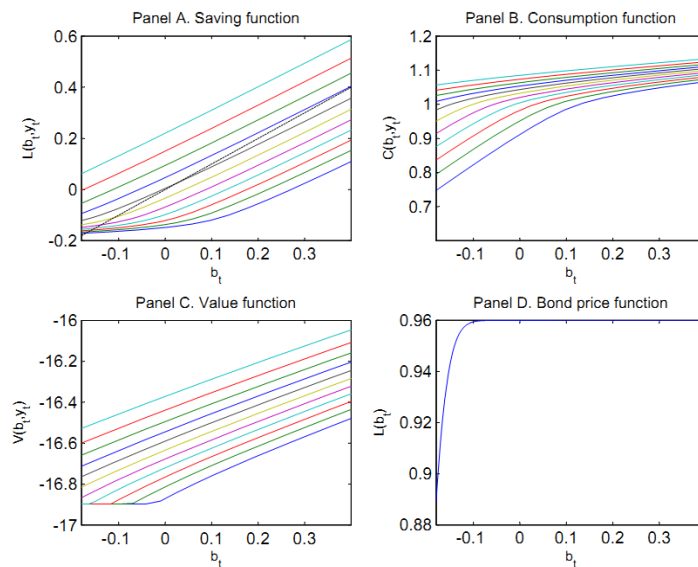
where F is a distribution function of a Normal distribution. We choose the remaining parameters in line with Arellano (2008), namely, we parameterize the utility function by $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$ with $\gamma = 2$, and we fixed $\beta = 0.94$.

We solve the model on $b \in [-0.18, 0.40]$. For each realization of output y we approximate the unknown value function and its derivative using a cubic spline with 21 nodes. Nodes are Chebyshev extrema scaled so that $b_1 = -0.18, b_{100} = 0.40$. To perform integration with respect to y' we use a Gauss-Hermite quadrature rule with 11 nodes. We implemented three algorithms: ECM-VF, ECM-DVF and conventional VFI which we use for the sake of comparison. ECM-VF was less stable numerically and required a sufficiently accurate initial guess for convergence ECM-VF is less stable numerically in the default risk model and requires a sufficiently accurate initial guess, while ECM-DVF was robust to the choice of an initial guess. For all three algorithms, we use an identical convergence criterion that the (maximum across nodes) change in the value function was smaller than 10^{-4} . All calculations are performed in MATLAB 2013b on a laptop with an 2.9GHz Intel i7-3520M processor. We provide a detailed description of the studied computational methods in Appendix D.

4.3.2 Numerical results

We show the constructed policy function in Figure 1.

Figure 1. A numerical solution to a default risk model.



As expected, bond issue policy function smoothly approaches the lower boundary. As panel A shows, the borrower always accumulates/decumulates assets in the highest/lowest income state. Panel B plots optimal consumption. It varies substantially with the borrower's assets, especially in low income states. The bond price function in panel C is simply a portion of a Normal distribution function. Panel D plots the value function $V^o(b, y)$. At low levels of assets it becomes flat: this is the interval where default is possible.

In Table 3, we report the running time and accuracy measures on a stochastic simulation produced by ECM-VF and ECM-DVF and we compare the results with those produced by a conventional VFI. The standard VFI takes about 282.2 seconds to converge, while the ECM-VF and ECM-DVF method take both about 5.5 seconds; hence, we observe about 50x speedups or higher. As in all previous numerical experiments, the accuracy of ECM-VF is somewhat lower than that of ECM-DVF as was in all previous experiments.

Table 3: Accuracy and speed of ECM-VF and ECM-DVF in the one-country model.^a

Default rule	ECM-VF			ECM-DVF			VFI		
	L ₁	L _∞	CPU	L ₁	L _∞	CPU	L ₁	L _∞	CPU
$\bar{y}(b) = 0.65 - b$	-3.85	-3.32	5.41	-3.00	-2.92	5.32	-3.85	-2.83	412.24
$\bar{y}(b) = 0.75 - b$	-3.86	-3.34	5.48	-3.00	-2.92	5.34	-3.86	-3.34	282.24
$\bar{y}(b) = 0.95 - b$	-3.85	-3.40	5.39	-2.99	-2.93	5.29	-3.48	-3.00	295.13

^a Notes: L₁ and L_∞ are, respectively, the average and maximum of unit-free absolute residuals in Bellman equation across test points (in log10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in seconds).

5 Conclusion

In the paper, we focus on a broad and empirically relevant class of DP problems characterized by a large, finite number of continuous state variables and a differentiable value function. There are three main challenges that such problems represent to numerical solution methods. First, the number of arguments in value and policy functions increases with the dimensionality of the problem and such functions become costly to approximate numerically. Second, the cost of integration increases as the number of exogenous random variables increases. Finally, larger problems are normally characterized by larger and more complex systems of equations which are more expensive to solve.

We develop ECM methods for DP problems that aim to address these challenges. Concerning the first two challenges, we build ECM on non-product approximation, integration and interpolation techniques that are designed for dealing with high-dimensional problems; see Maliar and Maliar (2014) for a review of such techniques. The last challenge is the main focus of our analysis, namely, we replace conventional backward VFI iteration with cheap forward iteration based on the envelope condition. We show that the computational expense of high-dimensional applications can be reduced even further by combining value and policy function iteration.

We find that solving for value function does not accurately identify the derivatives of value function. The accuracy of ECM can be significantly increased by solving for the derivatives of value function instead of the value function itself, or, alternatively, by solving jointly for value function and its derivatives. In the context of large-scale models studied in the JEDC project, the version of the ECM method that approximates derivatives of value function can successfully compete with the state-of-the-art Euler equation methods. Moreover, the ECM methods deliver accurate solutions to challenging default risk models with a kink in value and policy functions

and is faster by orders of magnitude than the conventional VFI in our examples.

References

- [1] Aguiar, M. and G. Gopinath (2006). Defaultable debt, interest rates and the current account. *Journal of International Economics*, 69(1), 64-83.
- [2] Aguiar, M. and M. Amador (2013). Sovereign Debt: A Review. NBER Working Papers 19388.
- [3] Arellano, C. (2008). Default risk and income fluctuations in emerging economies. *American Economic Review*, 98(3), 690–712.
- [4] Arellano, C., Y. Bai, and P. Kehoe (2013). Financial frictions and fluctuations in volatility. Federal Reserve Bank of Minneapolis Staff Report 466.
- [5] Aruoba, S. and F. Schorfheide, (2013). Macroeconomic dynamics near ZLB: a tale of two equilibria. NBER working paper 19248.
- [6] Aruoba, S., J. Fernández-Villaverde and J. Rubio-Ramírez, (2006). Comparing solution methods for dynamic equilibrium economies. *Journal of Economic Dynamics and Control* 30, 2477-2508.
- [7] Barillas, F. and J. Fernandez-Villaverde, (2007). A generalization of the endogenous grid method. *Journal of Economic Dynamics and Control*, Elsevier 31, 2698-2712.
- [8] Carroll, K. (2005). The method of endogenous grid points for solving dynamic stochastic optimal problems, *Economic letters* 91, 312-320.
- [9] Chatterjee, S. and B. Eyigungor, (2011). A quantitative analysis of the U.S. housing and mortgage markets and the foreclosure crisis. Working Papers 11-26, Federal Reserve Bank of Philadelphia.
- [10] Chatterjee, S., D. Corbae, M. Nakajima, and J. V. Ríos-Rull, (2007). A Quantitative Theory of Unsecured Consumer Credit with Risk of Default. *Econometrica* 75 (November): 1525-1589.
- [11] Den Haan, W. (1990). The optimal inflation path in a Sidrauski-type model with uncertainty. *Journal of Monetary Economics* 25, 389-409.

- [12] Den Haan, W. and A. Marcet, (1990). Solving the stochastic growth model by parameterized expectations. *Journal of Business and Economic Statistics* 8, 31-34.
- [13] Den Haan W., K. L. Judd and M. Juillard, (2011). Computational suite of models with heterogeneous agents II: Milticountry real business cycle models. *Journal of Economic Dynamics and Control* 35, 175–177.
- [14] Eaton, J., and M. Gersovitz (1981). Debt with Potential Repudiation: Theoretical and Empirical Analysis. *Review of Economic Studies*, 48(2): 289–309.
- [15] Fella, G., (2014). A generalized endogenous grid method for non-smooth and non-concave problems, *The Review of Economic Dynamics*, 17/2, 329-344.
- [16] Feng, Z., J. Miao, A. Peralta-Alva, and M. Santos, (2009). Numerical simulation of nonoptimal dynamic equilibrium models. Working papers Federal Reserve Bank of St. Louis 018.
- [17] Fernández-Villaverde, J., G. Gordon, P. Guerrón-Quintana, and J. Rubio-Ramírez, (2012). Nonlinear adventures at the zero lower bound. NBER working paper 18058.
- [18] Fukushima, K. and Y. Waki, (2011). A polyhedral approximation approach to concave numerical dynamic programming. Manuscript.
- [19] Gust, C., D. Lopez-Salido and M.E. Smith, (2012). The empirical implications of the interest-rate lower bound. Manuscript, Federal Reserve Board.
- [20] Hasanhodzic, J. and L. Kotlikoff, (2013). Generational risk – is it a big deal?: simulating an 80-period OLG model with aggregate shocks. NBER Working Paper 19179.
- [21] Ishakov, F., J. Rust and B. Schjerning, (2012). Extending endogenous grid method for solving discrete continuous sequential decision problems. Manuscript.
- [22] Judd, K., (1998). *Numerical Methods in Economics*. Cambridge, MA: MIT Press.
- [23] Judd, K., L. Maliar and S. Maliar, (2011). Numerically stable and accurate stochastic simulation approaches for solving dynamic models. *Quantitative Economics* 2, 173-210.

- [24] Judd, K., L. Maliar and S. Maliar, (2012). Merging simulation and projection approaches to solve high-dimensional problems. NBER working paper 18501.
- [25] Judd, K., L. Maliar, S. Maliar and R. Valero (2013). Smolyak Method for Solving Dynamic Economic Models: Lagrange Interpolation, Anisotropic Grid and Adaptive Domain, NBER 19326.
- [26] Judd, K., S. Yeltekin and J. Conklin, (2003). Computing supergame equilibria. *Econometrica*, 71, 1239-1254.
- [27] Juillard, M. and S. Villemot, (2011). Multi-country real business cycle models: accuracy tests and testing bench. *Journal of Economic Dynamics and Control* 35, 178–185.
- [28] Kollmann, R., S. Kim and J. Kim, (2011a). Solving the multi-country real business cycle model using a perturbation method. *Journal of Economic Dynamics and Control* 35, 203-206.
- [29] Kollmann, R., S. Maliar, B. Malin and P. Pichler, (2011b). Comparison of solutions to the multi-country real business cycle model. *Journal of Economic Dynamics and Control* 35, 186-202.
- [30] Krueger, D. and F. Kubler, (2004). Computing equilibrium in OLG models with production. *Journal of Economic Dynamics and Control* 28, 1411-1436.
- [31] Maliar, L. and S. Maliar, (2005). Solving nonlinear stochastic growth models: iterating on value function by simulations. *Economics Letters* 87, 135-140.
- [32] Maliar, L. and S. Maliar, (2013). Envelope condition method versus endogenous grid method for solving dynamic programming problems, manuscript.
- [33] Maliar, L. and S. Maliar, (2014). Numerical methods for large scale dynamic economic models. In: Schmedders, K., Judd, K. (Eds.), *Handbook of Computational Economics*, vol. 3, Elsevier Science, Amsterdam.
- [34] Maliar, L., S. Maliar and F. Pérez-Sebastián (2008). Sovereign Risk, FDI Spillovers, and Economic Growth. *Review of International Economics* 16/3, 463-477.
- [35] Maliar, S., L. Maliar and K. Judd, (2011). Solving the multi-country real business cycle model using ergodic set methods. *Journal of Economic Dynamic and Control* 35, 207–228.

- [36] Maliar, L., Maliar, S. and S. Villemot (2012). Taking perturbation to the accuracy frontier: a hybrid of local and global solutions. *Computational Economics*, forthcoming.
- [37] Malin, B., D. Krueger and F. Kubler, (2011). Solving the multi-country real business cycle model using a Smolyak-collocation method. *Journal of Economic Dynamics and Control* 35, 229-239.
- [38] Marcet, A., and G. Lorenzoni (1999). The parameterized expectation approach: some practical issues. In: R. Marimon and A. Scott (Eds.) *Computational Methods for Study of Dynamic Economies*. Oxford University Press, New York, pp. 143-171.
- [39] Moreira, H. and W. Maldonado (2003). A contractive method for computing the stationary solution to the Euler equation. *Economics Bulletin* 3, 1-14.
- [40] Pichler, P. (2011). Solving the multi-country real business cycle model using a monomial rule Galerkin method. *Journal of Economic Dynamics and Control* 35, 240-251.
- [41] Powell W., (2011). *Approximate Dynamic Programming*. Wiley: Hoboken, New Jersey.
- [42] Rust, J., (2008). Dynamic programming. In *The New Palgrave Dictionary of Economics* (S. Durlauf and L. Blume eds.), Palgrave Macmillan.
- [43] Santos, M., (1999). Numerical solution of dynamic economic models, in: J. Taylor and M. Woodford (Eds.), *Handbook of Macroeconomics*, Amsterdam: Elsevier Science, pp. 312-382.
- [44] Santos, M., (2000). Accuracy of numerical solutions using the Euler equation residuals. *Econometrica* 68, 1377-1402.
- [45] Stachursky, J., (2009). *Economic Dynamics: Theory and Computation*. Cambridge: MIT Press.
- [46] Stokey, N. L. and R. E. Lucas Jr. with E. Prescott, (1989). *Recursive Methods in Economic Dynamics*. Cambridge, MA: Harvard University Press.
- [47] Tsyrennikov (2013). *Fiscal Policy, Sovereign Debt and Default with Model Misspecification*, Cornell University manuscript.

- [48] Villemot, S., (2012). Accelerating the resolution of sovereign debt models using an endogenous grid method. Dynare working paper 17, <http://www.dynare.org/wp>.

Appendices

In Appendix A, we provide a description of ECM-VF and ECM-DVF for the one-country model. In Appendix B, we describe how to implement these methods for the multicountry model. In Appendix C, we present additional accuracy checks for the multicountry model. Finally, in Appendix D, we show numerical methods used to solve a default risk model.

Appendix A: ECM for the one-agent model

We first describe the ECM-VF method that solves for value function in one-agent model (1)–(3).

ECM-VF

Initialization.

- i). Choose an approximating function $V(\cdot; a) \approx V$.
 - ii). Choose integration nodes, ε_j , and weights, ω_j , $j = 1, \dots, J$.
 - iii). Construct a grid $\{k_m, z_m\}_{m=1, \dots, M}$.
 - iv). Make an initial guess on $a^{(1)}$.
-

Iterative cycle. At iteration i , given $a^{(i)}$, perform the following steps.

Step 1. For $m = 1, \dots, M$, compute:

- i). $c_m = \left[\frac{V_k(k_m, z_m; a^{(i)})}{1 - \delta + z_m \alpha k_m^{\alpha-1}} \right]^{-1/\gamma}$.
- ii). $k'_m = (1 - \delta) k_m + z_m k_m^\alpha - c_m$.
- iii). $v_m = \frac{c_m^{1-\gamma} - 1}{1-\gamma} + \beta \sum_{j=1}^J \omega_j V(k'_m, z_m^\rho \exp(\varepsilon_j); a^{(i)})$.

Step 2. Computation of a that fits the values v_m on the grid.

Run a regression to find $\hat{a} = \arg \min_a \sum_{m=1}^M \|v_m - V(k_m, z_m; a)\|$.

Step 3. Convergence check and fixed-point iteration.

Stop if $\frac{1}{\xi M} \sum_{m=1}^M \left| \frac{(k'_m)^{(i+1)} - (k'_m)^{(i)}}{(k'_m)^{(i)}} \right| < 10^{-10}$, where $\xi = 0.1$ is a damping parameter.

Otherwise, use damping to compute $a^{(i+1)} = (1 - \xi) a^{(i)} + \xi \hat{a}$ and go to Step 1.

We next describe ECM-DVF that solves for the derivatives of value function in the one-country model (the steps that are identical to those in ECM-VF are omitted).

ECM-DVF

Initialization.

i). Choose an approximating function $V_k(\cdot; a) \approx V_k$.

...

Iterative cycle. At iteration i , given $a^{(i)}$, perform the following steps.

Step 1. For $m = 1, \dots, M$, compute

...

iii). $d_m = \beta [1 - \delta + \alpha z k_m^{\alpha-1}] \sum_{j=1}^J \omega_j V_k(k'_m, z_m^\rho \exp(\varepsilon_j); a^{(i)})$.

Step 2. Computation of a that fits the values d_m on the grid.

Run a regression to find $\hat{a} = \arg \min_a \sum_{m=1}^M \|d_m - V_k(k_m, z_m; a)\|$.

...

5.1 Appendix B: ECM for multicountry model

We now describe the ECM-VF method that solves for value function in multicountry model (27)–(29).

ECM-VF

Initialization.

- i). Choose approximating functions $K^h(\cdot; a^h) \approx K^h$, $h = 1, \dots, N$ and $V(\cdot; \varpi) \approx V$.
 - ii). Choose integration nodes, $\varepsilon_j = (\varepsilon_j^1, \dots, \varepsilon_j^N)$, and weights, ω_j , $j = 1, \dots, J$.
 - iii). Fix the simulations length T and the initial condition $(\mathbf{k}_0, \mathbf{z}_0)$.
 - iv). Draw and fix a sequence of productivity levels $\{\mathbf{z}_t\}_{t=1, \dots, T}$ using (29).
 - v). Construct integration nodes, $\mathbf{z}_{t+1, j} = (z_{t+1, j}^1, \dots, z_{t+1, j}^N)$ with $z_{t+1, j}^h = (z_t^h)^\rho \exp(\varepsilon_j^h)$.
 - vi). Make an initial guess on $(a^1)^{(1)}, \dots, (a^N)^{(1)}$.
-

Iterative cycle. At iteration i , given $(a^1)^{(i)}, \dots, (a^N)^{(i)}$, perform the following steps.

Step 1. For $t = 1, \dots, T$,

- i). Use $k_{t+1}^h = \widehat{K}^h(\mathbf{k}_t, \mathbf{z}_t; (a^h)^{(i)})$, $h = 1, \dots, N$, to recursively calculate $\{\mathbf{k}_{t+1}\}_{t=0, \dots, T}$.
- ii). Compute $\{\mathbf{c}_t, \ell_t\}_{t=0, \dots, T}$ satisfying (28), (31) and (32) given $\{\mathbf{k}_t, \mathbf{z}_t, \mathbf{k}_{t+1}\}_{t=0, \dots, T}$.
- iii). Find $\widehat{\varpi}$ satisfying $V(\mathbf{k}_t, \mathbf{z}_t; \widehat{\varpi}) = \sum_{h=1}^N \tau^h u^h(c_t^h, \ell_t^h) + \beta \sum_{j=1}^J V(\mathbf{k}_{t+1}, \mathbf{z}_{t+1, j}; \widehat{\varpi})$.
- iv). Use $V(\cdot; \widehat{\varpi})$ to find $V_{k^h}(\mathbf{k}_t, \mathbf{z}_t; \widehat{\varpi})$ and to infer $V_{k^h}(\mathbf{k}_{t+1}, \mathbf{z}_{t+1, j}; \widehat{\varpi})$ for $j = 1, \dots, J$.
- v). Compute $\widehat{k}_{t+1}^h \equiv \sum_{j=1}^J \beta \frac{V_{k^h}(\mathbf{k}_{t+1}, \mathbf{z}_{t+1, j}; \widehat{\varpi})}{V_{k^h}(\mathbf{k}_t, \mathbf{z}_t; \widehat{\varpi})} \frac{[\pi_t^h + z_t^h f_k^h(k_t^h, \ell_t^h)]}{o_t^h} k_{t+1}^h$, $h = 1, \dots, N$.

Step 2. Computation of a^h that fits the values \widehat{k}_{t+1}^h on the grid.

Run regressions to find $\widehat{a}^h \equiv \arg \min_{a^h} \sum_{t=1}^T \left\| \widehat{k}_{t+1}^h - K^h(\mathbf{k}_t, \mathbf{z}_t; a^h) \right\|$.

Step 3. Convergence check and fixed-point iteration..

Stop if $\frac{1}{TN\xi} \sum_{t=1}^T \sum_{h=1}^N \left| \frac{(\widehat{k}_{t+1}^h)^{(i+1)} - (\widehat{k}_{t+1}^h)^{(i)}}{(\widehat{k}_{t+1}^h)^{(i)}} \right| < 10^{-7}$, where $\xi = 0.05$ is damping parameter.

Otherwise, use damping to compute $(a^h)^{(i+1)} = (1 - \xi)(a^h)^{(i)} + \xi \widehat{a}^h$ and go to Step 1.

We next describe ECM-DVF that solves for the derivatives of value function in the multicountry model (the steps that are identical to those in ECM-VF are omitted).

ECM-DVF

Initialization.

i). Choose approximating functions $K^h(\cdot; a^h) \approx K^h$ and $V_{k^h}(\cdot; \varpi^h) \approx V_{k^h}$, $h = 1, \dots, N$.

...

Iterative cycle. At iteration i , given $(a^1)^{(i)}, \dots, (a^N)^{(i)}$, perform the following steps.

Step 1. For $t = 1, \dots, T$,

...

iii). Find $d_t^h \equiv u_c^h(c_t^h, \ell_t^h) [\pi_t^h + z_t^h f_k^h(k_t^h, \ell_t^h)]$ and
find $\widehat{\varpi}^h \equiv \arg \min_{\varpi^h} \|d_t^h - V_{k^h}(\mathbf{k}_t, \mathbf{z}_t; \varpi^h)\|$, $h = 1, \dots, N$;

iv). Use $V_{k^h}(\cdot; \widehat{\varpi}^h)$ to find $V_{k^h}(\mathbf{k}_t, \mathbf{z}_t; \widehat{\varpi}^h)$ and
to infer $V_{k^h}(\mathbf{k}_{t+1}, \mathbf{z}_{t+1,j}; \widehat{\varpi}^h)$ for $j = 1, \dots, J$;

...

5.2 Appendix C: Additional accuracy tests for multicountry model

Below, we report the residuals in the model's equations on spheres of different radii for ECM-VF and ECM-DVF methods in the context of multicountry model (27)–(29); we follow Juillard and Villemot (2011) in the construction of these accuracy measures.

Table 4. Accuracy and speed of ECM-DVF in the multicountry model.^a

Number of Countries	Polynomial Degree	CPU	$r = 0.01$		$r = 0.1$		$r = 0.3$		Stochastic Simulation	
			L_1	L_∞	L_1	L_∞	L_1	L_∞	L_1	L_∞
$N = 2$	2 nd	29	-4.95	-3.66	-4.12	-2.69	-3.62	-2.13	-3.97	-2.51
	3 rd	34	-5.09	-3.71	-4.14	-2.72	-3.65	-2.18	-4.01	-2.51
$N = 4$	2 nd	155	-4.90	-3.66	-3.96	-2.70	-3.48	-2.13	-3.86	-2.48
	3 rd	1402	-4.92	-3.68	-3.99	-2.74	-3.50	-2.20	-3.90	-2.50
$N = 6$	2 nd	629	-4.86	-3.64	-3.91	-2.68	-3.43	-2.08	-3.84	-2.47
	3 rd	21809	-4.88	-3.66	-3.95	-2.69	-3.47	-2.16	-3.88	-2.51
$N = 8$	2 nd	2888	-4.84	-3.62	-3.88	-2.65	-3.40	-2.04	-3.83	-2.48
	3 rd	89872	-4.92	-3.68	-3.94	-2.66	-3.41	-1.94	-3.90	-2.48

^a L_1 and L_∞ are, respectively, the average and maximum of absolute value of residuals across optimality conditions and test points (in log10 units). CPU is the time necessary for computing a solution (in seconds). For the columns " $r = 0.01$ ", " $r = 0.1$ " and " $r = 0.3$ ", the set of test points is a draw of 1000 points located on a sphere with radii 0.01, 0.1 and 0.3, respectively. For the column "Stochastic Simulation", the set of test point is a stochastic simulation of 10.000 observations.

^b In a policy function of 1 country.

5.3 Appendix D: ECM for default risk model

We show the ECM-VF method that solves for value function in default risk model (42)-(43).

ECM-VF

Initialization.

- i). Choose an approximating function $V(\cdot; a) \approx V$.
 - ii). Choose integration nodes, ε_j , and weights, ω_j , $j = 1, \dots, J$.
 - iii). Construct a grid $\{b_m, y_m\}_{m=1, \dots, M}$ covering the area $y > \bar{y}(b)$.
 - iv). Compute $q(b', y)$ using (41).
 - v). Define $\mathcal{L}(b) \equiv q(b)b$ and precompute its inverse \mathcal{L}^{-1} .
 - vi). Make an initial guess on $a^{(1)}$.
-

Iterative cycle. At iteration i , given $a^{(i)}$, perform the following steps.

Step 1. For $m = 1, \dots, M$, compute:

- i). $c_m = V_b(b_m, z_m; a^{(i)})^{-1/\gamma}$.
- ii). $b'_m = \mathcal{L}^{-1}(b_m + y_m - c_m)$.
- iii). $v_m = \frac{c_m^{1-\gamma} - 1}{1-\gamma} + \beta \sum_{j=1}^J \omega_j \max \{ V(b'_m, y_m^\rho \exp(\varepsilon_j); a^{(i)}) V(b'_m, \bar{y}(b_m); a^{(i)}) \}$

Step 2. Computation of a that fits the values v_m on the grid.

Run a regression to find $\hat{a} = \arg \min_a \sum_{m=1}^M \|v_m - V(k_m, z_m; a)\|$.

Step 3. Convergence check and fixed-point iteration.

Stop if $\max |v_m^{(i+1)} - v_m^{(i)}| < 10^{-4}$, where $\xi = 0.1$ is a damping parameter.

Otherwise, use damping to compute $a^{(i+1)} = (1 - \xi) a^{(i)} + \xi \hat{a}$ and go to Step 1.

We next describe ECM-DVF that solves for the derivatives of value function in default risk model (42)-(43) (the steps that are identical to those in ECM-VF are omitted).

ECM-DVF

Initialization.

i). Choose an approximating function $V_k(\cdot; a) \approx V_k$.

...

Iterative cycle. At iteration i , given $a^{(i)}$, perform the following steps.

Step 1. For $m = 1, \dots, M$, compute

...

iii).
$$d_m = \frac{\beta \sum_{j=1}^J \omega_j V_k(b'_m, y'_m \exp(\varepsilon_j); a^{(i)}) \mathbf{1}(y'_m \exp(\varepsilon_j) > \bar{y}(b'_m))}{q_b(b'_m, y_m) b'_m + q(b'_m, y_m)}.$$

...

Step 2. Computation of a that fits the values d_m on the grid.

Run a regression to find $\hat{a} = \arg \min_a \sum_{m=1}^M \|d_m - V_k(k_m, z_m; a)\|$.

...

We next describe conventional VFI that solves for value function in default risk model (42)-(43) (the steps that are identical to those in ECM-VF are omitted).

VFI

...

Step 1. For $m = 1, \dots, M$, use a numerical solver to find

i).
$$\max_{b'_m} \left\{ \frac{c_m^{1-\gamma} - 1}{1-\gamma} + \beta \sum_{j=1}^J \omega_j \max [V(b'_m, y'_m \exp(\varepsilon_j); a^{(i)}), V(b'_m, \bar{y}(b_m); a^{(i)})] \right\},$$

where $c_m = b_m + y_m - b'_m q(b'_m, y_m)$.

...
