Dynare Working Papers Series
http://www.dynare.org/wp/

Accelerating the resolution of sovereign debt models using an endogenous grid method

Sébastien Villemot

Working Paper no. 17

November 2012

CEPREMAP
CENTRE POUR LA RECHERCHE ECONOMIQUE ET SES APPLICATIONS
142, rue du Chevaleret — 75013 Paris — France
http://www.cepremap.ens.fr

Accelerating the resolution of sovereign debt models using an endogenous grid method

Sébastien Villemot*

November 22, 2012

Abstract

State-of-the-art algorithms for solving sovereign debt models with endogenous default rely on value function iterations. These algorithms are consequently very slow and quickly become intractable, even for a state space of dimension as low as three. This paper shows how to adapt the endogenous grid method for sovereign debt models, leading to a dramatic speed gain by a factor comprised between 5 and 10. A second contribution is to quantify and compare the accuracy of the computed solutions by both the value function iterations and the endogenous grid method.

Keywords: sovereign debt; endogenous default; endogenous grid method; solution accuracy

JEL Classification: F34; C63

1 Introduction

The choice of a numerical solution method is an important decision when studying a sovereign debt model. As shown by Hatchondo et al. (2010), an imprecise method can lead to significant numerical errors over the solution of sovereign debt models, up to the point where some of the main conclusions of innovative papers turn out to be just wrong. Having a precise enough method at one's disposal is therefore critical.

But the economist faces a trade-off: a more precise method usually requires a higher computing time, and is often (though not always) more difficult to implement. In the field of sovereign debt models, the speed-accuracy frontier of solution methods is particularly unfavorable compared to other classes of models (such as the family of RBC and DSGE models).[1] Indeed, RBC/DSGE models benefit from fast and advanced techniques based on first order conditions, while sovereign debt models have been so far limited to the slower value function iteration procedure (hereafter referred to as VFI). The main reason for this situation is that sovereign debt models cannot be entirely specified in terms of first order conditions since the default decision involves a comparison between two value functions. Therefore standard DSGE techniques do not apply and alternative techniques have to be used.

In this paper I present a new method for solving sovereign debt models which significantly improves the existing speed-accuracy frontier. This method is an adaptation to sovereign debt models of the *endogenous grid method* (hereafter referred to as EGM) introduced by Carroll

*CEPREMAP. E-mail: sebastien@dynare.org

[1] See for example Aruoba et al. (2006) and Kollmann et al. (2011) for overviews of recent solution techniques for RBC/DSGE models and for characterizations of the current speed-accuracy frontier.

(2006) and extended by Barillas and Fernández-Villaverde (2007). I call the new method the "doubly endogenous grid method" (hereafter referred to as 2EGM). As a second contribution, I explore the accuracy of solution methods (whether VFI or 2EGM) in a more systematic way than previously done in the literature on sovereign debt models, by applying tests based on the Euler errors. The main result of the present paper is that both VFI and 2EGM are capable of delivering accurate solutions for the canonical sovereign debt model, but that 2EGM is much faster (by a factor between 5 and 10) than VFI for a comparable level of accuracy.

This paper is organized as follows. Section 2 presents a canonical model of sovereign debt used as a basis for illustration and experiments. Section 3 reviews the existing solution methods for sovereign debt models and briefly discusses their respective advantages. Section 4 presents the doubly endogenous grid method. Section 5 assesses the accuracy of both VFI and 2EGM on the canonical model of section 2. Section 6 presents a similar exercise on a larger model. Section 7 concludes. Appendix A gives some additional implementation details.

2 A canonical sovereign debt model

In this section I describe a "canonical" model of sovereign debt that is used as a support for describing the VFI and 2EGM methods and for computing the first set of numerical results. The main characteristics of this model were already described in Eaton and Gersovitz (1981) and Cohen and Sachs (1986). The exact version that I present here is model II of Aguiar and Gopinath (2006).

2.1 The economy

A sovereign country is inhabited by a representative consumer, who is able to tilt consumption away from output by borrowing or lending on the international financial markets. Output produced at time t is exogenous and given by the random variable $\tilde{y}_t$, which follows a non stationary Markovian process (in the sequel, non stationary variables—as well as functions of those variables—will be denoted with a tilde, like in $\tilde{x}$; the detrended counterpart of $\tilde{x}$ will be denoted x, without the tilde). More precisely, output is given by the following law of motion:

$$\tilde{y}_t = g_t\,\tilde{y}_{t-1} \tag{1}$$

$$\log(g_t) = (1-\rho_g)\left(\log(\mu_g) - \frac{\sigma_g^2}{2(1-\rho_g^2)}\right) + \rho_g\log(g_{t-1}) + \varepsilon_t^g \tag{2}$$

where $\rho_g \in [0,1)$, $\varepsilon_t^g \rightsquigarrow \mathcal{N}(0,\sigma_g^2)$.

The world financial markets are characterized by a constant riskless rate of interest r. Lenders are risk-neutral and subject to a zero-profit condition by competition. Debt is short-term and needs to be refinanced at every period.

At any time t, the country has accumulated foreign assets $\tilde{a}_t$ and may decide to default upon it (only if $\tilde{a}_t < 0$). When it does so, the country suffers forever after a negative productivity shock. One can say that default creates a panic that destroys capital either through an exchange-rate or a banking crisis. Post default output is therefore assumed to be:

$$\tilde{y}_t^B = (1-\delta)\tilde{y}_t$$

where the B superscript stands for "bad times," and $\delta \in [0,1)$ captures the magnitude of the default penalty on output. As another cost, the country is temporarily constrained to financial autarky; at every period, it is redeemed with probability λ and then recovers access to financial markets with its previous debts canceled.

2.2 Financial markets

The timing of events is as follows. First assume that the country has incurred a debt obligation falling due at time t (if $\tilde{a}_t < 0$) or has accumulated assets (if $\tilde{a}_t > 0$), and is currently not excluded from financial markets. At the beginning of period t the country learns the value of its output $\tilde{y}_t$. Then it decides to default or to reimburse its debt.

If the debt is reimbursed in full, the country can continue to trade bonds and chooses a new amount of assets $\tilde{a}_{t+1}$ which must be repaid at time $t+1$. Given the demand $\tilde{a}_{t+1}$ of the country for assets due tomorrow, the supply function of the international investors is the price $\tilde{q}(\tilde{y}_t, \tilde{a}_{t+1})$ at which they are willing to trade these short-term bonds. The implicit interest rate associated to this supply function is $1/\tilde{q}(\tilde{y}_t, \tilde{a}_{t+1}) - 1$; it will be equal to r if the bonds are considered riskless by the investors, and greater to that if there is a perceived risk of default. The risk premium Δ_t can therefore be expressed as:

$$\Delta_t = \frac{1}{(1+r)\tilde{q}(\tilde{y}_t, \tilde{a}_{t+1})} - 1 \tag{3}$$

Such financial agreements being concluded, the country eventually consumes, in the event it services its debt in full:

$$\tilde{c}_t^G = \tilde{y}_t + \tilde{a}_t - \tilde{q}(\tilde{y}_t, \tilde{a}_{t+1})\tilde{a}_{t+1} \tag{4}$$

where the G superscript stands for "good times." Alternatively, in the event of a debt crisis the country's consumption is nailed down to:

$$\tilde{c}_t^B = \tilde{y}_t^B = (1-\delta)\tilde{y}_t$$

2.3 Preferences

The decision to default or to stay current on the financial markets involves a comparison of two paths that imply expectations over the entire future. Let's call β the discount factor and u the instantaneous utility function of the representative agent of the country (assumed to be a constant relative risk aversion (CRRA) function):

$$u(\tilde{c}_t) = \frac{\tilde{c}_t^{1-\gamma}}{1-\gamma}$$

where γ is the coefficient of relative risk aversion.

Let's call $\tilde{V}^G$ (resp. $\tilde{V}^B$) the country's payoff conditional to repayment (resp. to default), and $\tilde{V}$ the country's unconditional payoff. In recursive form, those functions satisfy:

$$\tilde{V}(\tilde{a}_t, \tilde{y}_t) = \max\{\tilde{V}^G(\tilde{a}_t, \tilde{y}_t), \tilde{V}^B(\tilde{y}_t)\} \tag{5}$$

$$\tilde{V}^G(\tilde{a}_t, \tilde{y}_t) = \max_{\tilde{a}_{t+1}} \left\{ u(\tilde{y}_t + \tilde{a}_t - \tilde{q}(\tilde{y}_t, \tilde{a}_{t+1})) + \beta\, \mathbb{E}_t \tilde{V}(\tilde{a}_{t+1}, \tilde{y}_{t+1}) \right\} \tag{6}$$

$$\tilde{V}^B(\tilde{y}_t) = u((1-\delta)\tilde{y}_t) + \beta\, \mathbb{E}_t \left[(1-\lambda)\tilde{V}^B(\tilde{y}_{t+1}) + \lambda\, \tilde{V}(0, \tilde{y}_{t+1}) \right] \tag{7}$$

The default decision function:

$$\tilde{D}(\tilde{a}_t, \tilde{y}_t) = \mathbb{1}_{\tilde{V}^G(\tilde{a}_t, \tilde{y}_t) < \tilde{V}^B(\tilde{y}_t)} \tag{8}$$

is equal to 1 in case of default and 0 in case of repayment.

Finally, the investors are assumed to be risk neutral and in perfect competition, which implies that their credit supply function must satisfy the following zero profit condition:

$$(1+r)\tilde{q}(\tilde{y}_t, \tilde{a}_{t+1}) = \mathbb{E}_t \left[1 - \tilde{D}(\tilde{a}_{t+1}, \tilde{y}_{t+1}) \right] \tag{9}$$

2.4 Equilibrium and basic properties

In the following definition, time subscripts are dropped because I focus on recursive equilibria. A quote on a variable name (like a') designates a next period variable.

Definition 1 (Recursive equilibrium) *A recursive equilibrium for this economy is given by a price function $\tilde{q}(\tilde{y}, \tilde{a'})$, value functions $\tilde{V}(\tilde{a}, \tilde{y})$, $\tilde{V}^G(\tilde{a}, \tilde{y})$, $\tilde{V}^B(\tilde{y})$, and a default decision function $\tilde{D}(\tilde{a}, \tilde{y})$ such that:*

- *Given the credit supply function, the value functions and the default decision function satisfy the government optimization problem (5)–(8);*
- *Given the default decision function, the credit supply function satisfies the zero profit condition (9).*

The choice function for tomorrow's assets (conditionally to repayment today) is denoted $\tilde{a}'(\tilde{a}, \tilde{y})$.

Note that since the model exhibits a stochastic growth trend, it is necessary to normalize some variables (GDP, debt levels, value functions) in order to compute the numerical solution. To detrend, I normalize all variables at date t by the following factor:

$$\tilde{\Gamma}_t = \mu_g \tilde{y}_{t-1} \tag{10}$$

The detrended variables are denoted without a tilde. For example the detrended debt is $a_t = \frac{\tilde{a}_t}{\tilde{\Gamma}_t}$; note that the detrended debt level is almost equal to the debt-to-GDP ratio (up to the deviation of current growth rate to its mean).[2] Also, one has $g_t = y_t \mu_g$, and this proportional relationship between y and g is used implicitly at several places in this paper. The detrended equations of the model are the following:

$$\begin{aligned}
V(a_t, y_t) &= \max\{V^G(a_t, y_t), V^B(y_t)\} \\
V^G(a_t, y_t) &= \max_{a_{t+1}} \left\{ u(y_t + a_t - q(y_t, a_{t+1}) a_{t+1}\, g_t) + \beta\, g_t^{1-\gamma}\, \mathbb{E}_t V(a_{t+1}, y_{t+1}) \right\} \\
V^B(y_t) &= u((1-\lambda) y_t) + \beta\, g_t^{1-\gamma}\, \mathbb{E}_t \left[(1-\lambda) V^B(y_{t+1}) + \lambda\, V(0, y_{t+1}) \right] \\
(1+r) q(y_t, a_{t+1}) &= \mathbb{E}_t \left[1 - D(a_{t+1}, y_{t+1}) \right]
\end{aligned} \tag{11}$$

where the default decision function is:

$$D(a_t, y_t) = \mathbb{1}_{V^G(a_t, y_t) < V^B(y_t)}$$

Finally, Table 1 gives the calibration of the model which is used in the numerical exercises.

3 Solution methods: the state of the art

Solving the sovereign debt model presented in the previous section consists in computing the value functions V^G and V^B and the price function q. The value function V and the policy functions a' and D are then trivial to deduce given the others.

Since in the general case these functions do not have a closed form solution, the economist is only able to compute approximations of them, which I will denote by $\hat{V}^G$, $\hat{V}^B$ and $\hat{q}$ (more generally, in the following, $\hat{X}$ designates the numerical approximation to X).

I briefly describe the value function iteration (VFI) technique below:[3]

[2] The fact that the current growth g_t does not enter $\tilde{\Gamma}_t$ guarantees that if $\tilde{X}_t$ is in the information set at date $t-1$, then so is X_t.

[3] A similar description can be found in the appendix of Hatchondo et al. (2010).

Table 1: Calibration of the canonical model

Parameter	Symbol	Value
Mean gross growth rate	μ_g	1.006
Auto-correlation of the growth rate	ρ_g	0.17
Innovation variance of the growth rate	σ_g	3%
Loss of output in autarky (% of GDP)	δ	2%
Probability of settlement after default	λ	10%
World riskless interest rate	r	1%
Discount factor	β	0.8
Risk aversion	γ	2

1. Define a finite grid of points $(a_i, y_j)_{(i,j)\in I\times J}$ (where I and J are finite indexing sets), which will be used for interpolating $\hat{V}^G$ and $\hat{V}^B$.

2. Let n be the iterations counter and start with $n = 0$. Choose initial values $\hat{V}^{G,(0)}$ and $\hat{V}^{B,(0)}$ for the value functions (see section A.2 for a discussion on the choice of those initial values). Let $\hat{V}^{(0)} = \max\{\hat{V}^{G,(0)}, \hat{V}^{B,(0)}\}$.

3. At each point of the grid, compute the value functions $\hat{V}^{G,(n+1)}$ and $\hat{V}^{B,(n+1)}$ for the next iteration by solving equations (6) and (7) recursively:

$$\hat{V}^{G,(n+1)}(a_i, y_j) = \max_{a'} \left\{ u(y_j + a_i - \hat{q}^{(n+1)}(y_j, a')a'g_j) + \beta g_j^{1-\gamma} \int \hat{V}^{(n)}(a', y')\mathrm{d}F(y'|y_j) \right\} \tag{12}$$

$$\hat{V}^{B,(n+1)}(y_j) = u((1-\delta)y_j) + \beta g_j^{1-\gamma} \int \left[(1-\lambda)\hat{V}^{B,(n)}(y') + \lambda \hat{V}^{(n)}(0, y')\right] \mathrm{d}F(y'|y_j)$$

   where F is the cumulative distribution function of tomorrow's output conditional to today's output.

   Note that this step involves the computation of an integral[4] and a function maximization. The price function $\hat{q}^{(n+1)}$ also needs to be computed, and this can be done using one of the two alternative ways that are described further below.

4. If $\hat{V}^{(n+1)}$ is close enough to $\hat{V}^{(n)}$ (up to some target accuracy level), then stop. Otherwise, set $n = n + 1$ and go to step 3.

This algorithm has a nice intuitive interpretation: it is equivalent to solving the finite horizon problem whose number of periods is equal to the number of iterations in the algorithm. The solution for the infinite horizon problem is therefore approximated by the solution to a finite horizon problem with a large enough number of periods.

Moreover, this algorithm is known to converge because it closely replicates the constructive proof of the existence of an equilibrium (which itself relies on the well-known contraction fixed point theorem, see the proof in appendix of Aguiar and Gopinath, 2004).

The most costly step in the whole solution procedure is the maximization involved in (12); it is precisely this maximization that is no longer needed in the 2EGM procedure, as I will show below.

[4]This can be done using the quadrature techniques described for example in Judd (1998, chapter 7). See appendix A.1 for details on the quadrature technique used in this paper.

State-of-the-art solution techniques are all based on VFI, and differ mainly in two dimensions: whether they treat the state space as a discrete or a continuous set; and how they interpolate the value function outside of the grid that is used for approximation.

The most popular solution technique is the *discrete state space* (DSS) method, which consists in a complete discretization of the problem: the state space (a, y) and the choice space a' are discretized, the law of motion of GDP is approximated by a discrete Markov chain; as a consequence, the maximization problem (12) is easy to solve since it only involves a finite number of choices. In step 3, the price function $\hat{q}$ is computed using the following approximation of equation (9):

$$\hat{q}^{(n+1)}(y_j, a'_i) = \frac{1}{1+r} \sum_k \mathbb{1}_{\hat{V}^{G,(n)}(a'_i, y'_k) \geq \hat{V}^{B,(n)}(y'_k)} \mathbb{P}(y'_k | y_j)$$

where $\mathbb{P}(y'_k|y_j)$ is the transition probability in the discretized Markov chain. Note that DSS does not require any interpolation of the value functions outside the grid.

The DSS method is fast and easy to implement. But, as shown by Hatchondo et al. (2010), it is very imprecise unless a very fine grid (with thousands of points) is used.

The other solution technique based on VFI relies on *interpolation*, and treats the state space and the choice space as continuous. Between the points of the grid, the value functions are interpolated using well-known techniques, such as Chebychev polynomials or cubic splines. The maximization in (12) involves a costly nonlinear optimization. The price function is approximated by:

$$\hat{q}^{(n)}(y, a') = \frac{1 - F(y'^*|y)}{1+r} \tag{13}$$

where F is the conditional cumulative distribution function of y' given y, and y'^* is such that $\hat{V}^{G,(n)}(a', y'^*) = \hat{V}^{B,(n)}(y'^*)$ (note that a nonlinear solver is involved here).

The interpolation method is slower and more complicated to implement than DSS. But, as shown by Hatchondo et al. (2010), it is very precise, even when the grid only involves a few dozens of points.

One of the main lessons from the comparison exercise between DSS and the interpolation methods performed by Hatchondo et al. (2010) is that the interpolation methods have a much better speed/accuracy ratio. In order to achieve the same accuracy than interpolation, DSS needs to be given so many discretization points that it becomes painfully slow and basically pointless. DSS therefore seems of limited interest from a practical point of view, at least for the class of models under study.

4 The "doubly endogenous grid method" (2EGM)

In this section I present a new solution method for sovereign debt models. This technique is an adaptation of the *endogenous grid method* (EGM), initially introduced by Carroll (2006) and extended by Barillas and Fernández-Villaverde (2007).

The basic idea of this method is the following: it consists in using a fixed grid for the control variable (here a'), instead of a fixed grid for the state variable (here a) as in VFI. For a given value of the control variable a', the value of the state variable a for which a' is the optimal choice is derived using the first-order condition of the optimization problem. The grid over a becomes endogenous, hence the name of the method.

This method is much faster because it does not rely on a maximization at every point of the grid for every iteration; first order conditions are used instead of the maximization problem. The dramatic gain of speed obtained with the EGM is illustrated by Barillas and Fernández-Villaverde (2007): on a standard neoclassical growth model without labor, EGM is faster than

VFI by a factor of more than 11 (for a comparable accuracy level), and on the same model augmented with labor, EGM is faster by a factor of 56.

In the remainder of this section I describe how the EGM can be applied to sovereign debt models. I choose to call "doubly endogenous grid method" (2EGM) the resulting algorithm, for reasons that will become clear below. Extensions of this method to other sovereign debt models is straightforward, and such an extension to a larger model is studied in section 6.

The first step is to derive the Euler equation, *i.e.* the first order condition associated with the maximization problem (11):

$$u'(c_t)\left[q(y_t,a_{t+1})+a_{t+1}\frac{\partial q}{\partial a_{t+1}}(y_t,a_{t+1})\right]g_t=\beta g_t^{1-\gamma}\mathbb{E}_t\frac{\partial V}{\partial a_{t+1}}(a_{t+1},y_{t+1}) \tag{14}$$

where $c_t = y_t + a_t - q(y_t, a_{t+1})a_{t+1}g_t$.

As is customary for an Euler equation, this equation says that if assets are increased by one unit, the marginal loss of utility today must be equal the corresponding marginal gain in utility tomorrow. In particular, the term $q(y_t, a_{t+1}) + a_{t+1}\frac{\partial q}{\partial a_{t+1}}(y_t, a_{t+1})$ corresponds to the marginal decrease in today's consumption when a_{t+1} is increased by one unit: the first term is the direct crowding out effect of this marginal increase as evidenced by equation (4); the second term is the indirect effect of this marginal increase via the interest rate applied to the whole stock of assets.

I now turn to a description of the EGM applied to the canonical model of sovereign debt. Then I will explain why this does not work out of the box, and how to modify the algorithm in order to yield the functional 2EGM.

Let's first introduce the following function, which represents tomorrow's discounted expected utility:

$$\mathbb{V}(a',y)=\beta g^{1-\gamma}\int V(a',y')\mathrm{d}F(y'|y) \tag{15}$$

Note that the right hand side of the Euler equation (14) is equal to the derivative of $\mathbb{V}$ with respect to a', so that (14) can be rewritten in the following form:

$$c_t=u'^{-1}\left(\frac{\frac{\partial\mathbb{V}}{\partial a_{t+1}}(a_{t+1},y_{t+1})}{\left[q(y_t,a_{t+1})+a_{t+1}\frac{\partial q}{\partial a_{t+1}}(y_t,a_{t+1})\right]g_t}\right) \tag{16}$$

Then the EGM is as follows:

1. Define a finite grid of points for *tomorrow*'s assets $(a'_i)_{i\in I}$, and another one for today's output $(y_j)_{j\in J}$ (where I and J are finite indexing sets). These grids will remain constant during the iterations of the algorithm.

2. Set $n = 0$. Choose an initial interpolation grid for $\hat{V}^{G,(0)}$ and initial values for $\hat{V}^{G,(0)}$ and $\hat{V}^{B,(0)}$. The initial interpolation grid for $\hat{V}^{G,(0)}$ is $\left(a_{ij}^{(0)}, y_j\right)_{(i,j)\in I\times J}$ where $a_{ij}^{(0)} = a'_i$. That grid will vary over the iterations, hence the name of the *endogenous grid* method.

3. Construct an approximation $\hat{\mathbb{V}}^{(n)}$ of the function $\mathbb{V}$. This is done by applying formula (15) at every point of the fixed grid (a'_i, y_j), using a quadrature formula and the approximated function $\hat{V}^{(n)} = \max\{\hat{V}^{G,(n)}, \hat{V}^{B,(n)}\}$. Interpolation is used outside the fixed grid.

4. Compute $\hat{V}^{B,(n+1)}$ as you would in VFI:

$$\hat{V}^{B,(n+1)}(y_j)=u((1-\delta)y_j)+\beta g^{1-\gamma}\int\left[(1-\lambda)\hat{V}^{B,(n)}(y')+\lambda\hat{V}^{(n)}(0,y')\right]\mathrm{d}F(y'|y_j) \tag{17}$$

5. The step for computing $\hat{V}^{G,(n+1)}$ is as follows. For every point of the fixed grid (a'_i, y_j), use the rewritten Euler equation (16) to find the level of today's consumption consistent with this choice for *tomorrow's* assets:

$$c_{ij}^{(n+1)} = u'^{-1}\left(\frac{\frac{\partial \hat{\mathbb{V}}^{(n)}}{\partial a'}(a'_i, y_j)}{\left[\hat{q}^{(n)}(y_j, a'_i) + a'_i \frac{\partial \hat{q}^{(n)}}{\partial a'}(y_j, a'_i)\right] g_j}\right)$$

   This step involves a nonlinear solver (for the computation of $\hat{q}^{(n)}$, as in VFI) and two numerical differentiations (which are cheap), but *no maximization.*

   The level of today's assets consistent with this level of consumption is computed immediately using the resource constraint (4):

$$a_{ij}^{(n+1)} = \hat{q}^{(n)}(y_j, a'_i) a'_i g_j - y_j + c_{ij}^{(n+1)}$$

   The function $\hat{V}^{G,(n+1)}$ will be interpolated over the grid $\left(a_{ij}^{(n+1)}, y_j\right)$; this grid is determined endogenously during the resolution of the model. The value of the function at these points is then:

$$\hat{V}^{G,(n+1)}\left(a_{ij}^{(n+1)}, y_j\right) = u\left(c_{ij}^{(n+1)}\right) + \hat{\mathbb{V}}^{(n)}(a'_i, y_j) \tag{18}$$

6. Let $\hat{V}^{(n+1)} = \max\{\hat{V}^{G,(n+1)}, \hat{V}^{B,(n+1)}\}$. If $\hat{V}^{(n+1)}$ is close enough to $\hat{V}^{(n)}$ (up to some target accuracy level), then stop. Otherwise, set $n = n + 1$ and go to step 3.

This method is very appealing and intuitive despite its apparent complexity. It shares the core of the VFI method: choose an approximation grid and iterate backwards in time as in a finite-horizon model. The crucial difference is how the optimal decision for the level of debt is computed: in VFI, the algorithm deduces the choice for tomorrow's debt given today's debt; in EGM, it is the reverse: the algorithm takes as given a level for tomorrow's debt, and deduces the level of today's debt which is consistent with this choice. The analytical tools are therefore different between the two methods: VFI uses the maximization given by the Bellman equation (11) while EGM uses the first-order condition (14). And the latter happens to be much less computationally expensive to solve than the former, hence the dramatic gain in performance of EGM over VFI.

There is however a characteristic of the canonical sovereign debt model that makes the EGM fail if applied blindly. The problem comes from the fact that the choice function $a'(a, y)$ for tomorrow's level of assets is very "flat," *i.e.* it takes a narrow range of values. I illustrate this in Figure 1: over the range of values for which the model is solved ($a \in [-0.3, 0]$), the choice function a' takes its values in the interval $[-0.14, -0.20]$, which is much narrower.

As a consequence, if the full range $[-0.3, 0]$ is used for the fixed grid for a' (in step 1 of the EGM procedure), then the values deduced for a (in step 5) will be either very large or, worse, invalid (*e.g.* corresponding to a negative consumption). The algorithm will therefore fail.

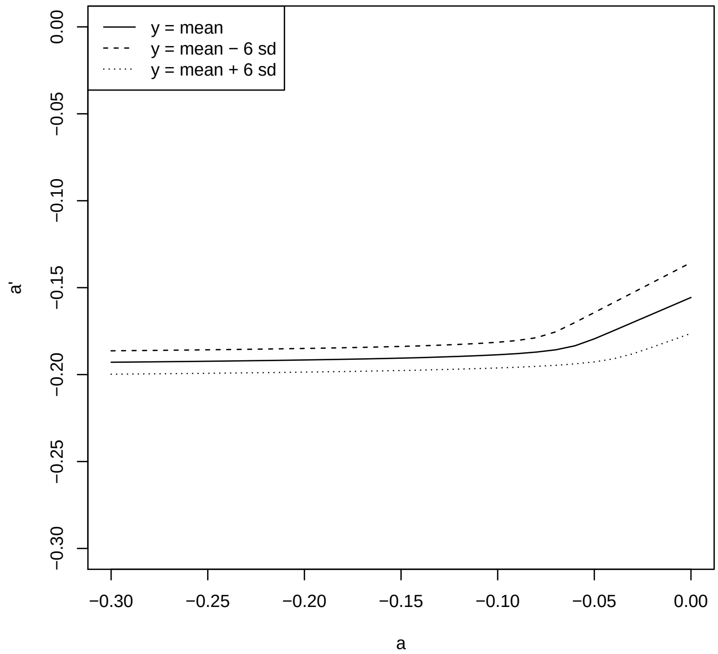
A solution would be to use the interval $[-0.14, -0.20]$ for the fixed grid over a', but this is not a practical solution since this range is precisely an output of the computation and cannot be guessed *ex ante* in the general case.

The solution that I suggest is to adapt the algorithm so that the grid over a' becomes also endogenous and converges towards the ergodic set as iterations run.[5] Hence the suggested name of "doubly endogenous grid method" (2EGM).

Here is the modified algorithm:

[5]The *ergodic set* is the set of points of the state space that are reached in equilibrium. This is a probabilistic

Figure 1: Choice function for tomorrow's level of debt, given today's level



The choice function is plot for 3 values of y which cover most of the ergodic distribution. The scales on horizontal and vertical axes are identical. *sd* stands for standard deviation.

1. Define a finite grid of points for output $(y_j)_{j \in J}$ (J is a finite indexing set). This grid will remain constant during the iterations of the algorithm. Also define a finite indexing set I for asset values.

2. Set $n = 0$. Define a maximum asset value $\bar{a} = 0$ and a minimum asset value $\underline{a}^{(0)}$. These bounds are meant to capture the ergodic set. The minimum asset value will be updated during iterations.

3. Choose initial interpolation grids and initial values for $\hat{V}^{G,(0)}$ and $\hat{V}^{B,(0)}$. The interpolation grid for $\hat{V}^G$ will vary over the iterations. The initial grid is $\left(a_{ij}^{(0)}, y_j\right)_{(i,j) \in I \times J}$ such that for a given j, the points $a_{ij}^{(0)}$ are evenly distributed in the range $[\underline{a}^{(0)}, \bar{a}]$. The initial values for the value functions are given in section A.3.

4. Construct an approximation $\hat{\mathbb{V}}^{(n)}$ of the function $\mathbb{V}$. First, generate a grid (a'_i, y_j) where the a'_i are evenly distributed in the range $[\underline{a}^{(n)}, \bar{a}]$. Then apply formula (15) at every point of the grid , using a quadrature formula and the approximated function $\hat{V}^{(n)} = \max\{\hat{V}^{G,(n)}, \hat{V}^{B,(n)}\}$. Interpolation is used outside the grid.

5. Compute $\hat{V}^{B,(n+1)}$ as in (17).

6. The step for computing $\hat{V}^{G,(n+1)}$ is as follows.

concept because potentially any point in the state space can be reached after a big enough shock. One should rather talk about the ergodic set *at a given probability level*—*e.g.* 99%: it is the subset of the state space where the model evolves at least 99% of the time in equilibrium.

- Consider the function f that maps tomorrow's assets to today's assets (using the Euler equation (14) and the resource constraint (4)):

$$f^{(n)}(a', y) = \hat{q}^{(n)}(y, a')a'g - y + u'^{-1}\left(\frac{\frac{\partial \hat{\mathbb{V}}^{(n)}}{\partial a'}(a', y)}{\left[\hat{q}^{(n)}(y, a') + a'\frac{\partial \hat{q}^{(n)}}{\partial a'}(y, a')\right]g}\right) \tag{19}$$

  Then for every $j \in J$, use a dichotomy-based algorithm to compute:

$$\bar{a}_j'^{(n)} = \max\left\{a' \mid f^{(n)}(a', y_j) \in [\underline{a}^{(n)}, \bar{a}]\right\}$$
$$\underline{a}_j'^{(n)} = \min\left\{a' \mid f^{(n)}(a', y_j) \in [\underline{a}^{(n)}, \bar{a}]\right\}$$

  The grid for tomorrow's assets $a_{ij}'^{(n)}$ is computed by drawing points in the range $\left[\underline{a}_j'^{(n)}, \bar{a}_j'^{(n)}\right]$ (see section A.3 for a discussion on the distribution of these points within the range).
  The idea here is to limit the grid for tomorrow's assets to points that are compatible with a level of today's asset lying in the range of interest.
- The interpolation grid for $\hat{V}^{G,(n+1)}$ is $\left(a_{ij}^{(n+1)}, y_j\right)$, defined by $a_{ij}^{(n+1)} = f^{(n)}\left(a_{ij}'^{(n)}, y_j\right)$
- Compute the value of $\hat{V}^{G,(n+1)}$ at interpolation points as in (18)

7. Update the minimum asset value:

$$\underline{a}^{(n+1)} = \min_{j \in J} \underline{a}_j'^{(n)} - \Delta$$

   where Δ is an offset (typically 1%).

   The idea here is to limit the asset range to levels that are effectively chosen by the sovereign in equilibrium.

8. Let $\hat{V}^{(n+1)} = \max\{\hat{V}^{G,(n+1)}, \hat{V}^{B,(n+1)}\}$. If $\hat{V}^{(n+1)}$ is close enough to $\hat{V}^{(n)}$ (up to some target accuracy level), then stop. Otherwise, set $n = n + 1$ and go to step 4.

The 2EGM is very flexible and robust because all the grids that it uses are endogenous and updated at every iteration. As a by-product, these grids will converge towards the ergodic set. This increases the efficiency of the algorithm: approximations of the decision and value functions are only computed on areas of interest in the state space. No resource is lost in computing functions in areas of the state space that are never reached in equilibrium.

The 2EGM is more computationally expensive than the original EGM, because there is a need to compute minimum and maximum bounds for tomorrow's assets using a dichotomy-based algorithm. But this extra cost does not prevent 2EGM from being far more efficient than VFI, as I show in the next section.

5 Assessing solution methods with the canonical sovereign debt model

In this section I apply both VFI and 2EGM to the canonical model of section 2, and I assess their respective performance in the terms of speed, accuracy and complexity of implementation.

The two implementations have been written using the C++ programming language, and have been tested on a 8-cores computing workstation. The implementations have been parallelized in order to exploit as much as possible of the power of all the CPU cores. Then the programs have been run both in single-threaded mode and in multi-threaded mode, in order to quantify the benefits of parallelization for both algorithms. More details on the implementations can be found in appendix A.

In order to compare the accuracy of the two methods, I use two devices. The first one is simply the moments of the model: these are the most interesting objects from the point of view of the economist, and they were used in the comparison study of Hatchondo et al. (2010). But using only the moments to compare the accuracy of solution methods is hardly satisfactory. First, it is in theory possible that a highly inaccurate solution yields the right moments under consideration, while being wrong in other dimensions. Second, and more fundamentally, the moments do not give an absolute measure of accuracy, since one does not know the true moments in the absence of a closed-form solution: the moments give an idea of the distance between two given solutions, but not their distance to the true solution.

As a second comparison device, I therefore use accuracy checks based on Euler equation errors. This type of accuracy checks have been first introduced in Judd (1992) and have since become the standard way of assessing the accuracy of solutions to rational expectation models (Jin and Judd, 2002; Barillas and Fernández-Villaverde, 2007; Juillard and Villemot, 2011). The idea is quantify how close the solution is to satisfying the first-order conditions of the model. On a basic RBC model, this amounts to verifying that the model satisfies the Euler equation, hence the name of the method.

The Euler equation of the canonical model can be expressed as follows (after substituting out the value function from (14)):

$$u'(c_t) = \frac{\beta\, g_t^{-\gamma} \int_{D_{t+1}=0} u'(c_{t+1}) \mathrm{d}F(y_{t+1}|y_t)}{q(y_t, a_{t+1}) + a_{t+1} \frac{\partial q}{\partial a_{t+1}}(y_t, a_{t+1})} \tag{20}$$

where D_{t+1} is a dummy equal to 1 if the country defaults tomorrow, 0 otherwise.

Note that the expectation of tomorrow's utility is only computed over the states of nature for which there is no default tomorrow. The technical reason is simply that $\frac{\partial V^B}{\partial a} = 0$. This is the Panglossian effect that is discussed in Cohen and Villemot (2011): from an *ex ante* perspective, the country rationally ignores the future states of nature in which he will not repay.

The unit-free Euler error of a solution is then defined as the relative difference between the left and right hand-sides of (20). The policy functions of the solution are used for computing today's and tomorrow's control variables. The Euler error can therefore be expressed as:

$$\mathcal{R}(a,y) = 1 - \frac{\beta\, g^{-\gamma} \int_{\hat{D}(\hat{a}'(a,y),y')=0} u'[\hat{c}(\hat{a}'(a,y), y', 0)] \mathrm{d}F(y'|y)}{\left[\hat{q}(y, \hat{a}'(a,y)) + \hat{a}'(a,y) \frac{\partial \hat{q}}{\partial a_{t+1}}(y, \hat{a}'(a,y))\right] u'[\hat{c}(a,y,0)]} \tag{21}$$

where $\hat{a}'(a,y)$ denotes the policy function for tomorrow's level of asset (conditionally to repayment), $\hat{D}(a,y)$ the decision function for default, $\hat{c}(a,y)$ the consumption function in case of repayment and $\hat{q}(y,a')$ the price function.

The main property of the Euler error is that it is equal to zero for the true solution of the model. As a consequence, the Euler error can be used as a loose metrics for measuring the distance of an approximated solution to the true solution.[6]

[6]Note however that the Euler error cannot be used to construct a distance (in the topological sense) to the true solution. Such a construction is not possible in the general case.

The Euler error defined in (21) is constructed for a given point in the state space, *i.e.* for a given pair (a, y), and conditionally to repayment. In order to create an overall measure of the "distance" of an approximated solution to the true solution, one needs to compute Euler errors at several representative points of the state space, and then report an aggregate measure. For this purpose, I simulate a series of 10,000 points in the state space, starting from $a = 0$ and $y = 0$, and at each period I recursively apply the policy functions. Then at each point of the simulation for which the country decides to repay, I compute the Euler error, and I report the mean and maximum of the absolute values of these errors across the simulation; this gives an idea of both the average and worst case performance of the solution. Note that no error is computed for the periods at which the country is in financial autarky (since there is no Euler equation in that case), but this is not an issue since no real computational challenge is involved there: no optimization is performed, the country simply behaves hand-to-mouth.

Table 2 shows the results of the comparison of VFI and 2EGM over the canonical model. Column (1) corresponds to VFI and column (2) to 2EGM using the same number of grid points than for VFI. Column (3) reports calculations done by Hatchondo et al. (2010) on the same model: their results should be comparable to column (1) since they use the same method and the same grid.

I first discuss the relative accuracy of the two algorithms. When used on the same grid and with the same convergence criterion, VFI and 2EGM are of comparable accuracy, as can be seen from both the Euler errors and the simulated moments. The Euler errors are of comparable magnitude (2EGM being slightly less precise, but by a tiny margin). The simulated moments are the same across the two methods up to the second decimal.

Concerning speed, I report two computing times: the first one when only a single thread (*i.e.* a single processor) is used, the second when 8 threads (or processor cores) are used simultaneously. Comparing single- and multi-threaded computing times gives an idea of how parallelizable both algorithms are: this information is critical since it is to be expected that future technology improvements in computers will be in terms of number of cores rather than in speed of individual cores (as was the rule in the past).

The results show that 2EGM is much faster than VFI. With a single thread, 2EGM is almost 10 times faster. With 8 threads, 2EGM is still much faster, but the ratio is reduced to 5. This suggests that VFI benefits more from parallelization than 2EGM, at least in the way I implemented both algorithms.

It should also be noted that the computing time that I report for VFI is 21 times smaller than what Hatchondo et al. (2010) achieved. This is to some extent the consequence of a better hardware, but it is probably also the consequence of a better implementation. Given that I have made a very efficient implementation of VFI, my results showing the superiority of 2EGM can therefore not be considered as biased against VFI because of a poor implementation of the latter. If there is any such bias, it more likely plays in the other direction, since in implementing 2EGM I could not take advantage of the experience gained from pre-existing implementations (which were nonexistent).

The last thing to note is that VFI and 2EGM have more or less the same degree of complexity when it comes to programming the algorithms: the two algorithms are implemented using about one thousand of single lines of code.[7]

[7] As reported by the SLOCCount program by David A. Wheeler, see `http://www.dwheeler.com/sloccount/sloccount.html`.

Table 2: Comparison of VFI and 2EGM on canonical model

	(1)	(2)	(3)
Solution characteristics			
Method	VFI	2EGM	VFI
Grid points for y	15	15	15
Grid points for a	30	30	30
Convergence criterion (in $\log_{10}$ units)	−6	−6	
Lines of C++ code	1,000	1,080	
Solution time			
Single thread	54.4s	5.8s	1182s
8 threads	15.9s	3.1s	
Moments			
Rate of default (%, per year)	0.86	0.86	0.88
Mean debt output ratio (%, annualized)	4.68	4.68	4.75
$\sigma(y)$ (%)	4.40	4.40	4.43
$\sigma(c)$ (%)	4.64	4.64	4.68
$\sigma(TB/y)$ (%)	0.92	0.92	0.94
$\sigma(\Delta)$ (%)	0.06	0.06	0.07
$\rho(c, y)$	0.98	0.98	0.98
$\rho(TB/y, y)$	−0.18	−0.18	−0.18
$\rho(\Delta, y)$	0.05	0.05	0.09
$\rho(\Delta, TB/y)$	0.53	0.53	0.52
Euler errors (in $\log_{10}$ *units)*			
Mean	−4.38	−4.20	
Max	−3.47	−3.39	

The model and calibration are those of section 2. Columns (1) and (2) report calculations by the author, while column (3) report calculations by Hatchondo et al. (2010). The convergence criterion is the maximum difference tolerated between value functions of two consecutive iterations when convergence is achieved. Moments are obtained by averaging over 500 simulated series of 1,500 points each, of which the first 1,000 are discarded. y is GDP, c is consumption, TB/y is trade balance over GDP, Δ is spread. GDP, consumption, trade balance and spread are detrended with an HP filter of parameter 1600. Euler errors are computed according to (21), and I report the mean and maximum errors over a path of 10,000 points simulated using the computed solution.

6 Assessing solution methods with the "trembling times" model

In this section I apply 2EGM to a larger model of sovereign default, specifically the "trembling times" model presented in Cohen and Villemot (2012, section 4), and then I perform a comparison exercise between 2EGM with VFI. For the sake of brevity, I do not reproduce here the details of the model; the interested reader should refer to the aforementioned paper. The important point to have in mind is that this model is more complex to solve than the previous one, because it has a state space of dimension three (instead of two): in addition to the level of assets a and to the Brownian component of the growth process y, this model adds a Poisson component z to the growth process.[8] Moreover, that model has two stochastic shocks (instead of one in the canonical model).

Similarly to the previous section, I report in Table 3 a comparison across several dimensions of VFI and 2EGM on the "trembling times" model.

In comparison to the solution of the canonical model presented in the previous section, I choose a coarser grid along the dimensions for a and y (only 10×10 points here, while I used 30×15 in the previous model). Since I use also 10 points for z, the total number of points is 1,000 here (against 450 in the previous section).

Also, I use a relatively loose convergence criterion for both VFI and 2EGM, but for different reasons. In the VFI case, the algorithm does not stop when the criterion is set to a tighter value (*i.e.* in that case the halting condition in step 4 of section 3 is never met). In the 2EGM case, the algorithm converges for lower values of the criterion (*e.g.* 10^{-5}), but the extra iterations do not improve the quality of the solution (at least from the angle of the Euler error), and are therefore a loss of computing time. The conclusion that can be drawn from these considerations is that, for the chosen grid, the results shown here correspond to the best outcome that both algorithms can deliver.

The resulting solutions appear to be much less accurate than those obtained for the canonical model in the previous section. Where the average relative Euler error was about 10^{-4}, it is now 100 times bigger, around 1%. Also note that the average error of 2EGM is smaller than that of VFI, by almost 20%. Concerning the maximum error, both algorithms perform very poorly: over a 10,000 periods simulation, the maximum error is about 10% for VFI and 34% for 2EGM. While these figures are important, they concern only a very small number of simulation points: for example, along the simulation path for 2EGM, only 0.05% of the points exhibit an Euler error greater than 10% (which is the maximum error for VFI). These points can therefore be considered as outliers and, given that on average 2EGM performs better than VFI by a significant margin, one can reasonably say that the two solutions shown here are of comparable accuracy.

In terms of simulated moments, the two solution techniques deliver results which are qualitatively similar, and quantitatively close albeit slightly different. The largest discrepancy comes from the probability of default: VFI gives a probability of 1.23%, when 2EGM gives a result of 2.54% which is about two times bigger. This discrepancy likely comes from the fact the probability of default is highly sensitive to the probability q of exiting the "trembling times" around the value that was chosen for it (5%), as shown in Figure 2 of Cohen and Villemot (2012). As a sensitivity analysis exercise, I have tried both a lower and a higher value for q, and in both cases the two solution methods give similar results (*i.e.* almost no default if q is large or, at the other extreme, a default frequency close to the Poisson parameter p if q is small).

Since the hardware and the programming techniques are the same between the implementations of both the canonical and the "trembling times" models, the computing times are directly

[8] There is also a fourth state variable tracking whether we are in normal or trembling times, but since it can take only two values it does not increase the dimensionality of the problem.

Table 3: Comparison of VFI and 2EGM on “trembling times” model

	(1)	(2)
Solution characteristics		
Method	VFI	2EGM
Grid points for y	10	10
Grid points for z	10	10
Grid points for a	10	10
Convergence criterion (in $\log_{10}$ units)	-1.7	-3.0
Lines of C++ code	1,423	1,525
Solution time		
Single thread	3,588s	413s
8 threads	1,396s	195s
Moments		
Rate of default (%, per year)	1.24	2.50
Mean debt output ratio (%, annualized)	38.58	38.17
$\sigma(y)$ (%)	4.45	4.45
$\sigma(c)$ (%)	6.47	6.04
$\sigma(TB/y)$ (%)	3.11	2.63
$\sigma(\Delta)$ (%)	0.40	0.57
$\rho(c, y)$	0.90	0.92
$\rho(TB/y, y)$	-0.44	-0.41
$\rho(\Delta, y)$	-0.47	-0.60
$\rho(\Delta, TB/y)$	0.79	0.64
Euler errors (in $\log_{10}$ units)		
Mean	-1.99	-2.08
Max	-0.98	-0.46

The model and calibration are those of section Cohen and Villemot (2012, section 4). The convergence criterion is the maximum difference tolerated between value functions of two consecutive iterations when convergence is achieved. Moments are obtained by averaging over 500 simulated series of 1,500 points each, of which the first 1,000 are discarded. y is GDP, c is consumption, TB/y is trade balance over GDP, Δ is spread. GDP, consumption, trade balance and spread are detrended with an HP filter of parameter 1600. Euler errors are computed over a path of 10,000 points simulated using the computed solution.

comparable. The ratio is between 62:1 and 88:1 depending on the algorithm and the number of threads chosen. This is a striking illustration of the curse of dimensionality: the choice of a coarser grid and of a looser convergence criterion does not prevent the computation time from exploding when only a single dimension is added, and in addition the result is a much less precise solution.

In terms of computing time, the comparison between VFI and 2EGM delivers a similar picture than the one obtained in the previous section: 2EGM is faster than VFI by a factor of 8.5 when there is a single thread, and by a factor of almost 7 when there are 8 threads.

Again, the programming complexity is roughly the same for both algorithms: about 1,500 lines of code. Note also that this is a 50% increase compared to the canonical model.

7 Conclusion

Building on earlier work by Carroll (2006) and Barillas and Fernández-Villaverde (2007) on the endogenous grid method, I have presented in this paper a new solution method for sovereign debt models à la Eaton and Gersovitz (1981). This technique is easy to implement and significantly improves the speed-accuracy frontier compared to pre-existing techniques based on value function iteration (VFI). For a similar accuracy, this *doubly endogenous grid method* (2EGM) is faster than VFI by a factor between 5 and 10. These properties have been verified on a simple sovereign debt model such as the one presented in section 2, and on the more complex model with 3 state variables and 2 stochastic shocks presented in Cohen and Villemot (2012).

Having a fast and accurate algorithm such as the 2EGM opens several interesting possibilities. One is to make easier the study of bigger sovereign debt models than the ones currently found in the literature; I already did so in this paper by solving the "trembling times" model which has more state variables and stochastic shocks than any model of the related literature to date. Future models at the juncture between the RBC/DSGE and the endogenous default traditions, in the spirit of Mendoza and Yue (2012), will likely feature a state space of even higher dimension and could also benefit from the 2EGM technique. Another possibility worth exploring is the estimation of sovereign debt models with Bayesian techniques: since such an estimation necessitates to solve the model a great number of times at different points in the parameter space, the 2EGM could be helpful in this endeavor.

References

Abramowitz, M. and I. A. Stegun (1964): *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York: Dover, ninth Dover printing, tenth GPO printing ed.

Aguiar, M. and G. Gopinath (2004): "Defaultable Debt, Interest Rates and the Current Account," NBER Working Papers 10731, National Bureau of Economic Research, Inc.

——— (2006): "Defaultable debt, interest rates and the current account," *Journal of International Economics*, 69, 64–83.

Aruoba, S. B., J. Fernández-Villaverde, and J. F. Rubio-Ramírez (2006): "Comparing solution methods for dynamic equilibrium economies," *Journal of Economic Dynamics and Control*, 30, 2477–2508.

Barillas, F. and J. Fernández-Villaverde (2007): "A generalization of the endogenous grid method," *Journal of Economic Dynamics and Control*, 31, 2698–2712.

Carroll, C. D. (2006): "The method of endogenous gridpoints for solving dynamic stochastic optimization problems," *Economics Letters*, 91, 312–320.

Cohen, D. and J. Sachs (1986): "Growth and external debt under risk of debt repudiation," *European Economic Review*, 30, 529–560.

Cohen, D. and S. Villemot (2011): "Endogenous debt crises," CEPR Discussion Papers 8270, Centre for Economic Policy Research.

——— (2012): "The sovereign default puzzle: Modelling issues and lessons for Europe," PSE Working Papers halshs-00692038, Paris School of Economics.

Eaton, J. and M. Gersovitz (1981): "Debt with Potential Repudiation: Theoretical and Empirical Analysis," *Review of Economic Studies*, 48, 289–309.

Galassi, M., J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi (2003): *GNU Scientific Library: Reference Manual*, Network Theory Ltd.

Hatchondo, J. C., L. Martinez, and H. Sapriza (2010): "Quantitative properties of sovereign default models: solution methods," *Review of Economic Dynamics*, 13, 919–933.

Jin, H.-H. and K. L. Judd (2002): "Perturbation methods for general dynamic stochastic models," Unpublished manuscript, Stanford University.

Judd, K. L. (1992): "Projection methods for solving aggregate growth models," *Journal of Economic Theory*, 58, 410–452.

——— (1998): *Numerical Methods in Economics*, MIT Press Books, The MIT Press.

Juillard, M. and S. Villemot (2011): "Multi-country real business cycle models: Accuracy tests and test bench," *Journal of Economic Dynamics and Control*, 35, 178–185.

Kollmann, R., S. Maliar, B. A. Malin, and P. Pichler (2011): "Comparison of solutions to the multi-country Real Business Cycle model," *Journal of Economic Dynamics and Control*, 35, 186–202.

Mendoza, E. G. and V. Z. Yue (2012): "A General Equilibrium Model of Sovereign Default and Business Cycles," *The Quarterly Journal of Economics*, 127, 889–946.

Sachs, J. and D. Cohen (1982): "LDC Borrowing with Default Risk," NBER Working Papers 0925, National Bureau of Economic Research, Inc.

A Appendix: Implementation details

In this section, I document several details of the implementation of the two algorithms compared in this paper.

A.1 Implementation details common to VFI and 2EGM

The programs used in this paper are written in the C++ language and compiled using the GNU C++ compiler.[9] Numerical computations are done using double-precision floating points numbers (IEEE 754), and the implementation makes a heavy use of various routines provided by the GNU Scientific Library.[10] Parallelization of algorithms is achieved using OpenMP directives.[11]

Value functions and decision rules are interpolated outside the grid using cubic splines (interpolation is done both *between* grid points and *outside* the convex envelope of the grid). Interpolations occurs sequentially along each dimension in the following order: first along the asset dimension a (except for value functions in case of default), then along the Brownian component of growth y, and finally along the Poisson component of growth z (only for the "trembling times" model). The interpolation engine is designed to be able to cope with values of $-\infty$ in some parts of the state space (in that case, interpolation is done ignoring the infinite values): this feature is particularly helpful when exploring points of the state space for which, in case of repayment, the markets provide no level of lending compatible with a positive consumption (*i.e.* points for which $V^G = -\infty$).

Numerical integration over the Gaussian distribution is achieved using a Gauss-Legendre quadrature (the distribution is truncated below and above 4 standard deviations) using 16 points for the canonical model and 10 points for the "trembling times" model (see Abramowitz and Stegun, 1964, p. 887, eq. 25.4.29).

The price function given in (13) is computed using "Brent's method," a well-known nonlinear solver combining an interpolation strategy and the bisection algorithm.

The programs have been run on a workstation with the following characteristics: two Intel Xeon X5460 quad-core processors clocked at 3.16Ghz with 256kb of L1 cache and 12Mb of L2 cache, 8Gb of DDR2 RAM clocked at 667Mhz. The workstation is running Debian GNU/Linux.

A.2 Implementation details specific to VFI

The maximization of equation (12) is done in two steps:

- first, a global search is performed by computing the objective over a pre-defined grid of points over tomorrow's assets a';
- the result of the global search is used as a starting point for the "Brent minimization algorithm," which combines a parabolic interpolation with the golden section algorithm.

Concerning the initial values, a natural candidate is the continuation value at the last period of the finite horizon version of the model:

$$\hat{V}^{G,(0)}(a_i, y_j) = u(y_j + a_i) \tag{22}$$

$$\hat{V}^{B,(0)}(y_j) = u((1-\delta)y_j)$$

However, if the domain over which the problem is solved includes points for which $y + a < 0$, then this particular initial value for $\hat{V}^G$ cannot be used (because u is only defined for positive values). In a quarterly model, this problem appears as soon as the (annualized) debt-to-GDP ratio is of 25%: since many countries (both emerging and developed) have bigger ratios, this is certainly a problem which one wants to circumvent.

[9] See http://www.gnu.org/software/gcc.

[10] See http://www.gnu.org/software/gsl/ or Galassi et al. (2003).

[11] See http://www.openmp.org.

The solution that I adopt is to use the following alternative initial value:

$$\hat{V}^{G,(0)}(a_i, y_j) = u(y_j + r\, a_i) \tag{23}$$

Since r is typically about 1%, this functional form is compatible with realistic debt-to-GDP ratios. It also has an economic interpretation: it corresponds to the utility that the country gets if it keeps its indebtment at a constant level.

A.3 Implementation details specific to 2EGM

A.3.1 Initial values

The initial values for $\hat{V}^G$ and $\hat{V}^B$ that I presented in the previous section for VFI do not work for 2EGM. The fundamental problem with (23) is that its derivative with respect to a is far from the derivative of V^G (which is close to one), and the 2EGM relies on this derivative in order to converge. The form in (22) has a better derivative, but is still not applicable for the same reasons than those exposed in the previous section. I therefore use the following form which solves both problems:

$$\hat{V}^{G,(0)}\left(a_{ij}^{(0)}, y_j\right) = \alpha \cdot u\left(y_j + \frac{a_{ij}^{(0)}}{\alpha}\right)$$

where α is a constant set to 3 for the canonical model and 10 for the "trembling times" model.

Also the algorithm only converges if the initial values are such that they imply no default ever, so I choose a value for $\hat{V}^{B,(0)}$ which is always smaller than $\hat{V}^{G,(0)}$:

$$\hat{V}^{B,(0)}(y_j) = \alpha \cdot u\left(y_j + \frac{\underline{a}^{(0)}}{\alpha}\right)$$

A.3.2 Generation of the endogenous grid

In step 6 (p. 9) of 2EGM, the grid for tomorrow's assets level is endogenously generated (this is precisely the step that I added relatively to the original EGM).

The first step is to compute the boundaries $\bar{a}_j'^{(n)}$ and $\underline{a}_j'^{(n)}$ of the range of interest. This is done with a dichotomy-based algorithm: for computing the lower bound $\underline{a}_j'^{(n)}$, I start with the hypothesis that this bound is located in a wide range, then at each iteration I cut the interval in two by keeping the right half (guessing which half to keep is done by applying the function (19) to the point at the middle of the interval); the same algorithm is used to find the upper bound $\bar{a}_j'^{(n)}$.

Once the bounds have been computed, the second step is to draw the points $a_{ij}'^{(n)}$ in this interval. The natural way of doing it would be to draw evenly distributed points in the interval, but this is highly inefficient and leads to poor accuracy results, because the ergodic set is located in the neighborhood of the lower bound $\underline{a}_j'^{(n)}$ (as shown by Figure 1). The optimal solution is to put more points where the country spends the more time, using a cubic formula:

$$a_{ij}'^{(n)} = \underline{a}_j'^{(n)} + \left(\frac{i}{|I|-1}\right)^3 \left(\bar{a}_j'^{(n)} - \underline{a}_j'^{(n)}\right)$$

(assuming that the indexing set I consists of integers from 0 to $|I|-1$).

A.3.3 Refinement iterations

Since iterations in 2EGM are based on the Euler equation (14), one would expect 2EGM to deliver very small Euler errors *by construction*. But things are not that simple, because Euler errors are computed using the expectancy of tomorrow's marginal utility (see (20)), while the formula used in 2EGM iterations is based on the derivative of the value function with respect to a (see (16)). In the true solution, these two are equal. But in the solution delivered by 2EGM, the discrepancy between the marginal utility of consumption and the derivative of the value function is not negligible and can lead to substantial Euler errors.

Hopefully there is an easy way to improve that situation. After the convergence of 2EGM, I run a few extra iterations where I use a modified version of equation (19): I naturally replace the derivative of the value function by the marginal utility of consumption (based on the policy function computed at the previous iteration). On the canonical model, this has a very small computational cost (only one extra iteration is necessary), but it dramatically improves the accuracy of the solution. Without this extra refinement iteration, 2EGM would have performed much worse than VFI. Unfortunately, I was not able to run this refinement step on the "trembling times" model: instead of improving the convergence, it leads to a divergence. A better understanding of this issue could be the subject of further research.