Envelope Condition Method with an Application to Default Risk Models

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

May 1, 2015

Abstract

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 of iteration on Bellman equation, ECM constructs policy functions using envelope conditions which are simpler to analyze numerically than first-order conditions. Second, to increase the accuracy of solutions, ECM solves for derivatives of value function jointly with 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 finally use ECM to solve a challenging default risk model with a kink in value and policy functions.

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 and two very competenet anonymous referees for many thoughtful comments and suggestions. Also, we benefited from comments of Grace Gu, Kenneth L. Judd, Seung Lee, seminar participants at Stanford University, Boston University, Santa Clara University and University of California at Santa Cruz as well as from participants of the 2014 meeting of Society of Computational Economics. Errors are ours. Lilia Maliar and Serguei Maliar acknowledge support from the Hoover Institution and Department of Economics at Stanford University, University of Alicante, Santa Clara University and MECD 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 of iteration on Bellman equation, ECM constructs policy functions using envelope conditions which are simpler to analyze numerically than first-order conditions. Second, to increase the accuracy of solutions, ECM solves for derivatives of value function jointly with 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 finally use ECM to solve a challenging default risk model with a kink in value and policy functions.

We present ECM in the context of three applications: a one-agent growth model, a multi-country model of international trade and a default risk model. In our first application, we consider a stylized optimal growth model with inelastic labor supply. To solve such a model, conventional VFI constructs policy function by finding a maximum of the right side of the Bellman equation. This is done either directly (by using numerical maximization) or via first-order condition (by using a numerical solver). In contrast, the ECM methods construct policy functions by finding a solution to the envelope condition. Because such a solution can be derived in a closed form, ECM requires only direct calculations and avoids the need of either numerical optimization or numerical solvers when iterating on the Bellman equation.

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 value function. This is because solving accurately for value function does not necessarily leads to sufficiently accurate approximations of its derivatives. For example, if 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 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.

We then investigate the convergence properties of the constructed class of ECM methods in the context of the studied optimal growth model. We establish that ECM has the same fixed point solution as the regular Bellman operator with some

additional technical restriction. However, the ECM operator does not possess the property of contraction mapping like the regular Bellman operator. In this respect, the ECM class of methods is similar to the Euler equation class of methods for which global convergence results are generally infeasible. However, the fact that the convergence theorems cannot be established for some numerical method does not mean that the method is not useful, in particular, Euler equation methods are useful in a variety of context. In our numerical experiments, the ECM method has good convergence properties and produces accurate solutions in a wide range of the model's parameters.

In our second application, we construct a version of ECM that is suitable for 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 is 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.

Finally, our third application is a default risk model of Arellano (2008). 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. Nonetheless, at the optimal debt level, the decision functions are continuously differentiable and satisfy FOCs; see Clausen and Strub (2013) for a version of the envelope theorem that applies to models with default risk and a survey of envelope theorems in the literature. We show that the ECM methods are fast in computing this model. Relative to the expensive VFI method, ECM speeds up the computation time by more than 50 times. However, the convergence is more diffi-

[1] Maliar and Maliar (2014) for a survey of these and other numerical techniques that are tractable in problems with a large number of state variables.

[2] The 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).

cult to attain in this model. Numerical errors in approximating value function along iteration may lead to nonmonotone policy functions and result in non-convergence. Damping and shape preserving restrictions on value function can help to deal with this problem.

While our analysis is limited to the benchmark default risk model, 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), Hopenhayn and Werning (2008), Bianchi, Hatchondo and Martinez (2014), Maliar, Maliar and Pérez-Sebastián (2008), Chatterjee and Eyigundor (2011), Arellano, Bai, and Kehoe (2013) and Tsyrennikov (2013); Aguiar, Amador, Farhi and Gopinath (2015); 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.

We next discuss the relation of the ECM method to other numerical methods in the literature. Dynamic programming methods are introduced in Bellman (1957) and Howard (1960) in the context of stationary, infinite-horizon Markovian problems. 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)), policy iteration (e.g., Santos and Rust (2004)), nonexpensive approximations (Stachurski (2008)), approximate DP methods (e.g., Powell (2011)), polyhedral approximations (e.g., Fukushima and Waki (2011)), random contractions (Pal and Stachurski (2013)); also see Rust (2008), Judd (1998), Santos (1999), Stachurski (2009) for literature reviews. From one side, many methods, which are accurate and reliable in problems with low dimensionality, are intractable 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). In contrast to such methods, ECM relies on accurate deterministic integration methods and is both tractable and accurate in problems with high dimensionality.

Endogenous grid method (EGM) of Carroll (2005) reduces the cost of conven-

tional 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 whether an inequality constraint binds or whether 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 contrasts, 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 (2011, 2012), Fernández-Villaverde, Gordon, Guerrón-Quintana, and Rubio-Ramírez (2012), Aruoba and Schorfheide (2013), López-Salido (2014), Maliar and Maliar (2015), etc.; and large-scale OLG models with up to 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 accurate polynomial approximations up to third degree, while the Euler equation methods participating in the comparison analysis of Kollmann, Maliar, Malin and Pichler (2011) are limited to less accurate 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 begin by illustrating 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 value function, V, that solves the Bellman equation,

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

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

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

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 strictly concave. The primes on variables denote next-period values, and $E[V(k',z')]$ is an expectation conditional on state (k,z).

2.2 First order condition (FOC) versus envelope condition (EC)

Under our assumptions, a solution to Bellman equation (1)–(3) exists, is interior and is unique; and optimal value function V is differentiable, strictly increasing and strictly concave; see Stokey and Lucas with Prescott (1989), Santos (1999) and Stachurski (2010) for a discussion. We compute the derivative of V in the left side of (1) to obtain

$$V_1(k,z) = u'(C(k,z)) C_1(k,z) + \beta E[V_1(K(k,z),z')] K_1(k,z), \tag{4}$$

where $c = C(k,z)$ and $k' = K(k,z)$ are the optimal policy functions. (Here and further in the paper, $F_i(...,x_i,...)$ denotes a first-order partial derivative of function $F(...,x_i,...)$ with respect to ith variable x_i).

According to (2), we have $C_1(k,z) = 1 - \delta + zf(k) - K_1(k,z)$. Substituting the latter result into (4) yields

$$\underbrace{V_1(k,z) - u'(C(k,z))[1 - \delta + zf'(k)]}_{\text{EC}} = \underbrace{\{\beta E[V_1(k',z')] - u'(C(k,z))\}}_{\text{FOC}} K_1(k,z). \tag{5}$$

The left side of (5) is the envelope condition (EC) and the right side corresponds to first order condition (FOC) of the maximization problem (1)–(3).

Conventional value function iteration (VFI) and policy iteration (PI) constructs policy functions such as consumption function by setting the FOC equal to zero (we denote such a function by $C_{FOC}(k,z)$):

$$u'\left(C_{FOC}(k,z)\right) = \beta E\left[V_1(k',z')\right]. \tag{6}$$

In contrast, the class of methods advocated in the present paper solves for consumption function by setting the EC equal to zero (we denote such a function by $C_{EC}(k,z)$):

$$V_1(k,z) = u'\left(C_{EC}(k,z)\right)\left[1-\delta+zf'(k)\right]. \tag{7}$$

We refer to this new class of methods as *envelope condition methods* (ECM), following Maliar and Maliar (2013).

In the true solution, both ways of constructing the consumption function, defined by (6) and (7), must lead to the same consumption function since the solution must satisfy both the FOC and EC, i.e., $C_{FOC} = C_{EC}$. However, operationally, the construction of C_{EC} from (7) is simple, (namely, C_{EC} can be derived in a closed form), whereas the construction of C_{FOC} from (6) is more complicated (it generally involves using a numerical solver). Replacing FOC with EC leads to a dramatic reduction in cost of value iteration, as the policy functions must be constructed a large number of times along iteration; see Maliar and Maliar (2013) for numerical examples. However, it turned out that the new way of constructing policy functions affects also the convergence properties of VFI and PI methods. In the remainder of the paper, we formulate several different variants of ECM methods, explore their convergence properties, discuss their relation to the literature and illustrate their applications with examples.

2.3 Value function iteration

We now describe a variant of ECM that finds a solution to Bellman equation by iterating on value function and we compare it to two related methods in the literature, conventional value function iteration and endogenous grid method of Carroll (2005).

2.3.1 ECM-VF

ECM-VF finds consumption from envelope condition (7). It constructs value function V satisfying (1)–(3) by the following iteration procedure:

Algorithm 1. ECM-VF

Given V, for each point (k, z), define the following recursion:
i). Find $c = u'^{-1}\left[\frac{V_1(k,z)}{1-\delta+zf'(k)}\right]$.
ii). Find $k' = (1-\delta)k + zf(k) - c$.
iii). Find $\widehat{V}(k,z) = u(c) + \beta E[V(k', z')]$.
Iterate on i)-iii) until convergence $\widehat{V} = V$.

The formulas in i) and ii) are envelope condition (7) and budget constraint (2), respectively, and the formula in iii) is Bellman equation (1), evaluated under optimal policy functions (which eliminates the maximization sign). We observe that under ECM-VF, neither numerical maximization nor numerical solver is necessary for iteration on the Bellman equation but just direct calculations.

2.3.2 Conventional VFI and EGM

Two related methods in the literature are conventional VFI and endogenous grid method (EGM) of Carroll (2005). Both methods perform time iteration on FOC (6), namely, they guess value function at $t+1$ and use the Bellman equation to compute value function at t. FOC (6), combined with budget constraint (2), becomes

$$u'(c) = \beta E[V_1((1-\delta)k + zf(k) - c, z')]. \tag{8}$$

Conventional VFI finds consumption from FOC (8).

Algorithm 2. Conventional VFI

Given V, for each point (k, z), define the following recursion:
i). Solve for c satisfying $u'(c) = \beta E[V_1((1-\delta)k + zf(k) - c, z')]$.
ii). Find $k' = (1-\delta)k + zf(k) - c$.
iii). Find $\widehat{V}(k,z) = u(c) + \beta E[V(k', z')]$.
Iterate on i)-iii) until convergence $\widehat{V} = V$.

Conventional VFI is expensive because Step i) requires us to numerically find a root to (8) for each (k, z) by interpolating V_1 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 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).

Carroll (2005) proposes a way to reduce the cost of conventional VFI. The EGM method 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 $E\left[V_1\left(k', z'\right)\right]$ up-front and thus can avoid costly interpolation and approximation of expectation in a rootfinding procedure.

Algorithm 3. EGM of Carroll (2005)

Given V, for each point (k', z), define the following recursion:
i). Find $c = u'^{-1}\left\{\beta E\left[V_1\left(k', z'\right)\right]\right\}$.
ii). Solve for k satisfying $k' = (1-\delta)k + zf(k) - c$.
iii). Find $\widehat{V}(k, z) = u(c) + \beta E\left[V\left(k', z'\right)\right]$.
Iterate on i)-iii) until convergence $\widehat{V} = V$.

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 numerically on each iteration (except of the very last iteration).

As we have seen, ECM-VF avoids the rootfinding completely in the studied model (even without a change of variables). Thus, it attains the same outcome 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.

However, 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.

2.4 Policy iteration

We now describe a variant of ECM that solves Bellman equation by iterating on policy function and we compare it to conventional PI methods, e.g., Santos and Rust (2004); see Santos and Rust (2004), Rust (2008) and Stachurski (2009) for a general discussion of policy iteration methods.

2.4.1 ECM-PI

We call by ECM-PI the variant of ECM that performs PI instead of VFI.

Algorithm 4. ECM-PI

Given C, for each point (k,z), define the following recursion:
i). Find $K(k,z) = (1-\delta)k + zf(k) - C(k,z)$.
ii). Solve for V satisfying $V(k,z) = u(C(k,z)) + \beta E[V(K(k,z),z')]$.
iii). Find $\widehat{C}(k,z) = u'^{-1}\left[\frac{V_1(k,z)}{1-\delta+zf'(k)}\right]$.
Iterate on i)-iii) until convergence $\widehat{C} = C$.

That is, ECM-PI method guesses policy functions for consumption $c = C(k,z)$, finds the corresponding policy function for capital $k = K(k,z)$, computes the corresponding V by iteration on the Bellman equation until convergence holding the policy functions fixed and recompute the policy function $\widehat{C}(k,z)$, iterating until convergence. An example of ECM iterating on policy function is shown in Section 4.2.1.

2.4.2 Conventional policy iteration

Conventional policy iteration methods constructs consumption function using FOC (6) instead of envelope condition (7).

Algorithm 5. Conventional PI

Given C, for each point (k,z), define the following recursion:
i). Find $K(k,z) = (1-\delta)k + zf(k) - C(k,z)$.
ii). Solve for V satisfying $V(k,z) = u(C(k,z)) + \beta E[V(K(k,z),z')]$.
iii). Find $\widehat{C}$ satisfying $u'\left(\widehat{C}(k,z)\right) = \beta E\left[V_1\left((1-\delta)k + zf(k) - \widehat{C}(k,z), z'\right)\right]$.
Iterate on i)-iii) until convergence $\widehat{C} = C$.

The difference between ECM-PI and conventional PI consists in step iii). in which the policy function for next iteration is constructed.

2.5 Iteration on derivatives of value function

The ECM methods described up to now solve for value function (either directly or via PI). We now describe versions of ECM that solve for derivative of value function instead of value function itself. We argue that the derivative-based version of ECM has similarity to the Euler-equation class of methods.

2.5.1 ECM-DVF

The studied ECM method suggests a useful recursion for the derivative of value function. We first construct consumption function $C(k,z)$ satisfying FOC (6) under the current value function $\beta E[V_1(k',z')] = u'(C(k,z))$, and we then use (4) to obtain the derivative of value function for next iteration:

$$V_1(k,z) = \beta[1-\delta+zf'(k)]E[V_1(k',z')]. \tag{9}$$

This leads to a solution method that we call ECM-DVF.

Algorithm 6. ECM-DVF

Given V_1 for each point (k,z), define the following recursion:
i). Find $c = u'^{-1}\left[\frac{V_1(k,z)}{1-\delta+zf'(k)}\right]$.
ii). Find $k' = (1-\delta)k + zf(k) - c$.
iii). Find $\widehat{V}_1(k,z) = \beta[1-\delta+zf'(k)]E[V_1(k',z')]$.
Iterate on i)-iii) until convergence $\widehat{V}_1 = V_1$.
Given the converged policy functions, find V satisfying $V(k,z) = u(c) + \beta E[V(k',z')]$.

The difference of ECM-DVF from the previously studied ECM-VF consists in that we iterate on V_1 without computing V on each iteration. We only compute V at the very end, when both V_1 and the optimal policy functions are constructed. Again, neither numerical maximization nor a numerical solver is necessary under ECM-DVF but only direct calculations.

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 value function. This is because solving accurately for value function does not lead to sufficiently accurate approximations of its derivatives. For example, assume that 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 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.

Maliar and Maliar (2013) also show how to construct a version of EGM-DVF of Carroll (2005) that iterates on the derivatives of value function instead of the value function itself in a way that is parallel to ECM-DVF. This paper finds that the performance of ECM-DVF and EGM-DVF is very similar in terms of their accuracy and speed in the context of a model with elastic labor supply. Finally, it is straightforward to formulate the variants of ECM-DVF and ECM-DVF that perform PI instead of VFI but we omit this extension to save on space.

2.5.2 Euler equation methods

ECM-DVF has similarity to Euler equation methods; see Judd (1998), Santos (1999) for a general discussion of such methods. The standard Euler equation follows from optimality conditions (6), (7): we update (7) to obtain $V_1(k', z')$ and we substitute the result into (6) to eliminate the unknown derivative of the value function,

$$u'(c) = \beta E\left[u'(c')\left(1-\delta+z'f'(k')\right)\right]. \tag{10}$$

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). Below, we provide an example of Euler equation method (other recursions for such methods are possible).

Algorithm 7. Euler equation algorithms

Given $C(k,z)$, for each point (k,z), define the following recursion:
i). Find $K(k,z) = (1-\delta)k + zf(k) - C(k,z)$.
ii). Find $c' = C(K(k,z), z')$.
iii). Find $\widehat{C}(k,z) = u'^{-1}\{\beta E[u'(c')(1-\delta+z'f'(k'))]\}$.
Iterate on i)-iii) until convergence $\widehat{C} = C$.

Similarly to ECM, Euler equation methods do not solve for value function but only for decision (policy) functions. One possible decision functions is a derivative of value function. Thus, the ECM recursion (9) can be also viewed as an Euler equation written in terms of the derivative of value function.

2.6 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), Santos (1999, 2000), Stachurski (2009) and Krueger (2012) for a discussion of these assumptions.

2.6.1 ECM-VF

We study the convergence of the ECM method iterating on value function, described in Section 2.2.1; see "Algorithm 1. ECM-VF". Our analysis relies on the comparison of ECM to the conventional VFI iteration; see "Algorithm 2. FOC-VFI". We define the regular Bellman operator T for the model (1)–(3) as

$$TV(k,z) \equiv u(C_{FOC}(k,z)) + \beta E[V(zf(k) - C_{FOC}(k,z), z')], \tag{11}$$

$$C_{FOC}(k,z) : u'(c) = \beta E[V_1(zf(k) - c, z')], \tag{12}$$

where $C_{FOC}(k,z)$ is the consumption function defined implicitly by FOC (6). (To simplify the notation, we assume $\delta = 1$ in this section).

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

$$QV(k,z) \equiv u(C_{EC}(k,z)) + \beta E[V(zf(k) - C_{EC}(k,z), z')], \tag{13}$$

$$C_{EC}(k,z) : V_1(k,z) = u'(c) zf'(k), \tag{14}$$

where $C_{EC}(k,z)$ is the consumption function defined implicitly by envelope condition (7).

Does ECM-VF have the same fixed point as conventional VFI? The operators T and Q are different but any fixed point of T is also a fixed point of Q and vice versa (with an additional technical restriction). Specifically, we have the following result.

Proposition 1. Let V^* be a fixed point of Q such that $K_1 \neq 0$. Than, $V^* = TV^*$ iff $V^* = QV^*$.

Proof. The proof follows by formula (5) that shows that imposing FOC enforces the envelope condition and vice versa.

i) To see that $V^* = TV^*$ implies $V^* = QV^*$, note that if V^* is a fixed point of T, than assuming interiority, C_{FOC} must satisfy FOC (12). Hence, according to (5), C_{FOC} must also satisfy envelope condition (14). Hence, $C_{FOC} = C_{EC}$ and hence, $V^* = QV^*$.

ii) In the opposite direction, note that if V^* is a fixed point of Q, then C_{EC} satisfies envelope condition (14). According to (5), C_{EC} must either satisfy FOC (12) or it must be the case that $K_1 = 0$. Since the latter possibility is ruled out by assumption, we conclude that $C_{FOC} = C_{EC}$. Hence, $V^* = QV^*$ implies $V^* = TV^*$. ‖

The restriction $K_1 \neq 0$ is important and allows us to rule out degenerate functions that satisfy envelope condition but not FOC and hence, that are not solutions to the Bellman equation. We illustrate this point by way of example.

Example 1. Assume $u(0) = f(0) = 0$ and consider $V(k,z) = u(zf(k))$. According to (14), we have

$$C_{EC}(k,z) = u'^{-1}\left[\frac{V_1(k,z)}{zf'(k)}\right] = u'^{-1}\left[\frac{u'(zf(k))\,zf'(k)}{zf'(k)}\right] = zf(k).$$

Inserting this result in the right side hand side of (13), we have

$$\begin{aligned} QV(k,z) = u(C_{EC}(k,z)) + \beta E\left[V(zf(k) - C_{EC}(k,z), z')\right] \\ u(zf(k)) + \beta E\left[V(zf(k) - zf(k), z')\right] \\ = u(zf(k)) + \beta E\left[V(0,z')\right] = u(zf(k)) \end{aligned}$$

Hence, $V(k,z) = u(zf(k))$ is a fixed point of Q which is not fixed point of T. This is a degenerate fixed point that has capital function $K = 0$ that does not satisfy the restriction $K_1 = 0$. ‖

Does ECM-VF have the contraction mapping property as conventional VFI? Our goal is to analyze the convergence properties of ECM-VF. In order to do this, it is useful to recall the convergence results for the regular Bellman operator T. It is well-known that the Bellman operator is a contraction mapping and thus, it guarantees a convergence to a fixed point V^* starting from an arbitrary initial guess V, in the space of continuous bounded functions, i.e., $T^nV \to V^*$ as $n \to \infty$. The proof of this fact is as follows:

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

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

Using the property of a maximum operator, we obtain

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

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? In other words, does Q guarantee a convergence to a fixed point $Q^n V \to V^*$ as $n \to \infty$ in the space of continuous bounded functions? Example 1 suggests a negative answer to this question. Indeed, if we use $V(k,z) = u(zf(k))$ for initializing ECM-VF (such a guess is frequently used for initializing conventional VFI in applications), ECM-VF will get stuck in a wrong fixed point with $K = 0$. Clearly, ECM-VF does not guarantee a convergence from any initial guess.

To further explore the convergence properties of ECM-VF, let us repeat the same steps for this method as we did for the regular Bellman operator. We have

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

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

$$\begin{aligned}c_V(k,z) &: V_1(k,z) = u'(c)\, zf'(k), \\ c_W(k,z) &: W_1(k,z) = u'(c)\, zf'(k).\end{aligned} \tag{17}$$

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

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

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.

To investigate the convergence properties of the term $\|u(c_V) - u(c_W)\|$, we study the recursion that ECM implies for the derivative V_1. On iteration n, we first construct consumption function $C^n(k,z)$ satisfying envelope condition (7) under the current value function $V_1^n(k,z) = u'(C^n(k,z))[1 - \delta + zf'(k)]$, and we then use (5) to obtain following recursion for the derivative of value function for iteration $n+1$:

$$V_1^{n+1}(k,z) = V_1^n(k,z) + \left\{ \beta E[V_1^n(k',z')] - \frac{V_1^n(k,z)}{1 - \delta + zf'(k)} \right\} K_1^n(k,z). \tag{19}$$

With the result (19), we obtain the following equation that describes the evolution of $C^n(k,z)$ along iteration for the ECM-VF method:

$$\begin{aligned} u'\left(C^{n+1}(k,z)\right) = {} & u'(C^n(k,z)) \\ & + \{\beta E[u'(C^n(k',z'))z'f'(k')] - u'(C^n(k,z))\} \frac{K_1^n(k,z)}{zf'(k)}. \end{aligned} \tag{20}$$

Effectively, the recursion (20) attempts to solve an Euler equation

$$u'(C^n(k,z)) = \beta E[u'(C^n(k',z'))z'f'(k')] \tag{21}$$

using fixed point iteration. However, the convergence of fixed point iteration on Euler equation is not generally guaranteed; see Maliar and Maliar (2014) for a discussion. The absence of strong convergence results is an important shortcoming of the class of Euler equation methods relative to the class of conventional value-iterative approaches.

Why do the Bellman and ECM operators have different convergence properties? Note that a solution to EC (7) does not maximize the right side of the Bellman equation for any V that occurs in iteration (it only does for a limiting fixed-point solution V^*). Similarly, the regular Bellman operator, enforces FOC (6) in each iteration but does not enforces envelope condition (7); it only enforces such a condition in the limiting fixed point V^*. Since ECM does not produce a maximum of the Bellman equation in each iteration, it is not possible to cancel out $\|u(c_V) - u(c_W)\|$, as is possible to do under the regular Bellman operator. This fact prevents us from having the property of contraction mapping.

2.6.2 ECM-PI

Global convergence for policy iteration is established for a space of concave value functions in Santos and Rust (2004). Thus, the convergence of policy iteration methods requires more stringent conditions than the convergence of the regular Bellman operator that is established for a more general class of continuous bounded functions.

Unfortunately, the restriction of concavity of value function is still not sufficient to insure the convergence of ECM-PI. This is because ECM-PI is subject to the same shortcomings as ECM-VF. Namely, the conventional PI method computes policy functions to exactly satisfy FOC (21) exactly on each iteration while the ECM-PI method again applies fixed-point iteration (20) to solve FOC only in the limit. This fact can be seen by comparing Algorithm 4 and 5 which have identical steps 1 and 2 but differ in step 3.

2.6.3 ECM-DVF

We next turn to ECM-DVF methods described in Section 2.2.2, specifically, we consider the recursion ECM-DVF (9) that is used in Algorithm 7. We introduce an operator D that corresponds to the ECM-DVF recursion as follows:

$$DV_1(k,z) \equiv zf'(k)\beta E\left[V_1\left(zf(k)-c_V(k,z),z'\right)\right] \tag{22}$$

$$c_{EC}(k,z) : V_1(k,z) = u'(c)zf'(k). \tag{23}$$

We investigate the convergence properties of the ECM-DVF operator as we did for ECM-VF. Let V_1 and W_1 be two bounded continuous functions. Then, we have

$$\begin{aligned}&|DV_1(k,\theta)-DW_1(k,\theta)| = \\ &\qquad zf'(k)\beta\left|EV_1\left[\left(zf(k)-c_V(k,z),z'\right)\right]-EW_1\left[\left(zf(k)-c_W(k,z),z'\right)\right]\right|\end{aligned}$$

By taking a supremum, we obtain

$$\|DV_1-DW_1\| \leq zf'(k)\beta\|V_1-W_1\|. \tag{24}$$

To have a contraction mapping, we need the term $zf'(k)\beta$ in (24) 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 contraction is to show that it satisfies Blackwell's (1965) sufficiency conditions; see Santos (1999) and Stachurski (2009) for a discussion of these conditions. It is easy to check that the operator

ECM-DVF possesses the property of monotonicity but not discounting which agrees with the result (24). (Curiously, for the previously considered operator ECM-VF (13), (14), we find exactly the opposite, namely, it possesses the property of discounting but may fail to satisfy monotonicity). Hence, our theoretical analysis does not provide a basis to affirm that ECM-DVF (22) is a contraction mapping.

2.6.4 Discussion

Conventional VFI and PI are classified as DP methods. 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 with Prescott (1989), Santos (1999) and Stachurski (2009) for reviews of formal results for such methods.

In turn, the ECM method solves for value and/or policy functions jointly and effectively includes iteration on Euler equation. For Euler equation class of 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 (2014).

The ECM class of methods is not related to a specific iterative procedure for finding a fixed point and is compatible with all the procedures discussed above. In our numerical experiments, we use fixed point iteration with damping because it is simple, inexpensive and reliable. Instead, we could have used time iteration or Newton-style solvers. For a version of the 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.

Finally, we shall emphasize that the absence of convergence results for Euler equation methods does not mean that such methods diverge and are not useful. Euler equation methods successfully converge in many applications under an appropriate implementation although we cannot prove it analytically. Moreover, even if iteration on Euler equation becomes explosive, it can often be stabilized by damping; see Maliar, Maliar and Judd (2011) for a graphical illustration. The same is true for the ECM class of methods advocated in the present paper.

2.7 Numerical analysis

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

2.7.1 Computational choices

We parameterize the model (1)–(3) using the Constant Relative Risk Aversion (CRRA) utility function, $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$, and the Cobb-Douglas production function, $f(k) = k^{\alpha}$, and we calibrate the parameters to the standard values: $\alpha = 1/3$, $\beta = 0.99$, $\delta = 0.025$, $\rho = 0.95$ and $\sigma = 0.01$. We consider two values of risk-aversion coefficient $\gamma = 1/3$ and $\gamma = 3$. 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.

2.7.2 Results

In Table 1, we show the results for the ECM-VF As a measure of accuracy, we report the average and maximum absolute unit-free residuals in Euler equation (10). Furthermore, in Table 2, we report the results for the ECM-DVF under the same parameterizations. The main finding is that both ECM-VF and ECM-DVF deliver high accuracy levels. The accuracy increases with a degree of approximating polynomial.

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

	$\gamma = 0.3$					$\gamma = 3$				
Polynomial	Euler equation		Value function		CPU	Euler equation		Value function		CPU
degree	L_1	L_∞	L_1	L_∞		L_1	L_∞	L_1	L_∞	
1st	-	-	-	-	-	-	-	-	-	-
2nd	-3.20	-2.67	-4.79	-4.40	0.94	-3.83	-3.50	-2.61	-2.60	0.32
3rd	-4.12	-3.29	-5.97	-5.45	0.59	-5.17	-4.61	-3.35	-3.34	0.22
4th	-5.06	-4.12	-7.07	-6.35	0.42	-5.27	-5.81	-3.88	-3.87	0.15
5th	-6.04	-4.92	-8.08	-7.23	0.33	-7.51	-6.91	-4.33	-4.32	0.13

[a] Notes: L_1 and L_∞ are, repectively, 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).

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

	$\gamma = 0.3$					$\gamma = 3$				
Polynomial	Euler equation		Value function		CPU	Euler equation		Value function		CPU
degree	L_1	L_∞	L_1	L_∞		L_1	L_∞	L_1	L_∞	
1st	-2.98	-2.68	-3.77	-3.60	8.86	-3.59	-3.37	-4.53	-4.45	3.32
2nd	-3.91	-3.53	-4.91	-4.56	1.08	-5.00	-4.49	-5.65	-5.40	0.46
3rd	-4.81	-4.31	-5.98	-5.35	0.88	-5.98	-5.54	-5.86	-5.71	0.31
4th	-5.79	-5.07	-6.48	-6.13	0.69	-7.24	-6.69	-5.87	-5.72	0.28
5th	-6.63	-5.85	-6.49	-6.19	0.50	-8.44	-7.89	-5.87	-5.73	0.16

[a] Notes: L_1 and L_∞ are, repectively, 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-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_1 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. Namely, 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. ECM-DVF needed more iterations to converge because it was less numerically stable than ECM-VF, and we stabilize it using damping 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 (henceforth, JEDC project) 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 productivity level, so that there are $2N$ state variables. In addition to a potentially large number of state variables, the model features an endogenous labor-leisure choice, heterogeneity in fundamentals and adjustment costs 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 $h = 1, ..., N$ 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

$$V(\boldsymbol{k},\boldsymbol{z}) = \max_{\left\{c^h,\ell^h,\left(k^h\right)'\right\}^{h=1,...,N}} \left\{\sum_{h=1}^{N} \tau^h u^h\left(c^h,\ell^h\right) + \beta E\left[V\left(\boldsymbol{k}',\boldsymbol{z}'\right)\right]\right\} \tag{25}$$

$$\text{s.t. } \sum_{h=1}^{N} c^h = \sum_{h=1}^{N}\left[z^h f^h\left(k^h,\ell^h\right) - \frac{\phi}{2}k^h\left(\frac{\left(k^h\right)'}{k^h}-1\right)^2 + k^h - \left(k^h\right)'\right], \tag{26}$$

$$\ln\left(z^h\right)' = \rho\ln z^h + \sigma\left(\varepsilon^h\right)', \tag{27}$$

where E is the operator of conditional expectation; c^h, ℓ^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 $h \in \{1,...,N\}$, respectively; c^h, ℓ^h, $\left(k^h\right)' \geq 0$; $\beta \in [0,1)$ is the discount factor; ϕ is the adjustment-cost parameter. In the process for productivity (27), $\rho \in (-1,1)$ is the autocorrelation coefficient of the productivity level; $\sigma > 0$ determines the standard deviation of the productivity level; and $\left(\left(\varepsilon^1\right)',...,\left(\varepsilon^N\right)'\right)^\top \sim \mathcal{N}\left(0_N,\Sigma\right)$ 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, $\boldsymbol{k} \equiv \left(k^1,...,k^N\right)$ and $\boldsymbol{z} \equiv \left(z^1,...,z^N\right)$, is given, and a prime on variables means their future values.

Again, we assume that the solution to DP problem (25)–(27) is interior and that value function V is differentiable. Hence, the planner's choices satisfy the FOCs and envelope condition, given, respectively, by

$$\beta E\left[V_h\left(\boldsymbol{k}',\boldsymbol{z}'\right)\right] = \lambda\left[1+\phi\cdot\left(\frac{\left(k^h\right)'}{k^h}-1\right)\right], \tag{28}$$

$$\tau^h u_1^h\left(c^h,\ell^h\right) = \lambda, \tag{29}$$

$$u_2^h\left(c^h,\ell^h\right)\tau^h = -\lambda z^h f_1^h\left(k^h,\ell^h\right), \tag{30}$$

$$V_h(\boldsymbol{k},\boldsymbol{z}) = \lambda\left[1+z^h f_1^h\left(k^h,\ell^h\right) + \frac{\phi}{2}\left(\left(\frac{\left(k^h\right)'}{k^h}\right)^2-1\right)\right], \tag{31}$$

where λ is the Lagrange multiplier, associated with the economy's resource constraint.

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 value function given policy function than to solve for policy functions given value function.

We first eliminate λ by combining FOC (28) and envelope condition (31) ,

$$1 = \frac{\beta E\left[V_h\left(\boldsymbol{k}', \boldsymbol{z}'\right)\right]}{V_h\left(\boldsymbol{k}, \boldsymbol{z}\right)} \frac{\left[\pi^h + z^h f_1^h\left(k^h, \ell^h\right)\right]}{o^h}, \tag{32}$$

where o^h and π^h are given by

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

Condition (32) relates today's and tomorrow's derivatives of the value function. We next use (32) to parameterize capital policy functions, namely, we premultiply both sides of (32) with $\left(k^h\right)'$ to obtain

$$\left(k^h\right)' = \frac{\beta E\left[V_h\left(\boldsymbol{k}', \boldsymbol{z}'\right)\right]}{V_h\left(\boldsymbol{k}, \boldsymbol{z}\right)} \frac{\left[\pi^h + z^h f_1^h\left(k^h, \ell^h\right)\right]}{o^h} \left(k^h\right)'. \tag{33}$$

The optimal capital policy functions, $\left(k^h\right)' = K^h\left(\boldsymbol{k}, \boldsymbol{z}\right)$, $h = 1, ..., N$, must satisfy a fixed point property: if we substitute such functions in the right side of (33), we must obtain the same functions in the left side. Conditions (33) for $h = 1, ..., N$ provide us with a way to implement fixed point iteration on capital policy functions. Namely, we guess some policy functions $K^h\left(\boldsymbol{k}, \boldsymbol{z}\right)$, $h = 1, ..., N$, substitute them in the right side of (33), recompute $\left(k^h\right)'$ in the left side and iterate on these steps until convergence.

Parameterization (33) is analogous to the one used in Maliar, Maliar and Judd (2011) to reparameterize the Euler equations in model (25)–(27),

$$\left(k^h\right)' = E\left\{\frac{\beta u_1^h\left(\left(c^h\right)', \left(\ell^h\right)'\right)}{u_1^h\left(c^h, \ell^h\right)} \frac{\left[\left(\pi^h\right)' + \left(z^h\right)' f_1^h\left(\left(k^h\right)', \left(\ell^h\right)'\right)\right]}{o^h}\right\} \left(k^h\right)'. \tag{34}$$

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 (32) can be a non-trivial and costly task, especially, when the dimensionality is large. In contrast, fixed point iteration schemes like (33) and (34) 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.

Algorithm 8. ECM-VF

Given $K^h(\boldsymbol{k}, \boldsymbol{z})$, $h = 1, ..., N$ for each $(\boldsymbol{k}, \boldsymbol{z})$:
i). Compute $\left(k^h\right)' = K^h(\boldsymbol{k}, \boldsymbol{z})$, $h = 1, ..., N$.
ii). Find $(\boldsymbol{c}, \boldsymbol{\ell})$ satisfying (26), (29) and (30) for each given $\left(\boldsymbol{k}, \boldsymbol{z}, \boldsymbol{k}'\right)$.
iii). Find V satisfying $V(\boldsymbol{k}, \boldsymbol{z}) = \sum_{h=1}^{N} \tau^h u^h\left(c^h, \ell^h\right) + \beta E\left[V\left(\boldsymbol{k}', \boldsymbol{z}'\right)\right]$.
iv). Use V to find $V_h(\boldsymbol{k}, \boldsymbol{z})$ and to infer future values $V_h\left(\boldsymbol{k}', \boldsymbol{z}'\right)$, $h = 1, ..., N$.
v). Compute $\left(\widehat{k}^h\right)' = \frac{\beta E[V_h(\boldsymbol{k}, \boldsymbol{z})]}{V_h(\boldsymbol{k}, \boldsymbol{z})} \frac{\left[\pi^h + z^h f_1^h\left(k^h, \ell^h\right)\right]}{o^h} \left(k^h\right)'$, $h = 1, ..., N$.
The optimal policy functions satisfy $\left(\widehat{k}^h\right)' = \left(k^h\right)'$, $h = 1, ..., N$.

In step ii), we need to compute $(\boldsymbol{c}, \boldsymbol{\ell})$ satisfying (26), (29) and (30) given $(\boldsymbol{k}, \boldsymbol{z}, \boldsymbol{k}')$. This requires us to solve a system of $2N + 1$ equations with $2N + 1$ unknowns $(\boldsymbol{c}, \boldsymbol{\ell})$ and λ. This system can be solved with a standard Newton-style numerical solver but the cost of such a solver may become prohibitive when the dimensionality of the problem increases. Maliar, Maliar and Judd (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.

Algorithm 9. ECM-DVF

Given $K^h(\boldsymbol{k}, \boldsymbol{z})$, $h = 1, ..., N$ for each $(\boldsymbol{k}, \boldsymbol{z})$:
i). Compute $\left(k^h\right)' = K^h(\boldsymbol{k}, \boldsymbol{z})$, $h = 1, ..., N$.
ii). Find $(\boldsymbol{c}, \boldsymbol{\ell})$ satisfying (26), (29) and (30) for each given $\left(\boldsymbol{k}, \boldsymbol{z}, \boldsymbol{k}'\right)$.
iii). Find $V_h(\boldsymbol{k}, \boldsymbol{z}) = u_1^h\left(c^h, \ell^h\right)\left[\pi^h + z^h f_1^h\left(k^h, \ell^h\right)\right]$, $h = 1, ..., N$.
iv). Use V_h to infer future values $V_h\left(\boldsymbol{k}', \boldsymbol{z}'\right)$, $h = 1, ..., N$.
v). Compute $\left(\widehat{k}^h\right)' = \frac{\beta E[V_h(\boldsymbol{k}, \boldsymbol{z})]}{V_h(\boldsymbol{k}, \boldsymbol{z})} \frac{\left[\pi^h + z^h f_1^h\left(k^h, \ell^h\right)\right]}{o^h}\left(k^h\right)'$, $h = 1, ..., N$.
The optimal policy functions satisfy $\left(\widehat{k}^h\right)' = \left(k^h\right)'$, $h = 1, ..., N$.

Again, in step iii), we need to solve the same system of equations as under ECM-VF, i.e., to compute $(\boldsymbol{c}, \boldsymbol{\ell})$ satisfying (26), (29) and (30) given $(\boldsymbol{k}, \boldsymbol{z}, \boldsymbol{k}')$.

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, Maliar and Judd (2011), Judd, Maliar and Maliar (2011, 2012), Maliar and Maliar (2015); 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 of 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.[3] However, GSSA differs from the latter methods in two respects: first, 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 pure simulation methods; see Judd et

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

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, Maliar, Malin and Pichler (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.[4] The utility and production functions are given by

$$u^h\left(c_t^h,\ell_t^h\right)=\frac{\left(c_t^h\right)^{1-1/\gamma^h}}{1-1/\gamma^h}-B^h\frac{\left(\ell_t^h\right)^{1+1/\eta^h}}{1+1/\eta^h},\qquad zf^h\left(k_t^h,\ell_t^h\right)=z^hA\left(k_t^h\right)^{\alpha}\left(\ell_t^h\right)^{1-\alpha}-\delta k^h, \tag{35}$$

where $\left\{\gamma^h,B^h,\eta^h\right\}$ 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, Maliar, Malin and Pichler (2011). 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 $h=1,...,N$, respectively. The steady state level of productivity is normalized to one, $\overline{z}^h=1$. We also normalize the steady state levels of capital and labor to one, $\overline{k}^h=1$, $\overline{\ell}^h=1$, which implies $\overline{c}^h=A$, $\overline{\lambda}=1$ and leads to $A=\frac{1-\beta}{\alpha\beta}$, $\tau^h=u^h\left(A,1\right)$ and $B^h=\left(1-\alpha\right)A^{1-1/\gamma^h}$. 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, respectively. As an initial guess, we use a linear approximation of capital policy function. To solve for the polynomial coefficients, we use fixed point iteration. To solve for consumption and labor satisfying (26), (29) and (30), we use an iteration-on-allocation

[4]Model 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) provide a description of all models studied in the comparison analysis of Kollmann, Maliar, Malin and Pichler (2011).

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, Maliar and Maliar (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 3, 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. We report two accuracy measures: One measure is the size of absolute unit-free residuals across 1,000 draws of state variables located on spheres in the state space (centered at steady state) with radii 0.01, 0.10, and 0.30. Roughly speaking, this measure shows how accurate is our solution when we deviate from the steady state by 1%, 10% and 30%, respectively. The other measure is the size of the residuals on a stochastic simulation of 10,000 observations; this measure shows how accurate our solution in the high-probability area of the state space – the ergodic set. These two accuracy measures are used in the JEDC comparison analysis of Kollmann, Maliar, Malin and Pichler (2011); also see Juillard and Villemot (2011) for more details.

Our main finding is that the 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, Maliar, Malin and Pichler (2011) 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 rapidly 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_1 with a polynomial of one degree less, i.e., we "lose" one polynomial degree.

In turn, the ECM-DVF method is very accurate. It reaches the accuracy frontier

Table 3: Accuracy and speed of ECM-VF in the multicountry model.

Number of countries	Polyn. degree	CPU	r=0.01		r=0.1		r=0.3		Simulation	
			L_1	L_∞	L_1	L_∞	L_1	L_∞	L_1	L_∞
ECM-VF method										
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
ECM-DVF method										
N=2	1^{st}	173	-5.14	-4.24	-4.78	-3.14	-4.00	-2.25	-4.82	-3.01
	2^{nd}	1189	-6.58	-5.73	-6.22	-4.49	-5.22	-3.14	-6.06	-4.21
	3^{rd}	1734	-7.84	-6.95	-7.37	-5.56	-6.00	-4.08	-7.10	-4.93
N=4	1^{st}	531	-5.12	-4.32	-4.65	-3.37	-3.84	-2.48	-4.82	-3.19
	2^{nd}	2039	-6.35	-5.61	-6.22	-4.61	-5.14	-3.53	-6.01	-4.32
	3^{rd}	8092	-7.47	-6.52	-6.95	-5.51	-5.65	-3.98	-6.87	-4.89
N=6	1^{st}	635	-5.10	-4.30	-4.60	-3.36	-3.79	-2.56	-4.83	-3.26
	2^{nd}	2723	-6.71	-5.71	-5.94	-4.48	-4.97	-3.44	-5.88	-4.27
	3^{rd}	38698	-7.28	-6.37	-6.66	-5.12	-5.25	-3.74	-6.61	-4.76
N=8	1^{st}	1071	-5.11	-4.29	-4.58	-3.42	-3.76	-2.62	-4.84	-3.34
	2^{nd}	4541	-6.55	-5.65	-5.69	-4.34	-4.74	-3.31	-5.72	-4.16
	3^{rd}	165911	-7.35	-6.46	-6.42	-4.82	-5.00	-3.40	-6.46	-4.71

Notes: Columns "r=0.01", "r=0.1", "r=0.3" contain the results of accuracy evaluation across 1,000 draws of state variables located on spheres in the state space (centered at steady state) with radii 0.01, 0.10, and 0.30, respectively, and column "Simulation" contains the results of accuracy evaluation on a stochastic simulation of 10,000 observations. The statistics L_1 and L_∞ are, respectively, average and maximum absolute unit-free residuals (in log10 units) across all equilibrium conditions. CPU is the running time (in seconds).

attained in the comparison analysis of Kollmann, Maliar, Malin and Pichler (2011). In particular, in an accuracy check on a stochastic simulation, our third-degree polynomial solutions are more accurate than second-degree polynomial solutions reported in Kollmann, Maliar, Malin and Pichler (2011) 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, Maliar, Malin and Pichler (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.[5]

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 of 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 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. Hopenhayn and Werning (2008) analyzed the optimal financing of an investment project subject to the risk of default and show that the optimal contract

[5]The only method (apart from ECM) that has produced third-degree polynomial solutions to a similar model is a perturbation-based hybrid Euler equation method of Maliar, Maliar and Villemot (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 $10^{-4.69}$.

may allow default along the equilibrium path. 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. Bianchi, Hatchondo and Martinez (2014) used a default risk model for investigating the optimal accumulation of international reserves as a hedge against roll over risk. Aguiar, Amador, Farhi and Gopinath (2013) studied fiscal and monetary policy in a monetary union with the potential for rollover crises in sovereign debt markets. See Aguiar and Amador (2013) for a review of the literature on sovereign debt. The main challenge for this literature, however, is the computational burden of models with default. Computational limitations constitute a substantial obstacle to analyze richer models. We argue that the ECM methods can significantly reduce the computational expense of default risk models.

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

$$\log(y_t) = \rho \log(y_{t-1}) + \varepsilon_t, \tag{36}$$

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. If a borrower defaults, he goes to autarky and gets punished with output loss $y^d \leq y$ in every period spent in autarky. In the future, the borrower re-incorporates in the world economy with exogenously given probability θ.

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

$$\max_{b_{t+1}} \left\{ q_t b_{t+1} - \frac{1-\delta_t}{1+r} b_{t+1} \right\}, \tag{37}$$

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 $q_t \in \left(0, \frac{1}{1+r}\right)$.

A recursive formulation of Arellano's (2008) model. 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 compares these two possibilities and chooses the one that implies higher welfare,

$$V^o(b,y) \equiv \max \left\{ V(b,y), V^d(y) \right\}. \tag{38}$$

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

$$\begin{aligned} V(b,y) &= \max_{b'} \left\{ u(c) + \beta \int \max \left[V(b',y'), V(b', \overline{y}(b')) \right] dF(y') \right\} \\ \text{s.t.} \quad c &= y + b - q(b',y) b', \end{aligned} \tag{39}$$

where F is a distribution function of y' and $q(b',y)$ is the bond-price function that is related to the probability of default $\delta(b',y)$ as $q(b',y) = \frac{1-\delta(b',y)}{1+r}$. If a borrower defaults, his value function V^d is given by

$$V^d(y) = u\left(y^d\right) + \beta E \left[\theta V^o(0,y') + (1-\theta) V^d(y') \right], \tag{40}$$

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

A version of the default risk model with an exogenous default rule. We will also consider a version of default risk model with exogenous default risk rule. Namely, we assume that the borrower's default decision is represented by an exogenous function $\overline{y}(b_t)$ such that an agent with the debt b_t will default whenever the random income y_t falls below the threshold level $y_t < \overline{y}(b_t)$. By using (36), we can compute the probability of default $\delta(b_{t+1}, y_t)$ at $t+1$

$$\delta(b_{t+1}, y_t) = prob(\underbrace{y_t^{\rho} \exp(\varepsilon_t)}_{y_{t+1}} < \overline{y}(b_{t+1})) = F\left[\ln\left(\frac{\overline{y}(b_{t+1})}{y_t^{\rho}}\right)\right], \tag{41}$$

where F is a cumulative distribution function on a normal distribution. Using (41), we can represent the price function $q(b_{t+1}, y_t)$ by

$$q(b_{t+1}, y_t) = \frac{1}{1+r}\left(1 - F\left[\ln\left(\frac{\overline{y}(b_{t+1})}{y_t^{\rho}}\right)\right]\right). \tag{42}$$

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.

Endogenous versus exogenous default rule. There is a relation between Arellano's (2008) model and the default risk model with exogenous default rule. In the model of Arellano (2008), a condition for default is defined implicitly by (38): a borrower defaults for those y for which $V(b, y) < V^d(y)$ given b. 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 $\overline{y}(b)$. In quantitative simulations for more general shock processes and default costs, that paper also contains default decisions that are cutoff rules. Hence, the cutoff rule for default $\overline{y}(b)$ induces the corresponding value function condition $V(b, y) < V^d(y)$ and vice versa. In particular, if we take the cutoff rule $\overline{y}(b)$ that is implied by Arellano's (2008) analysis, we will get the same solution in the models with endogenous and exogenous default rules.

Thus, the model with exogenous default rule is useful for three reasons: First, the ECM analysis requires us to solve such a model as a part of the solution procedure of Arellano's (2008) model with endogenous default rule. Second, the model with exogenous default rule is a convenient setup for testing the performance of the proposed numerical solution methods. Finally, the model with exogenous default rule has interest of its own and can be used as a simpler alternative to the conventional model with endogenous default rule in some applications.

4.2 Envelope condition method

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

$$u'(c)[q_1(b',y)b' + q(b',y)] = \beta E\left[V_1^o(b',y')\right]. \tag{43}$$

The envelope condition, $V_1(b,y) = u'(c)$, in turn implies the following ECM-DVF recursion

$$V_1(b,y) = \frac{\beta E\left[V_1^o(b',y')\right]}{q_1(b',y)b' + q(b',y)} = \frac{\beta \int V_1(b',y')1(y' > \overline{y}(b'))dF(y')}{q_1(b',y)b' + q(b',y)}, \tag{44}$$

where $1(X)$ is an indicator of the event X, and F is a distribution function of y'. The term corresponding to the indicator function $1(y' \leq \overline{y}(b'))$ does not appears in (44) because for $y' \leq \overline{y}(b')$, we have $V_1^o(b',y') = \frac{\partial V^d(y')}{\partial b'} = 0$. Although value function has a kink in the default point, it is never optimal for the agent to reach that point (this fact follows by the generalize envelope theorem of Clausen and Strub (2013)). As a result, the optimal choice satisfies (43) and (44).

4.2.1 ECM-VF for the model with an exogenous default rule

We first show ECM-VF for the default risk model with exogenous default rule (39), (40) and (42).

Algorithm 10. ECM-VF

Fix $\overline{y}(b)$ and choose a set of points (b,y) such that $y > \overline{y}(b)$.
Precompute $q(b',y)$ using (42).
Define $\mathcal{L}(b',y) \equiv q(b',y)b$ and precompute its inverse $\mathcal{L}^{-1}$.
Given $V(b,y)$, for each point (b,y), compute:
  i). $c = u'^{-1}[V_1(b,y)]$.
  ii). $b' = \mathcal{L}^{-1}(b + y - c)$.
  iii). $\widehat{V}(b,y) = u(c) + \beta E\left[\max\left\{V(b',y'), V(b',\overline{y}(b'))\right\}\right]$.
Iterate on i)-iii) until convergence $\widehat{V} = V$.

While we can solve for b' satisfying (39) 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 for the model with an exogenous default rule

We now show ECM-DVF for the default risk model with an exogenous default rule (39), (40) and (42).

Algorithm 11. ECM-DVF

Fix $\overline{y}(b)$ and choose a set of points (b, y) such that $y > \overline{y}(b)$.
Precompute $q(b', y)$ using (42).
Define $\mathcal{L}(b) \equiv q(b)b$ and precompute its inverse $\mathcal{L}^{-1}$.
Given $V_1(b, y)$, for each point (b, y), define:
i). $c = u'^{-1}[V_1(b, y)]$.
ii). $b' = \mathcal{L}^{-1}(b + y - c)$.
iii). $\widehat{V}_1(b, y) = \frac{\beta E\left[V_1^o(b', y')\right]}{q_1(b', y)b' + q(b', y)} = \frac{\beta \int V_1(b', y') 1(y' > \overline{y}(b'))dF(y')}{q_1(b', y)b' + q(b', y)}$.
The optimal value function satisfies $\widehat{V}_1 = V_1$.
Given a converged $\widehat{V}_1$, find $\widehat{V}$ satisfying $\widehat{V}(b, y) = u(c) + \beta E\left[\max\left\{\widehat{V}(b', y'), \widehat{V}(b', \overline{y}(b'))\right\}\right]$.

When implementing ECM-DVF, one needs to be careful not to include grid points for which $q_1(b', y)b' + q(b', y) < 0$. 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 sloped portion of that Laffer curve.

4.2.3 ECM for the model with an endogenous default rule

An algorithm for solving the version of the model (38)-(40) with endogenous default risk is identical to the one used in Arellano (2008) except that the conventional VFI iteration cycle is replaced by ECM-VF and ECM-DVF methods.

To be more specific, given $V(b, y)$ and $V^d(y)$, we first find the cut-off rule $\overline{y}(b)$ satisfying $V(b, y) = V^d(y)$ (according to (38), the agent will default whenever $y < \overline{y}(b)$). Next, for given exogenous default rule $y < \overline{y}(b)$, we solve for the new value function $\widehat{V}$ using either algorithm Algorithms 10 or Algorithm 11, and we update the autarky value function V^d according to (40). If V and V^d converged, we end iteration; otherwise, we proceed to next iteration.

4.3 Numerical analysis

We now construct numerical solutions for the default risk model. We first report the results for the test model with an exogenous default rule, and we then study Arellano's (2008) model with an endogenous default rule.

4.3.1 The model with an exogenous default rule

As an example, we consider a simple default rule $y_t \leqslant \bar{y}(b_t) \equiv \underline{n} - b_t$, 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 $\underline{n} = \{0.65, 0.75, 0.95\}$ that imply that the borrower's debt rises above 0.35, 0.25 and 0.05 of an average period's income, respectively. We assume that the income in (36) is i.i.d, $y \sim \mathcal{N}(0, 0.05)$. Then, the probability of default (41) is given by

$$\delta\left(b_{t+1}, y_t\right) = prob(y_{t+1} \leqslant \bar{y}(b_{t+1})) = F(\underline{n} - b_{t+1}),$$

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 $u\left(c\right) = \frac{c^{1-\gamma}-1}{1-\gamma}$ with $\gamma = 2$, and we fix $\beta = 0.94$.

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. Grid points are Chebyshev extrema scaled so that $b_1 = -0.18$, $b_{100} = 0.40$. (As an alternative, we tried to use grid points that are uniformly spaced and we find that it leads to comparable and slightly lower accuracy measures). 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 construct policy functions using FOC. We find that ECM-VF is less stable numerically in the default risk model and requires a sufficiently accurate initial guess for convergence, while ECM-DVF is more robust to the choice of an initial guess. For all the 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 C.

In Table 4, 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 methods take both about 5.5 seconds; hence, we observe about 50x speedups or higher. As in all previous numerical

Table 4: 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, repectively, 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).

experiments, the accuracy of ECM-VF is somewhat lower than that of ECM-DVF as was in all the previous experiments.

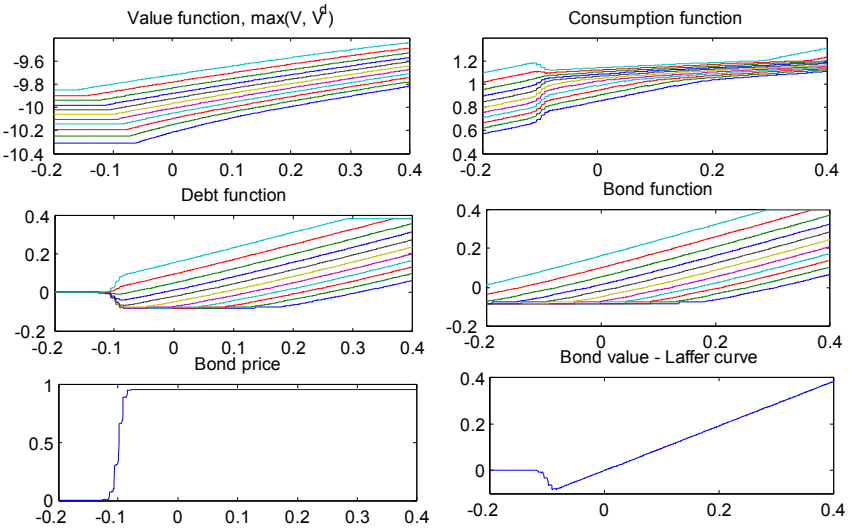
4.3.2 The model with endogenous default rule

We next apply ECM for solving Arellano's (2008) model with an endogenous default rule. In such a model, the convergence of ECM was more difficult to attain. Specifically, a simultaneous iteration on both value functions and endogenous bond price function led to numerical errors that accumulate along iterations, resulting in non-monotone policy functions. Significant damping was needed to stabilize explosive iteration.

We find that the best use of the ECM method in the context of the given problem is to refine a low accuracy solution produced by a version of VFI based on discretization. To be specific, we discretize the domain for bonds into 1000 equally spaced grid points $b_1, ..., b_{1000}$, where $b_1 = -0.18$ and $b_{1000} = 0.40$, and we solve for value function by finding maximum of the right side of Bellman equation (39) across a finite number of discretized bond holdings. The running time was low, 22 seconds, but the accuracy of the solution was also low.

We show the constructed value function, policy functions and prices produced by

the conventional VFI discretization method in Figure 1.



Remarks: The domain for assets is discretized into 1000 uniformly spaced grid points. The domain for income is discretized into 11 Gauss Hermite quadrature states shown in the upper 4 panels.

Figure 1. Discretization method: value function, policy functions and prices.

As is seen from Figure 1, the constructed value and decision functions have the form of step functions. This indicates that the discretization method produces considerable approximation errors. Indeed, the average and maximum unit-free residuals in the Bellman equation on a solution domain (39) are -1.84 and -0.84, respectively, indicating that the approximation errors can be as large as 15% for the VFI method based on discretization.

We next refine the VFI solution using ECM-VF implemented on the same 1000 grid points. The running time for the ECM method was approximately 8 seconds. In Figure 2, we plot the constructed value function, policy functions and prices of

the ECM solution.

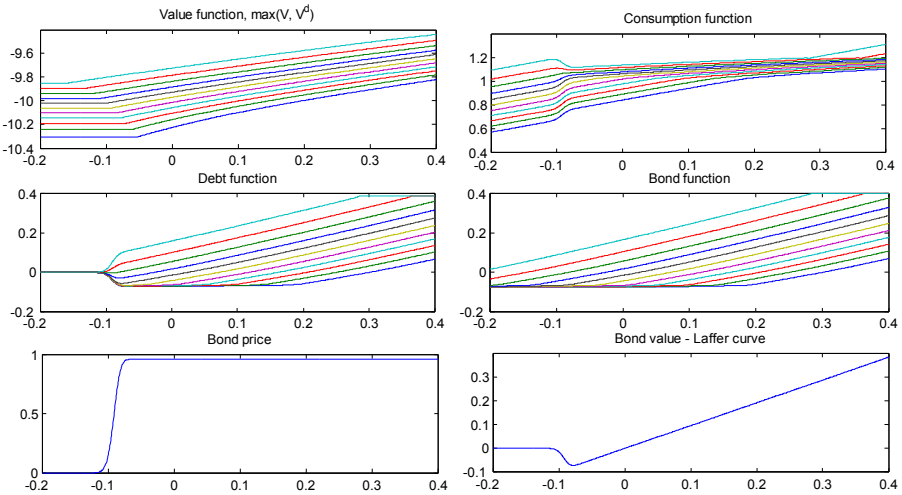


Figure 2. Envelope condition method: value function, policy functions and prices.

The step functions are not appreciated for the ECM solution. This is because the constricted policy and value functions are accurately interpolated off the grid points using cubic splines. The average and maximum unit-free residuals in the Bellman equation on a solution domain (39) are -4.85 and -3.09 after the refinement, respectively. This indicates that the residuals are of order 0.1% , which is about 100 times smaller than those produced by the VFI discretization method. It would be possible to attain higher accuracy levels under conventional VFI if a finer grid is used, for example, 100,000 points instead of 1,000 points but memory of our computer was insufficient to run such experiments. In contrast, ECM was feasible and required just few seconds to produce highly accurate solution.

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. Moreover, some applications require solving a sequence of similar economic models a large number of times, for example, nested fixed point methods for econometric estimation as in Fernández-Villaverde and Rubio-Ramírez (2007) and extended function path (EFP) framework of Maliar, Maliar, Taylor, and Tsener (2015) for analyzing nonstationary and unbalanced growth models.

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 expensive VFI based on FOCs or direct maximization with a cheap forward-style ECM 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 produce 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 with exogenous default rules. These are promising results given a high computational expense of default risk models. However, the convergence of ECM methods was hard to achieve in a default risk model with endogenous default rules. Further research is needed to enhance the convergence properties of the ECM methods in this class of models.

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] Aguiar, M., M. Amador, E. Farhi and G. Gopinath, (2015). Coordination and crisis in monetary unions. Manuscript.

[4] Arellano, C. (2008). Default risk and income fluctuations in emerging economies. American Economic Review, 98(3), 690–712.

[5] Arellano, C. and A. Ramanarayanan, (2012). Dafault and the maturity structure in sovereign bonds. Journal of Political Economy 120/2, 187-232.

[6] Arellano, C., Y. Bai, and P. Kehoe (2013). Financial frictions and fluctuations in volatility. Federal Reserve Bank of Minneapolis Staff Report 466.

[7] Aruoba, S. and F. Schorfheide, (2013). Macroeconomic dynamics near ZLB: a tale of two equilibria. NBER working paper 19248.

[8] 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.

[9] Barillas, F. and J. Fernandez-Villaverde, (2007). A generalization of the endogenous grid method. Journal of Economic Dynamics and Control 31, 2698-2712.

[10] Bellman, R, (1957), Dynamic Programming. Princeton University Press, Princeton, N.J.

[11] Bianchi, J., J.C.Hatchondo and L. Martinez, (2009), International Reserves and Rollover Risk. Manuscript.

[12] Blackwell, D. (1965). Discounted dynamic programming. The Annals of Mathematical Statistics 36, 226-235.

[13] Carroll, K. (2005). The method of endogenous grid points for solving dynamic stochastic optimal problems, Economic letters 91, 312-320.

[14] Chatterjee, S. and B. Eyigundor, (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.

[15] 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.

[16] Clausen, A. and C. Strub, (2013). A general and intuitive envelope theorem. Manuscript.

[17] Den Haan, W. (1990). The optimal inflation path in a Sidrauski-type model with uncertainty. Journal of Monetary Economics 25, 389-409.

[18] Den Haan, W. and A. Marcet, (1990). Solving the stochastic growth model by parameterized expectations. Journal of Business and Economic Statistics 8, 31-34.

[19] 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.

[20] Eaton, J., and M. Gersovitz (1981). Debt with potential repudiation: theoretical and empirical analysis. Review of Economic Studies, 48(2): 289–309.

[21] Fella, G., (2014). A generalized endogenous grid method for non-smooth and non-concave problems, The Review of Economic Dynamics, 17/2, 329-344.

[22] 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.

[23] Fernández-Villaverde, J. and J. Rubio-Ramírez, (2007). Estimating macroeconomic models: A likelihood approach. Review of Economic Studies 74, 1059-1087.

[24] Fernández-Villaverde, J., G. Gordon, P. Guerrón-Quintana, and J. Rubio-Ramírez, (2012). Nonlinear adventures at the zero lower bound. NBER working paper 18058.

[25] Fukushima, K. and Y. Waki, (2011). A polyhedral approximation approach to concave numerical dynamic programming. Manuscript.

[26] Gust, C., D. Lopez-Salido and M. E. Smith, (2012). The empirical implications of the interest-rate lower bound. Manuscript, Federal Reserve Board.

[27] 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.

[28] Hopenhayn and Werning (2008), Equilibrium default. Manuscript.

[29] Howard, R. (1960), Dynamic programming and Markov processes. Cambridge, MA: MIT Press.

[30] Ishakov, F., J. Rust and B. Schjerning, (2012). Extending endogenous grid method for solving discrete continuous sequential decision problems. Manuscript.

[31] Judd, K., (1998). Numerical Methods in Economics. Cambridge, MA: MIT Press.

[32] Judd, K., L. Maliar and S. Maliar, (2011). Numerically stable and accurate stochastic simulation approaches for solving dynamic models. Quantitative Economics 2, 173-210.

[33] Judd, K., L. Maliar and S. Maliar, (2012). Merging simulation and projection approaches to solve high-dimensional problems. NBER working paper 18501.

[34] Judd, K., L. Maliar, S. Maliar and R.Valero (2014). Smolyak Method for Solving Dynamic Economic Models: Lagrange Interpolation, Anisotropic Grid and Adaptive Domain. Journal of Economic Dynamic and Control 44(C), 92-123.

[35] 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.

[36] 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.

[37] 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.

[38] Krueger, D. and F. Kubler, (2004). Computing equilibrium in OLG models with production. Journal of Economic Dynamics and Control 28, 1411-1436.

[39] Krueger, D. (2012). Macroeconomic theory. Manuscript.

[40] Maliar, L. and S. Maliar, (2005). Solving nonlinear stochastic growth models: iterating on value function by simulations. Economics Letters 87, 135-140.

[41] Maliar, L. and S. Maliar, (2013). Envelope condition method versus endogenous grid method for solving dynamic programming problems, manuscript.

[42] 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.

[43] Maliar, L. and S. Maliar, (2015). Merging Simulation and Projection Approaches to Solve High-Dimensional Problems with an Application to a New Keynesian model, Quantitative Economics 6, 1-47.

[44] Maliar, L., S. Maliar and F. Pérez-Sebastián (2008). Sovereign Risk, FDI Spillovers, and Economic Growth. Review of International Economics 16/3, 463-477.

[45] 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.

[46] Maliar, L., Maliar, S. and S. Villemot, (2013). Taking perturbation to the accuracy frontier: a hybrid of local and global solutions. Computational Economics, 42/3, 307-325.

[47] Maliar, L., S. Maliar, J.B. Taylor and I. Tsener, (2015). A tractable framework for analyzing a class of nonstationary markov models. Manuscript.

[48] 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.

[49] 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.

[50] Moreira, H. and W. Maldonado (2003). A contractive method for computing the stationary solution to the Euler equation. Economics Bulletin 3, 1-14.

[51] Pal, J. and J. Stachurski, (2013). Fitted value function iteration with probability one contractions. Journal of Economic Dynamics and Control 37, 251-264.

[52] 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.

[53] Powell, W., (2011). Approximate Dynamic Programming. Wiley: Hoboken, New Jersey.

[54] Rust, J., (2008). Dynamic programming. In The New Palgrave Dictionary of Economics (S. Durlauf and L. Blume eds.), Palgrave Macmillan.

[55] 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.

[56] Santos, M., (2000). Accuracy of numerical solutions using the Euler equation residuals. Econometrica 68, 1377-1402.

[57] Santos, M. and J. Rust, (2008). Convergence properties of policy iteration. SIAM Journal on Control and Optimization 42/6, 2094-2115.

[58] Stachurski, J., (2008). Continuous state dynamic programming via nonexpansive approximation. Computational Economics 31, 141-160.

[59] Stachurski, J., (2009). Economic Dynamics: Theory and Computation. Cambridge: MIT Press.

[60] Stokey, N. L. and R. E. Lucas Jr. with E. Prescott, (1989). Recursive Methods in Economic Dynamics. Cambridge, MA: Harvard University Press.

[61] Tsyrennikov (2013). Fiscal policy, sovereign debt and default with model misspecification, Cornell University, Manuscript.

[62] 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. Finally, in Appendix C, 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).

Algorithm 1. ECM-VF (with implementation details)

Initialization.

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

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

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

i). $c_m = \left[\frac{V_1\left(k_m, z_m; a^{(n)}\right)}{1-\delta+z\alpha k_m^{\alpha-1}}\right]^{-1/\gamma}$.
ii). $k'_m = (1-\delta) k_m + z_m k_m^{\alpha} - c_m$.
iii). $v_m = \frac{c_m^{1-\gamma}-1}{1-\gamma} + \beta \sum_{j=1}^{J} \omega_j V\left(k'_m, z_m^{\rho} \exp(\varepsilon_j); a^{(n)}\right)$.

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

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

Step 3. Convergence check and fixed-point iteration.

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

Otherwise, use damping to compute $a^{(n+1)} = (1-\xi) a^{(n)} + \xi \widehat{a}$ 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).

Algorithm 6. ECM-DVF (with implementation details)

Initialization.

i). Choose an approximating function $V_1(\cdot; a) \approx V_1$.

...

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

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

...

iii). $d_m = \beta \left[1 - \delta + \alpha z k_m^{\alpha-1}\right] \sum_{j=1}^{J} \omega_j V_1\left(k'_m, z_m^{\rho} \exp(\varepsilon_j); a^{(n)}\right)$.

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

Run a regression to find $\widehat{a} = \arg\min_a \sum_{m=1}^{M} \|d_m - V_1(k_m, z_m; a)\|$.

...

5.1 Appendix B: ECM for multicountry model

We now describe the ECM-VF method that solves for value function in multicountry model (25)–(27).

Algorithm 8. ECM-VF (with implementation details)

Initialization.

i). Choose approximating functions $K^h\left(\cdot; a^h\right) \approx K^h$, $h = 1, ..., N$ and $V\left(\cdot; \varpi\right) \approx V$.

ii). Choose integration nodes, $\boldsymbol{\varepsilon}_j = \left(\varepsilon_j^1, ..., \varepsilon_j^N\right)$, and weights, ω_j, $j = 1, ..., J$.

iii). Fix the simulations length T and the initial condition $(\boldsymbol{k}_0, \boldsymbol{z}_0)$.

iv). Draw and fix a sequence of productivity levels $\{\boldsymbol{z}_t\}_{t=1,...,T}$ using (27).

v). Construct integration nodes, $\boldsymbol{z}_{t+1,j} = \left(z_{t+1,j}^h, ..., z_{t+1,j}^h\right)$ with $z_{t+1,j}^h = \left(z_t^h\right)^\rho \exp\left(\varepsilon_j^h\right)$.

vi). Make an initial guess on $\left(a^1\right)^{(1)}, ..., \left(a^h\right)^{(1)}$.

Iterative cycle. At iteration n, given $\left(a^1\right)^{(n)}, ..., \left(a^h\right)^{(n)}$, perform the following steps.

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

i). Use $k_{t+1}^h = \widehat{K}^h\left(\boldsymbol{k_t}, \boldsymbol{z}_t; \left(a^h\right)^{(n)}\right)$, $h = 1, ..., N$, to recursively calculate $\{\boldsymbol{k}_{t+1}\}_{t=0,...,T}$.

ii). Compute $\{\boldsymbol{c}_t, \boldsymbol{\ell}_t\}_{t=0,...,T}$ satisfying (26), (29) and (30) given $\{\boldsymbol{k}_t, \boldsymbol{z}_t, \boldsymbol{k}_{t+1}\}_{t=0,...,T}$.

iii). Find $\widehat{\varpi}$ satisfying $V\left(\boldsymbol{k}_t, \boldsymbol{z}_t; \widehat{\varpi}\right) = \sum_{h=1}^N \tau^h u^h\left(c_t^h, \ell_t^h\right) + \beta \sum_{j=1}^J V\left(\boldsymbol{k}_{t+1}, \boldsymbol{z}_{t+1,j}; \widehat{\varpi}\right)$.

iv). Use $V\left(\cdot; \widehat{\varpi}\right)$ to find $V_h\left(\boldsymbol{k}_t, \boldsymbol{z}_t; \widehat{\varpi}\right)$ and to infer $V_h\left(\boldsymbol{k}_{t+1}, \boldsymbol{z}_{t+1,j}; \widehat{\varpi}\right)$ for $j = 1, ..., J$.

v). Compute $\widehat{k}_{t+1}^h \equiv \sum_{j=1}^J \beta \frac{V_h\left(\boldsymbol{k}_{t+1}, \boldsymbol{z}_{t+1,j}; \widehat{\varpi}\right)}{V_h\left(\boldsymbol{k}_t, \boldsymbol{z}_t; \widehat{\varpi}\right)} \frac{\left[\pi_t^h + z_t^h f_1^h\left(k_t^h, \ell_t^h\right)\right]}{o_t^h} k_{t+1}^h$, $h = 1, ..., N$.

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

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

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

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

Otherwise, use damping to compute $\left(a^h\right)^{(n+1)} = (1 - \xi)\left(a^h\right)^{(n)} + \xi \widehat{a}^h$ 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).

Algorithm 9. ECM-DVF (with implementation details)

Initialization.

i). Choose approximating functions $K^h\left(\cdot;a^h\right)\approx K^h$ and $V_h\left(\cdot;\varpi^h\right)\approx V_h$, $h=1,...,N$.

...

Iterative cycle. At iteration n, given $\left(a^1\right)^{(n)},...,\left(a^h\right)^{(n)}$*, perform the following steps.*

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

...

iii). Find $d_t^h\equiv u_1^h\left(c_t^h,\ell_t^h\right)\left[\pi_t^h+z_t^h f_1^h\left(k_t^h,\ell_t^h\right)\right]$ and
find $\widehat{\varpi}^h\equiv\arg\min_{\varpi^h}\left\|d_t^h-V_h\left(\boldsymbol{k}_t,\boldsymbol{z}_t;\varpi^h\right)\right\|$, $h=1,...,N$;

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

...

5.2 Appendix C: ECM for default risk model

We show the ECM-VF method that solves for value function in default risk model (38)-(39).

Algorithm 10. ECM-VF (with implementation details)

Initialization.

i). Choose an approximating function $V(\cdot; a) \approx V$.
ii). Choose integration nodes, ε_j, and weights, ω_j, $j = 1, ..., J$.
iii). Construct a grid $\{b_m, y_m\}_{m=1,...,M}$ covering the area $y > \overline{y}(b)$.
iv). Compute $q(b', y)$ using (42).
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 n, given $a^{(n)}$, perform the following steps.

Step 1. For $m = 1, ..., M$, compute:
i). $c_m = V_1\left(b_m, z_m; a^{(n)}\right)^{-1/\gamma}$.
ii). $b'_m = \mathcal{L}^{-1}(b_m + y_m - c_m)$.
iii). $v_m = \frac{c_m^{1-\gamma}-1}{1-\gamma} + \beta \sum_{j=1}^{J} \omega_j \max\left\{V\left(b'_m, y_m^{\rho} \exp(\varepsilon_j); a^{(n)}\right) V\left(b'_m, \overline{y}(b_m); a^{(n)}\right)\right\}$

Step 2. Computation of a that fits the values v_m on the grid.
Run a regression to find $\widehat{a} = \arg\min_a \sum_{m=1}^{M} \|v_m - V(k_m, z_m; a)\|$.

Step 3. Convergence check and fixed-point iteration.
Stop if $\max\left|v_m^{(n+1)} - v_m^{(n)}\right| < 10^{-4}$, where $\xi = 0.1$ is a damping parameter.
Otherwise, use damping to compute $a^{(n+1)} = (1-\xi)a^{(n)} + \xi\widehat{a}$ and go to Step 1.

We next describe ECM-DVF that solves for the derivatives of value function in default risk model (38)-(39) (the steps that are identical to those in ECM-VF are omitted).

Algorithm 11. ECM-DVF (with implementation details)

Initialization.

i). Choose an approximating function $V_1(\cdot; a) \approx V_1$.

...

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

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

...

iii). $d_m = \frac{\beta \sum_{j=1}^{J} \omega_j V_1\left(b'_m, y_m^{\rho} \exp(\varepsilon_j); a^{(n)}\right) 1(y_m^{\rho} \exp(\varepsilon_j) > \overline{y}(b'_m))}{q_1(b'_m, y_m) b'_m + q(b'_m, y_m)}$.

...

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

Run a regression to find $\widehat{a} = \arg\min_a \sum_{m=1}^{M} \|d_m - V_1(k_m, z_m; a)\|$.

...

We next describe conventional VFI that solves for value function in default risk model (38)-(39) (the steps that are identical to those in ECM-VF are omitted).

Algorithm 12. FOC-VFI (with implementation details)

...

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

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

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

...