NBER WORKING PAPER SERIES

MERGING SIMULATION AND PROJECTION APPROACHES TO SOLVE HIGH-DIMENSIONAL PROBLEMS

Kenneth L. Judd
Lilia Maliar
Serguei Maliar

Working Paper 18501
http://www.nber.org/papers/w18501

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
November 2012

This is a substantially revised version of the NBER working paper 15965 entitled "A Cluster-Grid Projection Method: Solving Problems with High Dimensionality. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

Merging Simulation and Projection Approaches to Solve High-Dimensional Problems
Kenneth L. Judd, Lilia Maliar, and Serguei Maliar
NBER Working Paper No. 18501
November 2012
JEL No. C61,C63

ABSTRACT

We introduce an algorithm for solving dynamic economic models that merges stochastic simulation and projection approaches: we use simulation to approximate the ergodic measure of the solution, we construct a fixed grid covering the support of the constructed ergodic measure, and we use projection techniques to accurately solve the model on that grid. The grid construction is the key novel piece of our analysis: we select an -distinguishable subset of simulated points that covers the support of the ergodic measure roughly uniformly. The proposed algorithm is tractable in problems with high dimensionality (hundreds of state variables) on a desktop computer. As an illustration, we solve one- and multicountry neoclassical growth models and a large-scale new Keynesian model with a zero lower bound on nominal interest rates.

Kenneth L. Judd
Hoover Institution
Stanford University
Stanford, CA 94305-6010
and NBER
kennethjudd@mac.com

Serguei Maliar
Office T-24 Hoover Institution
Stanford University
CA 94305-6010, USA
maliars@stanford.edu

Lilia Maliar
Office T-24 Hoover Institution
Stanford University
CA 94305-6010, USA
maliarl@stanford.edu

1 Introduction

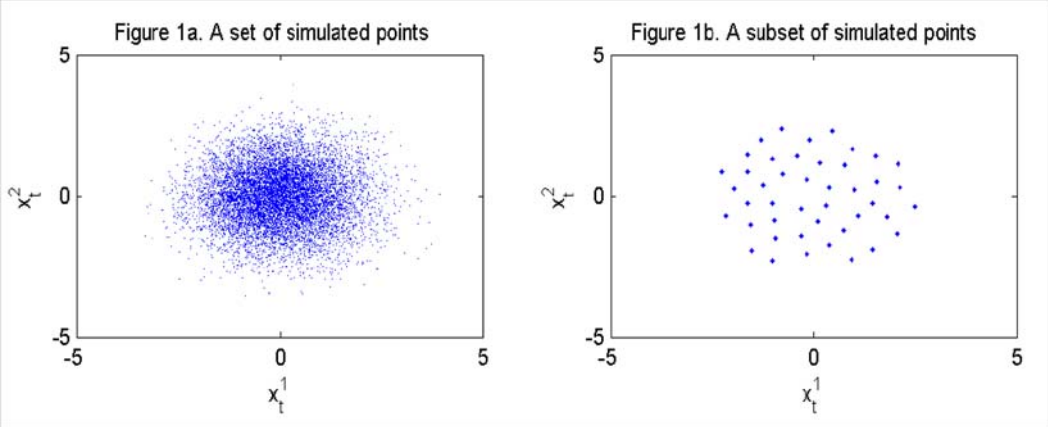
We introduce an algorithm for solving dynamic economic models that merges stochastic simulation and projection approaches: we use simulation to approximate the ergodic measure of the solution, we construct a fixed grid covering the support of the constructed ergodic measure, and we use projection techniques to accurately solve the model on that grid. The grid construction is the key novel piece of our analysis: we select an ε-distinguishable subset of simulated points that covers the support of the ergodic measure roughly uniformly. The proposed algorithm is tractable in problems with high dimensionality (hundreds of state variables) on a desktop computer. As an illustration, we solve one- and multicountry neoclassical growth models and a large-scale new Keynesian model with a zero lower bound on nominal interest rates.

One stream of literature for solving dynamic economic models relies on stochastic simulation. This stream includes the methods for solving rational expectations models, e.g., Fair and Taylor (1983), Marcet (1988), Smith (1993), Maliar and Maliar (2005), Judd, Maliar and Maliar (2011), as well as the literature on learning, e.g., Marcet and Sargent (1989), Bertsekas and Tsitsiklis (1996), Pakes and McGuire (2001), Powell (2011). The key advantage of stochastic simulation methods is that the geometry of the set on which the solution is computed is adaptive. Namely, such methods solve economic models on a set of points realized in equilibrium which makes it possible to avoid the cost of finding the solution in areas of the state space that are effectively never visited in equilibrium. However, a set of simulated points itself is not an efficient choice either as a grid for approximating a solution (it contains many closely-located and hence, redundant points) or as a set of nodes for approximating expectation functions (the accuracy of the Monte Carlo integration method is low).

Another stream of literature for solving dynamic economic models relies on projection techniques; see, e.g., Wright and Williams (1984), Judd (1992), Christiano and Fisher (2000) and Kruger and Kubler (2004). Projection methods use efficient discretizations of the state space and effective deterministic integration methods, and they deliver very accurate solutions. However, conventional projection methods are limited to fixed geometries such as a multidimensional hypercube. In order to capture all points that are visited in equilibrium, a hypercube must typically include large areas of the state space that have a low probability to happen in equilibrium. The size of the hypercube domain, on which projection methods operate, grows rapidly with the dimensionality of the problem.

We propose a solution method that combines the best features of stochastic simulation and projection methods, namely, it combines an adaptive geometry of stochastic simulation methods with efficient discretization techniques of projection methods.

The grid we construct covers the high-probability area of the state space roughly uniformly. In Figure 1a and Figure 1b, we show a two-dimensional example of such a grid.



The technique we use to construct the grid in the figure is called an ε-*distinguishable set* (EDS) technique, and it consists in constructing a set of points, which are situated at the distance at least ε from one another, where $\varepsilon > 0$ is a parameter. In this paper, we establish computational complexity, dispersion, cardinality and degree of uniformity of the EDS grid constructed on simulated series. Furthermore, we perform the worst-case analysis, and we relate our results to recent mathematical literature on covering problems (see, Temlyakov, 2011) and random sequential packing problems (see, Baryshnikov et al., 2008).

Our construction of the EDS grid relies on the assumption that a solution to the model is known. Since the solution is unknown in the beginning, we proceed iteratively: guess a solution, simulate the model, construct an EDS grid, solve the model on that grid using a projection method, and iterate on these steps until the grid converges.

We complement the efficient EDS grid with other computational techniques suitable for high-dimensional problems, namely, low-cost monomial integration rules and a fixed-point iteration method for finding parameters of the equilibrium rules. Taken together, these techniques make the EDS algorithm tractable in problems with high dimensionality – hundreds of state variables!

We first apply the EDS method to the standard neoclassical growth models with one and multiple agents (countries). The EDS method delivers accuracy levels comparable to the best accuracy attained in the related literature. In particular, we are able to compute global quadratic solutions for equilibrium problems with up to 80 state variables on a desktop computer using a serial Matlab software (the running

time ranges from 30 seconds to 24 hours). The maximum unit-free approximation error on a stochastic simulation is always smaller than 0.01%.

Our second and more novel application is a new Keynesian model which includes a Taylor rule with a zero lower bound (ZLB) on nominal interest rates. This model has eight state variables and is characterized by a kink in equilibrium rules due to the ZLB. We parameterize the model using the estimates of Smets and Wouters (2003, 2007), and Del Negro et al. (2007). The EDS method is tractable for polynomial degrees 2 and 3: the running time is less than 25 minutes in all cases considered. For comparison, we also assess the performance of perturbation solutions of orders 1 and 2. We find that if the volatility of shocks is low and if we allow for negative nominal interest rate, both the EDS and perturbation methods deliver sufficiently accurate solutions. However, if either the ZLB is imposed or the volatility of shocks increases, the perturbation method is significantly less accurate than the EDS method. In particular, under some empirically relevant parameterizations, the perturbation methods of orders 1 and 2 produce errors that are as large as 25% and 38% on a stochastic simulation, while the corresponding errors for the EDS method are less than 5%. The difference between the EDS and perturbation solutions is economically significant. When the ZLB is imposed, the perturbation method considerably understates the duration of the ZLB episodes and the magnitude of the crises.

The EDS projection method can be used to accurately solve small-scale models that were previously studied using other global methods.[1] However, a comparative advantage of the EDS algorithm is its ability to solve large-scale problems that other methods find intractable or expensive. The speed of the EDS algorithm also makes it potentially useful in estimation methods that solve economic models at many parameters vectors; see Fernández-Villaverde and Rubio-Ramírez (2007) for a discussion. Finally, the EDS grid can be used in applications unrelated to solution methods that require to produce a discrete approximation to the ergodic distribution of a stochastic process with a continuous density function.

The rest of the paper is as follows: In Section 2, we describe the construction of the EDS grid and establish its properties. In Section 3, we integrate the EDS grid into a projection method for solving dynamic economic models. In Section 4, we apply the EDS algorithm to solve one- and multi-agent neoclassical growth models. In Section 5, we compute a solution to a new Keynesian model with the ZLB. In Section 6, we conclude.

[1]For reviews of methods for solving dynamic economic models, see Taylor and Uhlig (1990), Gaspar and Judd (1997), Judd (1998), Marimon and Scott (1999), Santos (1999), Christiano and Fisher (2000), Aruoba, Fernández-Villaverde and Rubio-Ramírez (2006), Den Haan (2010), and Kollmann, Maliar, Malin and Pichler (2011).

2 A discrete approximation to the ergodic set

In this section, we first introduce a technique that produces a discrete approximation to the ergodic set of a stochastic process with a continuous density function, we then establish properties of the proposed approximation, and we finally relate our results to mathematical literature. Later, we will use the resulting discrete approximation as a grid of a projection-style solution method.

2.1 A class of stochastic processes

We focus on a class of discrete-time stochastic processes that can be represented in the form

$$x_{t+1} = \varphi\left(x_t, \epsilon_{t+1}\right), \qquad t = 0, 1, ..., \tag{1}$$

where $\epsilon \in E \subseteq \mathbb{R}^p$ is a vector of p independent and identically distributed shocks, and $x \in X \subseteq \mathbb{R}^d$ is a vector of d (exogenous and endogenous) state variables. The distribution of shocks is given by a probability measure Q defined on a measurable space $(E, \mathbb{E})$, and x is endowed with its relative Borel σ-algebra denoted by $\mathbb{X}$.

Many dynamic economic models have equilibrium laws of motion for state variables that can be represented by a stochastic system in the form (1). For example, the standard neoclassical growth model, described in Section 4, has the laws of motion for capital and productivity that are given by $k_{t+1} = K\left(k_t, a_t\right)$ and $a_{t+1} = a_t^{\rho} \exp\left(\epsilon_{t+1}\right)$, respectively, where $\epsilon_{t+1} \sim \mathcal{N}\left(0, \sigma^2\right)$, $\sigma > 0$ and $\rho \in (-1, 1)$; by setting $x_t \equiv \left(k_t, a_t\right)$, we arrive at (1).

To characterize the dynamics of (1), we use the following definitions.

Def 1. *A transition probability is a function $\mathcal{P} : X \times X \to [0, 1]$ that has two properties: (i) for each measurable set $\mathcal{A} \in \mathbb{X}$, $\mathcal{P}\left(\cdot, \mathcal{A}\right)$ is $\mathbb{X}$-measurable function; and (ii) for each point $x \in X$, $\mathcal{P}\left(x, \cdot\right)$ is a probability measure on $(X, \mathbb{X})$.*

Def 2. *An (adjoint) Markov operator is a mapping $\mathcal{M}^* : X \to X$ such that $\mu_{t+1}\left(\mathcal{A}\right) = \left(\mathcal{M}^* \mu_t\right)\left(\mathcal{A}\right) \equiv \int \mathcal{P}\left(x, \mathcal{A}\right) \mu_t\left(dx\right)$.*

Def 3. *An invariant probability measure μ is a fixed point of the Markov operator $\mathcal{M}^*$ satisfying $\mu = \mathcal{M}^* \mu$.*

Def 4. *A set $\mathcal{A}$ is called invariant if $\mathcal{P}\left(x, \mathcal{A}\right) = 1$ for all $x \in \mathcal{A}$. An invariant set $\mathcal{A}^*$ is called ergodic if it has no proper invariant subset $\mathcal{A} \subset \mathcal{A}^*$.*

Def 5. *An invariant measure μ is called ergodic if either $\mu\left(\mathcal{A}\right) = 0$ or $\mu\left(\mathcal{A}\right) = 1$ for every invariant set $\mathcal{A}$.*

These definitions are standard to the literature on dynamic economic models; see Stokey, Lucas and Prescott (1989). $\mathcal{P}\left(x, \mathcal{A}\right)$ is the probability that the stochastic system (1) whose today's state is $x_t = x$ will move tomorrow to a state $x_{t+1} \in \mathcal{A}$. The

Markov operator $\mathcal{M}^*$ maps today's probability into tomorrow's probability, namely, if $\mu_t(\mathcal{A})$ is the probability that the system (1) is in $\mathcal{A}$ at t, then $(\mathcal{M}^*\mu_t)(\mathcal{A})$ is the probability that the system will remain in the same set at $t+1$. Applying the operator $\mathcal{M}^*$ iteratively, we can describe the evolution of the probability starting from a given $\mu_0 \in \mathbb{X}$. An invariant probability measure μ is a steady state solution of the stochastic system (1). An invariant set $\mathcal{A}$ is the one that keeps the system (1) forever in $\mathcal{A}$, and an ergodic set $\mathcal{A}^*$ is an invariant set of the smallest possible size. Finally, an invariant probability measure is ergodic if all the probability is concentrated in just one of the invariant sets.

The dynamics of (1) produced by economic models can be very complex. In particular, the Markov process (1) may have no invariant measure or may have multiple invariant measures. These cases represent challenges to numerical methods that approximate solutions to dynamic economic models. However, there is another challenge that numerical methods face – the curse of dimensionality. The most regular problem with a unique, smooth and well-behaved solution can become intractable when the dimensionality of the state space gets large. The challenge of high dimensionality is the focus of our analysis. We employ the simplest possible set of assumptions that allows us to expose and to test computational techniques that are tractable in high-dimensional applications.

Assumption 1. *There exists a unique ergodic set $\mathcal{A}^*$ and the associated ergodic measure μ.*

Assumption 2. *The ergodic measure μ admits a representation in the form of a density function $g : X \to \mathbb{R}^+$ such that $\int_{\mathcal{A}} g(x)\,dx = \mu(\mathcal{A})$ for every $\mathcal{A} \subseteq \mathbb{X}$.*

Let us comment on these assumptions. The existence of invariant probability measure μ follows by Krylov-Bogolubov theorem under assumptions of a"tightness" and weak continuity (Feller property) of the operator $(\mathcal{M}^*\mu)$; see Stachursky (2009, Theorem 11.2.5). The "tightness" assumption can be replaced with an assumption that X is compact; see Stokey, Lucas and Prescott (1989, Theorem 12.10). The existence and uniqueness of the ergodic probability measure require far more restrictive assumptions such as Doeblin and strong "mixing" type of conditions; see Stokey, Lucas and Prescott (1989, Theorems 11.9 and 11.10, respectively). Finally, the existence of a density function is equivalent to the existence of Radon-Nykodim derivative of μ with respect to the Lebesgue measure; see, e.g., Stachursky (2009, Theorem 9.1.18).

2.2 A two-step EDS technique for approximating the ergodic set

We propose a two-step procedure for forming a discrete approximation to the ergodic set. First, we identify an area of the state space that contains nearly all the probability mass. Second, we cover this area with a finite set of points that are roughly evenly spaced.

2.2.1 An essentially ergodic set

We define a high-probability area of the state space using the level set of the density function.

Def 6. *A set $\mathcal{A}^\eta \subseteq \mathcal{A}^*$ is called a η-level ergodic set if $\eta > 0$ and*

$$\mathcal{A}^\eta \equiv \{x \in X : g(x) \geq \eta\}.$$

The mass of $\mathcal{A}^\eta$ under the density $g(x)$ is equal to $p(\eta) \equiv \int_{g(x)\geq\eta} g(x)\,dx$. If $p(\eta) \approx 1$, then $\mathcal{A}^\eta$ contains all X except for points where the density is lowest, in which case $\mathcal{A}^\eta$ is called an *essentially ergodic set*.

By construction, the correspondence $\mathcal{A}^\eta : \mathbb{R}^+ \rightrightarrows \mathbb{R}^d$ maps η to a compact set. The correspondence $\mathcal{A}^\eta$ is upper semi-continuous but may be not lower semi-continuous (e.g., if x is drawn from a uniform distribution $[0,1]$). Furthermore, if g is multimodal, then for some values of η, $\mathcal{A}^\eta$ may be disconnected (composed of disjoint areas). Finally, for $\eta > \max_x \{g(x)\}$, the set $\mathcal{A}^\eta$ is empty.

Our approximation to the essentially ergodic set builds on stochastic simulation and relies on the law of iterated logarithm. Formally, let P be a set of n independent random draws $x_1, ..., x_n \subseteq \mathbb{R}^d$ generated with the distribution function $\mu : \mathbb{R}^d \to \mathbb{R}^+$. For a given subset $J \subseteq \mathbb{R}^d$, we define $C(P; J)$ as a characteristic function that counts the number of points from P in J. Let $\mathcal{J}$ be a family generated by the intersection of all subintervals of $\mathbb{R}^d$ of the form $\Pi_{i=1}^d [-\infty, v_i)$, where $v_i > 0$.

Proposition 1 *(Law of iterated logarithm). For every d and every continuous function μ, we have*

$$\lim_{n\to\infty} \left\{ \sup_{J\in\mathcal{J}} \left| \frac{C(P;J)}{n} - \mu(J) \right| \cdot \left(\frac{2n}{\log\log n} \right)^{1/2} \right\} = 1, \quad a.e. \tag{2}$$

Proof. See Kiefer (1961, Theorem 2). ■

That is, the empirical distribution function $\widehat{\mu}(J) \equiv \frac{C(P;J)}{n}$ converges asymptotically to the true distribution function $\mu(J)$ for every $J \in \mathcal{J}$ at the rate given in (2).

We use the following algorithm to select a subset of simulated points that belongs to an essentially ergodic set $\mathcal{A}^{\eta}$.

(Algorithm $\mathcal{A}^{\eta}$): Selection of points within an essentially ergodic set.
Step 1. Simulate (1) for T periods.
Step 2. Select each κth point to get a set P of n points $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$.
Step 3. Estimate the density function $\widehat{g}(x_i) \approx g(x_i)$ for all $x_i \in P$.
Step 4. Remove all points for which the density is below η.

In Step 2, we include in the sample P only each κth observation to make random draws (approximately) independent. As concerning Step 3, there are various methods in statistics that can be used to estimate the density function from a given set of data; see Scott and Sain (2005) for a review. We use one of such methods, namely, a multivariate kernel algorithm with a normal kernel which estimates the density function in a point x as

$$\widehat{g}(x) = \frac{1}{n(2\pi)^{d/2}\overline{h}^{d}} \sum_{i=1}^{n} \exp\left[-\frac{D(x, x_i)}{2\overline{h}^{2}}\right], \tag{3}$$

where $\overline{h}$ is the bandwidth parameter, and $D(x, x_i)$ is the distance between x and x_i. The complexity of Algorithm $\mathcal{A}^{\eta}$ is $O(n^2)$ because it requires to compute pairwise distances between all the sample points. Finally, in Step 3, we do not choose the density cutoff η but a fraction of the sample to be removed, δ, which is related to η by $p(\eta) = \int_{g(x)\geq\eta} g(x)\,dx = 1 - \delta$. For example, $\delta = 0.05$ means that we remove 5% of the sample which has the lowest density.

2.2.2 An ε-distinguishable set (EDS)

Our next objective is to construct a uniformly-spaced set of points that covers the essentially ergodic set (to have a uniformly-spaced grid for a projection method). We proceed by selecting an ε-distinguishable subset of simulated points in which all points are situated at least on the distance ε from one another. Simulated points are not uniformly-spaced but the EDS subset will be roughly uniform, as we will show in Section 2.3.

Def 7. *Let (X, D) be a bounded metric space. A set P^{ε} consisting of points $x_1^{\varepsilon}, ..., x_M^{\varepsilon} \in X \subseteq \mathbb{R}^d$ is called ε-distinguishable if $D\left(x_i^{\varepsilon}, x_j^{\varepsilon}\right) > \varepsilon$ for all $1 \leq i, j \leq M : i \neq j$, where $\varepsilon > 0$ is a parameter.*

EDSs are used in mathematical literature that studies the entropy; see Temlyakov (2011) for a review. This literature focuses on a problem of constructing an EDS that covers a given subset of $\mathbb{R}^d$ (such as a multidimensional hypercube). We study a different problem, namely, we construct an EDS for a given discrete set of points. To this purpose, we introduce the following algorithm.

(**Algorithm P^ε): Construction of an** EDS.
Let P be a set of n point $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$.
Let P^ε begin as an empty set, $P^\varepsilon = \{\varnothing\}$.
Step 1. Select $x_i \in P$. Compute $D(x_i, x_j)$ to all x_j in P.
Step 2. Eliminate from P all x_j for which $D(x_i, x_j) < \varepsilon$.
Step 3. Add x_i to P^ε and eliminate it from P.
Iterate on Steps 1-3 until all points are eliminated from P.

The cost of this algorithm is assessed below.

Proposition 2 *The complexity of Algorithm P^ε is of order $O(nM)$.*

Proof. Consider the worst-case scenario for the complexity. We take $x_1 \in P$, compute $n-1$ distances to all other points and obtain that all such distances are larger than ε so that no point is eliminated. Further, we take $x_2 \in P$, compute $n-2$ distances to all other points and again, no point is eliminated. We proceed till $x_M \in P$ for which we compute $n-M$ distances. Thus, the first M points are placed into P^ε. Subsequently, we eliminate the remaining $n-M$ points. Under this scenario, the complexity is $(n-1)+(n-2)...+(n-M) = \sum_{i=1}^{M}(n-i) = nM - \frac{M(M+1)}{2} \leq nM$. ■

When no points are eliminated from P, i.e., $M = n$, the complexity is quadratic, $O(n^2)$. However, the number of points M in an EDS is bounded from above if X is bounded; see Proposition 4. This means that asymptotically, when $n \to \infty$, the complexity of Algorithm P^ε is linear, $O(n)$.

2.2.3 Distance between points

Both estimating the density function and constructing an EDS requires us to measure the distance between simulated points. Generally, variables in economic models have different measurement units and are correlated. This affects the distance between the simulated points and hence, affects the resulting EDS. Therefore, prior to using Algorithm $\mathcal{A}^\eta$ and Algorithm P^ε, we normalize and orthogonalize the simulated data.

To be specific, let $X \in \mathbb{R}^{n\times d}$ be a set of simulated data normalized to zero mean and unit variance. Let $x_i \equiv \left(x_i^1, ..., x_i^d\right)$ be an observation $i = 1$ (there are n observations), and let $x^\ell \equiv \left(x_1^\ell, ..., x_n^\ell\right)^\top$ be a variable ℓ (there are d variables), i.e., $X = \left(x^1, ..., x^d\right) = (x_1, ..., x_n)^\top$. We first compute the singular value decomposition of X, i.e., $X = UQV^\top$, where $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{d\times d}$ are orthogonal matrices, and $Q \in \mathbb{R}^{d\times d}$ is a diagonal matrix. We then perform a linear transformation of X using PC$\equiv XV$. The variables PC$= \left(\text{PC}^1, ..., \text{PC}^d\right) \in \mathbb{R}^{n\times d}$ are called *principal components* (PCs) of X, and are orthogonal (uncorrelated), i.e., $\left(\text{PC}^{\ell'}\right)^\top \text{PC}^\ell = 0$ for any $\ell' \neq \ell$. As a measure of distance between two observations x_i and x_j, we use the Euclidean distance between their PCs, namely, $D\left(x_i, x_j\right) = \left[\sum_{\ell=1}^d \left(\text{PC}_i^\ell - \text{PC}_j^\ell\right)^2\right]^{1/2}$, where all principal components PC^1, ...,PC^d are normalized to unit variance.

2.2.4 An illustration of the EDS technique

In this section, we will illustrate the EDS technique described above by way of example. We consider the standard neoclassical growth model with a closed-form solution (see Section 4 for a description of this model). We simulate time series for capital and productivity level of length $1,000,000$ periods, and we select a sample of $10,000$ observations by taking each 100th point (to make the draws independent); see Figure 2a. We orthogonalize the data using principal component (PC) transformation, and we normalize the PCs to unit variance; see Figure 2b. We estimate the density function using the multivariate kernel algorithm with the standard bandwidth of $\overline{h} = n^{-1/(d+4)}$, and we remove from the sample 5% of simulated points in which the density is the lowest; see Figure 2c. We construct an EDS; see Figure 2d. We plot such a set in the PC and original coordinates in Figure 2e and Figure 2f, respectively. As we see, the EDS technique delivers a set of points that covers the same area as does the set of simulated points but that is spaced roughly uniformly.[2] In Section 2.3, we will characterize the properties of the constructed EDS analytically.

[2] Our two-step procedure produces an approximation not only to the ergodic set but also to the ergodic distribution (because in the first step, we estimate the density function in all simulated points including those that form an ε-distinguishable set). The density weights show what fraction of the sample each "representative" point represents, and can be used to construct weighted-average approximations. Given our purpose to construct a set of evenly-spaced points, we do not use the density weights, treating all points equally.

Figure 2a. Simulated points

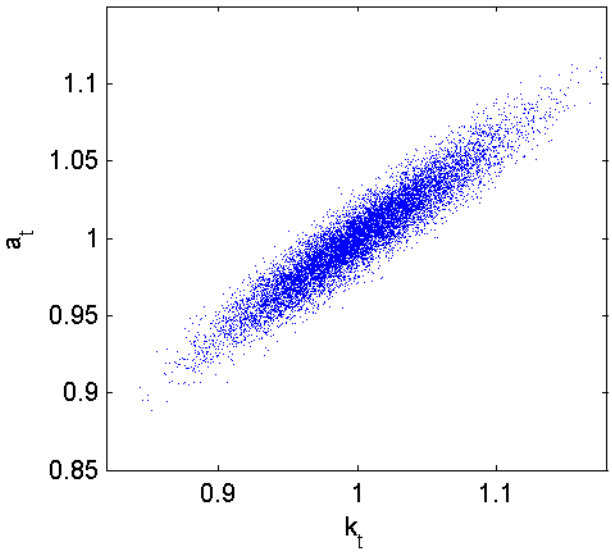


Figure 2b. Principal components (PCs)

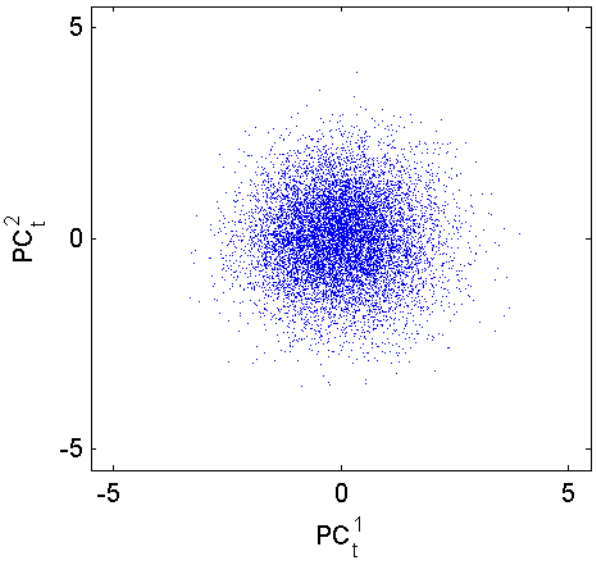


Figure 2c. Density levels on PCs

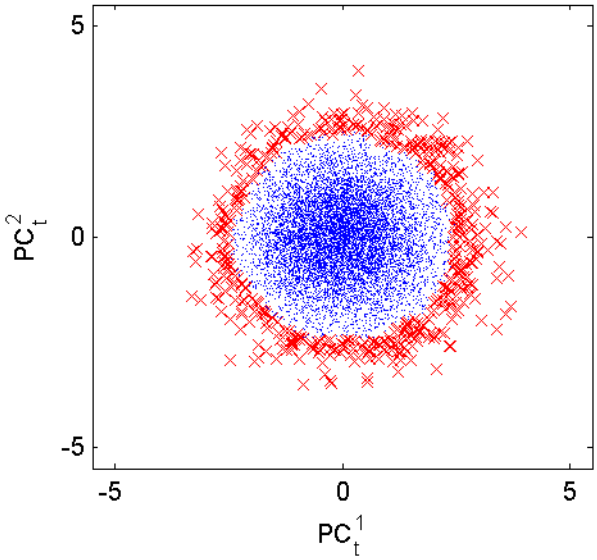


Figure 2d. Constructing EDS

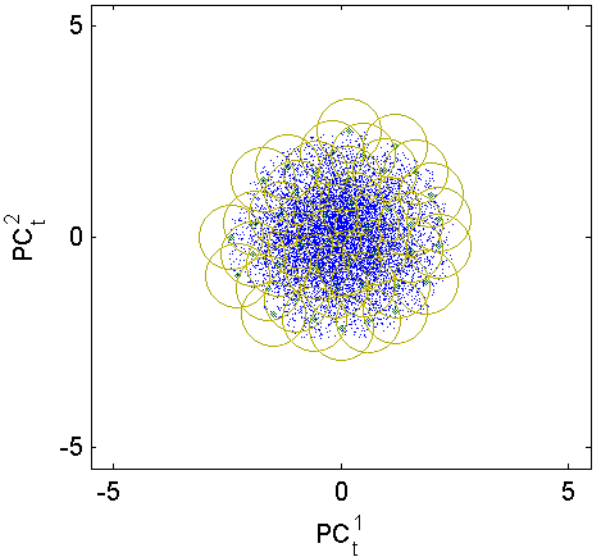


Figure 2e. EDS on PCs

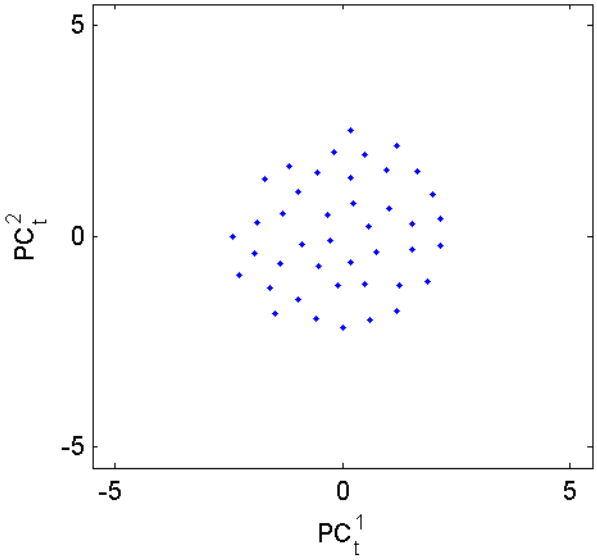
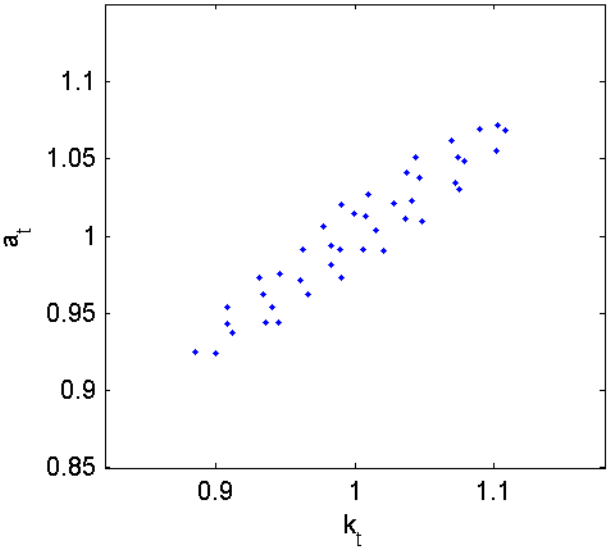


Figure 2f. EDS on original data



2.2.5 Other procedures for approximating the ergodic set

We have described one specific procedure for forming a discrete approximation to the essentially ergodic set of stochastic process (1). Below, we outline other procedures that can be used for this purpose.

First, our two-step procedure has a complexity of order $O(n^2)$ because the kernel algorithm computes pairwise distances between all observations in the sample. This is not a problem for the size of applications we study in the present paper, however, it might be expensive for larger samples. In Appendix A, we describe a procedure that has a lower complexity, namely, $O(nM)$. We specifically invert the steps in the two-step procedure: we first construct an EDS on all simulated points, and we then remove from the EDS those points where the density is low.

Second, we can use methods from cluster analysis to select a set of representative points from a given set of simulated points (instead of constructing an EDS). Namely, we partition the simulated data into clusters (groups of closely-located points), and we replace each cluster with one representative point. This technique is exposed in Appendix B using two clustering methods, an agglomerative hierarchical method and K-means method.[3]

2.3 Properties of EDSs

In this section, we characterize the dispersion of points, the number of points, and the degree of uniformity of the constructed EDS. Also, we discuss the relation of our results to recent mathematical literature.

2.3.1 Dispersion of points in the EDS

We borrow the notion of dispersion from the literature on quasi-Monte Carlo optimization methods; see, e.g., Niederreiter (1992, p.148) for a review. Dispersion measures are used to characterize how dense is a given set of points in a given area of the state space.

Def 7. *Let P be a set consisting of points $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$, and let (X, D) be a bounded metric space. The dispersion of P in X is given by*

$$d_n(P; X) = \sup_{x \in X} \inf_{1 \leq i \leq n} D(x, x_i), \tag{4}$$

[3]The clustering methods were used to produce all the numerical results in the earlier versions of the paper, Judd et al. (2010, 2011). In the current version of the paper, we switch to EDSs because their properties are easier to characterize analytically and their construction has a lower cost. In our examples, projection methods operating on cluster grids and those operating on EDSs deliver comparable accuracy of solutions.

where D is a (Euclidean) metric on X.

Let $B(x;r)$ denote a ball with the center x and radius r. Then, $d_n(P;X)$ is the smallest possible radius r such that the family of closed balls $B(x_1;r), ..., B(x_n;r)$ covers X.

Def 8. *Let S be a sequence of elements on X, and let $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$ be the first n terms of S. The sequence S is called low-dispersion if $\lim_{n\to\infty} d_n(S;X) = 0$.*

In other words, a sequence S is low-dispersion if it becomes increasingly dense when $n \to \infty$. Below, we establish bounds on the dispersion of points in an EDS.

Proposition 3 *Let P be any set of n points $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$ with a dispersion $d_n(P;X) < \varepsilon$. Let (X, D) be a bounded metric space, and let P^ε be an EDS $x_1^\varepsilon, ..., x_M^\varepsilon$ constructed by Algorithm P^ε. Then, the dispersion of P^ε is bounded by $\varepsilon < d_M(P^\varepsilon;X) < 2\varepsilon$.*

Proof. The first equality follows because for each $x_i^\varepsilon \in P^\varepsilon$, Algorithm P^ε removes all points $x_i \in P$ such that $D\left(x_i, x_j^\varepsilon\right) < \varepsilon$. To prove the second inequality, let us assume that $d_M(P^\varepsilon;X) \geq 2\varepsilon$ toward a contradiction.

(i) Then, there is a point $x \in X$ for which $\inf_{x_j^\varepsilon \in P^\varepsilon} D\left(x, x_j^\varepsilon\right) \geq 2\varepsilon$, i.e., all points in EDS P^ε are situated at the distance at least 2ε from x (because we assumed $d_M(P^\varepsilon;X) \geq 2\varepsilon$).

(ii) An open ball $B(x;\varepsilon)$ contains at least one point $x^* \in P$. This is because $d_n(P;X) < \varepsilon$ implies $\inf_{x_i \in P} D(x, x_i) < \varepsilon$ for all x, i.e., the distance between any point $x \in X$ and its closest neighbor from P is smaller than ε.

(iii) Algorithm P^ε does not eliminate x^*. This algorithm eliminate only those points around all $x_j^\varepsilon \in P^\varepsilon$ that are situated on the distance smaller than ε, whereas any point inside $B(x;\varepsilon)$ is situated on the distance larger than ε from any $x_j^\varepsilon \in P^\varepsilon$.

(iv) For x^*, we have $\inf_{x_j^\varepsilon \in P^\varepsilon} D\left(x^*, x_j^\varepsilon\right) > \varepsilon$, i.e., x^* is situated at the distance larger than ε from any $x_j^\varepsilon \in P^\varepsilon$.

Then, x^* must belong to EDS P^ε. Since $x^* \in B(x;\varepsilon)$, we have $D(x, x^*) < \varepsilon < 2\varepsilon$, a contradiction. ■

Proposition 3 states that any $x \in X$ has a neighbor $x_j^\varepsilon \in P^\varepsilon$ which is situated at most at the distance 2ε. The dispersion of points in an EDS goes to zero as $\varepsilon \to 0$.

2.3.2 Number of points in the EDS

The number of points in an EDS is unknown a priory. It depends on the value of ε and the order in which the points from P are processed. Temlyakov (2011, Theorem

3.3 and Corollary 3.4) provides the bounds on the number of points in a specific class of EDSs, namely, those that cover a unit ball with balls of radius ε. We also confine our attention to a set X given by a ball, however, in our case, balls of radius ε do not provide a covering of X. The bounds for our case are established in the following proposition.

Proposition 4 *Let P be any set of n points $x_1, ..., x_n \in B(0,r) \subseteq \mathbb{R}^d$ with a dispersion $d_n(P;X) < \varepsilon$. Then, the number of points in P^ε constructed by Algorithm P^ε is bounded by $\left(\frac{r}{2\varepsilon}\right)^d \leq M \leq \left(1+\frac{r}{\varepsilon}\right)^d$.*

Proof. To prove the first inequality, notice that, by Proposition 3, the balls $B(x_1^\varepsilon; 2\varepsilon)$, ..., $B(x_M^\varepsilon; 2\varepsilon)$ cover X. Hence, we have $M\lambda_d(2\varepsilon)^d \geq \lambda_d r^d$, where λ_d is the volume of a d-dimensional unit ball. This gives the first inequality.

To prove the second inequality, we consider an EDS on $B(0;r)$ that has the maximal cardinality $M^{\max}$. Around each point of such a set, we construct a balls with the radius ε. By definition, the distance between any two points in an EDS is larger than ε, so the balls $B(x_1^\varepsilon;\varepsilon), ..., B(x_M^\varepsilon;\varepsilon)$ are all disjoint. To obtain an upper bound on M, we must construct a set that encloses all these balls. The ball $B(0;r)$ does not necessarily contain all the points from these balls, so we add an open ball $B(0;\varepsilon)$ to each point of $B(0;r)$ in order to extend the frontier by ε. This gives us a set $B(0;r+\varepsilon) = B(0;r) \oplus B(0;\varepsilon) \equiv \{y : y = x+b, x \in B(0;r), b \in B(0;\varepsilon)\}$. Since the ball $B(0;r+\varepsilon)$ encloses the balls $B(x_1^\varepsilon;\varepsilon), ..., B(x_M^\varepsilon;\varepsilon)$ and since $M \leq M^{\max}$, we have $M\lambda_d(\varepsilon)^d \leq \lambda_d(r+\varepsilon)^d$. This yields the second inequality. ■

In some applications, we need to construct an EDS that has a given target number of points $\overline{M}$ (for example, in a projection method, we may need a grid with a certain number of grid points). To this purpose, we can use a simple bisection method. We fix ε_1 and ε_2 such that $M(\varepsilon_1) \leq \overline{M} \leq M(\varepsilon_2)$, take $\varepsilon = \frac{\varepsilon_1+\varepsilon_2}{2}$, construct an EDS and find $M(\varepsilon)$; if $M(\varepsilon) > \overline{M}$, then we set $\varepsilon_1 = \varepsilon$ and otherwise, we set $\varepsilon_2 = \varepsilon$, and proceed iteratively until $M(\varepsilon)$ converges to some limit (which was close to the target value $\overline{M}$ in our experiments). To find the initial values of ε_1 and ε_2, we use the bounds established in Proposition 4, namely, we set $\varepsilon_1 = 0.5 r_1 \overline{M}^{-1/d}$ and $\varepsilon_2 = r_2\left(\overline{M}^{1/d} - 1\right)^{-1}$, where r_1 and r_2 are, respectively, the largest and smallest PCs of the simulated points (since the essentially ergodic set is not necessarily a hypersphere, as is assumed in Proposition 4, we take r_1 and r_2 to be the radii of the limiting hyperspheres that contain none and all PCs of the simulated points, respectively).

2.3.3 Discrepancy

We now analyze the degree of uniformity of EDSs. The standard notion of uniformity in the literature is discrepancy from the uniform distribution; see Niederreiter (1992, p. 14).

Def 10. *Let P be a set consisting of points $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$, and let $\mathcal{J}$ be a family of Lebesgue-measurable subsets of X. The discrepancy of P under $\mathcal{J}$ is given by $\mathcal{D}_n(P;\mathcal{J}) = \sup_{J\in\mathcal{J}} \left|\frac{C(P;J)}{n} - \lambda(J)\right|$, where $C(P;J)$ counts the number of points from P in J, and $\lambda(J)$ is a Lebesgue measure of J.*

$\mathcal{D}_n(P;\mathcal{J})$ measures the discrepancy between the fraction of points $\frac{C(P;J)}{n}$ contained in J, and the fraction of space $\lambda(J)$ occupied by J. If the discrepancy is low, $\mathcal{D}_n(P;\mathcal{J}) \approx 0$, the distribution of points in X is close to uniform. The measure of discrepancy commonly used in the literature is the star discrepancy.

Def 11. *The star discrepancy $\mathcal{D}_n^*(P;\mathcal{J})$ is defined as the discrepancy of P over the family J generated by the intersection of all subintervals of $\mathbb{R}^d$ of the form $\Pi_{i=1}^d [-\infty, v_i)$, where $v_i > 0$.*

Let S be a sequence of elements on X, and let $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$ be the first n terms of S. Niederreiter (1992, p. 32) suggests to call a sequence S low-discrepancy if $\mathcal{D}_n^*(S;\mathcal{J}) = O\left(n^{-1}(\log n)^d\right)$, i.e., if the star discrepancy converges to zero asymptotically at a rate at least of order $n^{-1}(\log n)^d$.

The star discrepancy of points which are randomly drawn from a uniform distribution $[0,1]^d$ converges to zero asymptotically, $\lim_{n\to\infty} \mathcal{D}_n^*(S;\mathcal{J}) = 0$, a.e. The rate of convergence follows directly from the law of iterated logarithm stated in Proposition 1, and it is $(\log\log n)^{1/2}(2n)^{-1/2}$; see Niederreiter (1992, p. 166-168) for a general discussion on how to use Kiefer's (1961) results for assessing the discrepancy of random sequences.

If a sequence is a low-discrepancy one, then it is also a low-dispersion one; see Niederreiter (1992, Theorem 6.6). Indeed, a sequence that covers X uniformly must become increasingly dense everywhere on X as $n \to \infty$. However, the converse is not true. A sequence that becomes increasingly dense on X as $n \to \infty$ does not need to become increasingly uniform since density may be distributed unevenly. Thus, the result of Proposition 3 that the dispersion of an EDS converges to zero in the limit does not mean that its discrepancy does so.

Nonetheless, we can show that for any density function g, the discrepancy of an EDS is bounded on a multidimensional sphere.

Def 12. *The spherical discrepancy $\mathcal{D}_M^s(P^\varepsilon;\mathcal{B})$ is defined as the discrepancy of P over the family J generated by the intersection of d-dimensional open balls $B(0;r)$*

centered at 0 *with the radius* $r \leq 1$.

Proposition 5 *Let* P *be any set of* n *points* $x_1, ..., x_n \in B(0;1) \subseteq \mathbb{R}^d$ *with a dispersion* $d_n(P;X) < \varepsilon$. *Then, the discrepancy of an EDS constructed by Algorithm* P^ε *under* $\mathcal{B}$ *is bounded by* $\mathcal{D}_M^s(P^\varepsilon;\mathcal{B}) \leq \frac{\sqrt{2^d}-1}{\sqrt{2^d}+1}$.

Proof. Let $\lambda \equiv \lambda(B(0;r)) = \lambda(B(0;1))\, r^d = r^d$ be a Lebesgue measure of $B(0;r)$, and let $\frac{C(P^\varepsilon;B(0;r))}{M}$ be the fraction of points from P^ε in the ball $B(0;r)$. Consider the case when $\lambda(B(0;r)) \geq \frac{C(P^\varepsilon;B(0;r))}{M}$ and let us compute the maximum discrepancy $\mathcal{D}_M^{\max}(P^\varepsilon;\mathcal{B})$ across all possible EDSs using the results of Proposition 4,

$$\mathcal{D}_M^{\max}(P^\varepsilon;\mathcal{B}) = \lambda - \frac{C^{\min}(P^\varepsilon;B(0;r))}{C^{\min}(P^\varepsilon;B(0;r)) + C^{\max}(P^\varepsilon;B(0,1)\setminus B(0;r))} \leq$$
$$\lambda - \frac{C^{\min}(P^\varepsilon;B(0,r))}{C^{\min}(P^\varepsilon;B(0;r)) + C^{\max}(P^\varepsilon;B(0;1)) - C^{\max}(P^\varepsilon;B(0;r))}$$
$$= r^d - \frac{\left(\frac{r}{2\varepsilon}\right)^d}{\left(\frac{r}{2\varepsilon}\right)^d + \left(1+\frac{1}{\varepsilon}\right)^d - \left(1+\frac{r}{\varepsilon}\right)^d}$$
$$\leq r^d - \frac{\left(\frac{r}{2\varepsilon}\right)^d}{\left(\frac{r}{2\varepsilon}\right)^d + \left(\frac{1}{\varepsilon}\right)^d - \left(\frac{r}{\varepsilon}\right)^d} = \lambda - \frac{\lambda}{2^d - \lambda\left[2^d - 1\right]} \equiv F(\lambda),$$

where $C^{\min}(P^\varepsilon;X)$ and $C^{\max}(P^\varepsilon;X)$ are respectively the minimum and maximum cardinality of an EDS P^ε on X. Maximizing $F(\lambda)$ with respect to λ yields $\lambda^* = \frac{\sqrt{2^d}}{\sqrt{2^d}+1}$ and $F(\lambda^*) = \frac{\sqrt{2^d}-1}{\sqrt{2^d}+1}$, as is claimed. The case when $\lambda(B(0;r)) \leq \frac{C(P^\varepsilon;B(0;r))}{M}$ leads to the same bound. ■

2.4 Relation to mathematical literature

We now compare our results to mathematical literature that focuses on related problems.

2.4.1 Existence results for a covering-number problem

Temlyakov (2011) studies the problem of finding a covering number – a minimum number of balls of radius ε which cover a given compact set (such as a d-dimensional hypercube or hypershpere). In particular, he shows the following result. There exists an EDS P^ε on a unit hypercube $[0,1]^d$ whose star discrepancy is bounded by

$$\mathcal{D}_M^*(P^\varepsilon;\mathcal{J}) \leq c \cdot d^{3/2} \left[\max\{\ln d, \ln M\}\right]^{1/2} M^{-1/2}, \tag{5}$$

where c is a constant; see Temlyakov (2011, Proposition 6.72). The discrepancy of such an EDS converges to 0 as $M \to \infty$ (i.e., $\varepsilon \to 0$). However, constructing an EDS with the property (5) is operationally difficult and costly. Also, Temlyakov (2011) selects points from a compact subset of $\mathbb{R}^d$, and his analysis cannot be directly applied to our problem of finding an ε-distinguishable subset of a given finite set of points.

2.4.2 Probabilistic results for random sequential packing problems

Probabilistic analysis of an EDS is non-trivial because points in such a set are spatially dependent: once we place a point in an EDS, it affects the placement of all subsequent points.

Some related probabilistic results are obtained in the literature on a random sequential packing problem.[4] Consider a bounded set $X \subseteq \mathbb{R}^d$ and a sequence of d-dimensional balls whose centers are i.i.d. random vectors $x_1, ..., x_n \in X$ with a given density function g. A ball is packed if and only if it does not overlap with any ball which has already been packed. If not packed, the ball is discarded. At saturation, the centers of accepted balls constitute an EDS. A well-known unidimensional example of this general problem is a car-parking model of Rényi (1958). Cars of a length ε park at random locations along the roadside of a length 1 subject to a non-overlap with the previously parked cars. When cars arrive at uniform random positions, Rényi (1958) shows that they occupy about 75% of the roadside at jamming (namely, the expected value is $\lim_{\varepsilon \to 0} E[M]\,\varepsilon \approx 0.748$).

For a multidimensional case, Baryshnikov et al. (2008) show that the sequential packing measure, induced by the accepted balls centers, satisfies the law of iterated logarithm (under some additional assumptions). This fact implies that the discrepancy of EDS converges to 0 asymptotically if the density of points in an EDS is uniform in the limit $\varepsilon \to 0$. However, the density of points in an EDS depends on the density function g of the stochastic process (1) used to produce the data (below, we illustrate this dependence by way of examples). Hence, we have the following negative conclusion: an EDS needs not be uniform in the limit even in the probabilistic sense (unless the density function is uniform).

[4] This problem arises in spacial birth-growth models, germ-grain models, percolation models, spacial-graphs models; see, e.g., Baryshnikov et al. (2008) for a review.

2.4.3 Our best- and worst-case scenarios

One-dimensional versions of our Propositions 4 and 5 have implications for Rényi's (1958) car parking model. Namely, Proposition 4 implies that the cars occupy between 50% and 100% of the roadside ($\frac{1}{2} \leq \lim_{\varepsilon \to 0} M\varepsilon \leq 1$). These are the best- and worst-case scenarios in which the cars are parked on the distances ε and 0 from each other, respectively. (In the former case, evil drivers park their cars to leave as little parking space to other drivers as possible, and in the latter case, a police officer directs the cars to park in a socially efficient way). The density functions that support the worst- and best-scenarios are the ones that contain the Dirac point masses on the distances 2ε and ε, respectively.

Proposition 5 yields the worst-case scenario for discrepancy in Rényi's (1958) model, $\mathcal{D}_M^s\left(P^\varepsilon; \mathcal{B}\right) \leq \frac{\sqrt{2}-1}{\sqrt{2}+1} \approx 0.17$, which is obtained under $\lambda^* = \frac{\sqrt{2}}{\sqrt{2}+1}$. This bound is attainable. Indeed, consider an EDS on $[0,1]$ such that on the interval $[0,\lambda^*]$, all points are situated on a distance ε, and on $[\lambda^*,1]$, all points are situated on the distance 2ε. In the first interval, we have $\frac{\lambda^*}{2\varepsilon} \leq M \leq \frac{\lambda^*}{2\varepsilon} + 1$ points and in the second interval, we have $\frac{1-\lambda^*}{\varepsilon} \leq M \leq \frac{1-\lambda^*}{\varepsilon} + 1$ points. On the first interval, the limiting discrepancy is $\lim_{\varepsilon \to 0} \left[\lambda^* - \frac{\frac{\lambda^*}{2\varepsilon}}{\frac{\lambda^*}{2\varepsilon} + \frac{1-\lambda^*}{\varepsilon}} \right] = \frac{\sqrt{2}-1}{\sqrt{2}+1} \approx 0.17$, which is the same value as implied by Proposition 5. To support this scenario, we assume that g has the Dirac point masses on the distances ε and 2ε in the intervals $[0,\lambda^*]$ and $[\lambda^*,1]$, respectively.

When the dimensionality increases, our bounds become loose. Proposition 4 implies $\left(\frac{1}{2}\right)^d \leq \lim_{\varepsilon \to 0} M\varepsilon^d \leq 1$ which means that M can differ by a factor of 2^d under the worst- and best-case scenarios; for example, when $d = 10$, M can differ by a factor of 1024. Furthermore, when $d = 10$, Proposition 5 implies that $\mathcal{D}_M^s\left(P^\varepsilon; \mathcal{B}\right) \leq \frac{\sqrt{2^{10}}-1}{\sqrt{2^{10}}+1} \approx 0.94$, which is almost uninformative since $\mathcal{D}_M^s\left(P^\varepsilon; \mathcal{B}\right) \leq 1$ by definition. However, we cannot improve upon the general results of Propositions 4 and 5: our examples with Dirac point masses show that there exist density functions g under which the established bounds are attained.

3 Incorporating the EDS grid into projection methods

We refer to the set of points produced by the two-step procedure of Section 2.2 as an *EDS grid*. In this section, we incorporate the EDS grid into projection methods for solving dynamic economic models, namely, we use the EDS grid as a set of points

on which the solution is approximated.

3.1 Comparison of the EDS grid with other grids in the literature

Let us first compare the EDS grid to other grids used in the literature for solving dynamic economic models. We must make a distinction between a geometry of the set on which the solution is computed and a specific discretization of this set. A commonly-used geometry in the related literature is a fixed multidimensional hypercube. Figures 3a-3d plot 4 different discretizations of the hypercube: a tensor-product Chebyshev grid, a low-discrpancy Sobol grid, a sparse Smolyak grid and a monomial grid, respectively (in particular, these grids were used in Judd (1992), Rust (1998), Krueger and Kubler (2004), and Pichler (2011), respectively).

An adaptive geometry is an alternative used by stochastic simulation methods; see Marcet (1988), Smith (1993), Maliar and Maliar (2005), Judd et al. (2011) for examples of methods that compute solutions on simulated series.[5] Focusing on the right geometry can be critical for the cost, as the following example shows.

Example. Consider a vector of uncorrelated random variables $x \in \mathbb{R}^d$ drawn from a multivariate Normal distribution $x \sim \mathcal{N}(0, I_d)$, where I_d is an identity matrix. An essentially ergodic set $\mathcal{A}^\eta$ has the shape of a hypersphere. Let us surround such a hypersphere with a hypercube of a minimum size. For dimensions 2, 3, 4, 5, 10, 30 and 100, the ratio of the volume of a hypersphere to the volume of the hypercube is equal to 0.79, 0.52, 0.31, 0.16, $3 \cdot 10^{-3}$, $2 \cdot 10^{-14}$ and $2 \cdot 10^{-70}$, respectively. These numbers suggest that an enormous savings in cost are possible by focusing on an essentially ergodic set instead of the standard multidimensional hypercube.

However, a stochastic simulation is not an efficient discretization of the high-probability set: a grid of simulated points is unevenly spaced, has many closely-located, redundant points and contains some points in low-density regions.

The EDS grid is designed to combine the best feature of the existing grids. It combines an adaptive geometry (similar to the one used by stochastic simulation methods) with an efficient discretization (similar to that produced by low-discrepancy methods on a hypercube). In Figure 3e, we show an example of a cloud of simulated points of irregular shape, and in Figure 3f, we plot the EDS grid delivered by the two-step procedure of Section 2.2. As we can see, the EDS grid appears to cover the high-probability set uniformly.

[5] For a detailed description of Marcet's (1988) method, see Marcet and Lorenzoni (1999).

Figure 3a. Tensor-product Chebyshev grid

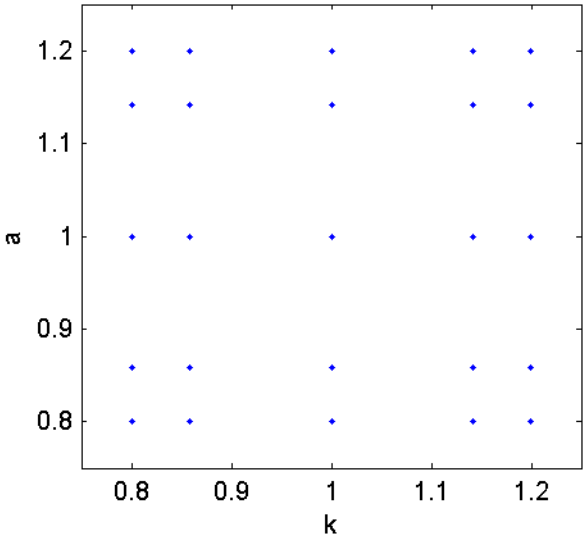


Figure 3b. Sobol sequence

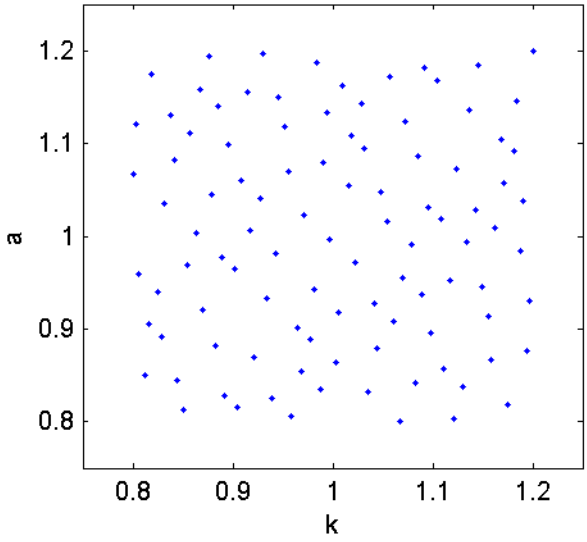


Figure 3c. Smolyak grid

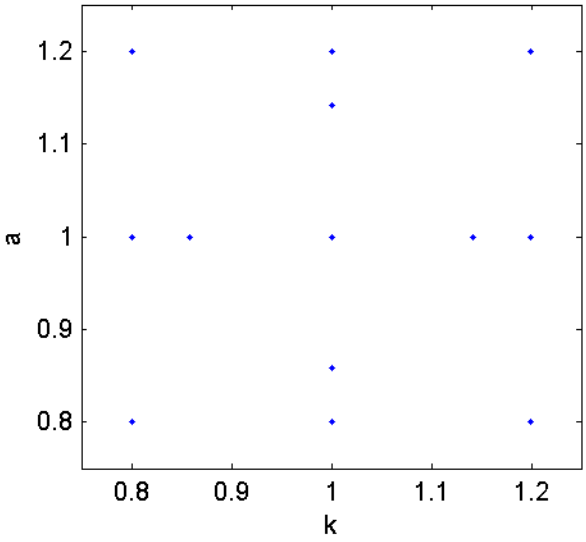


Figure 3d. Monomial grid

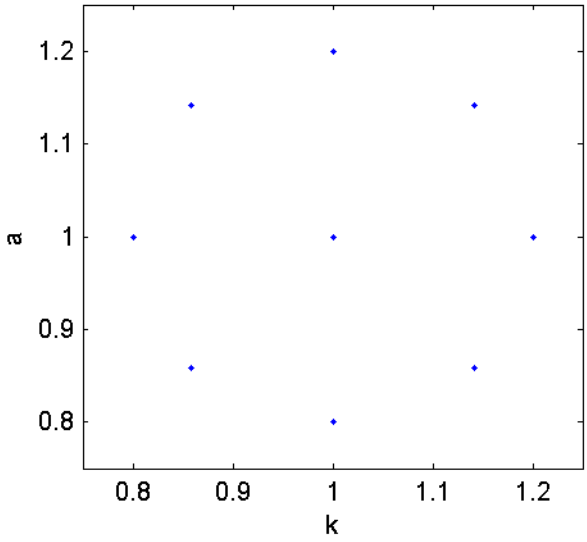


Figure 3e. Simulated points

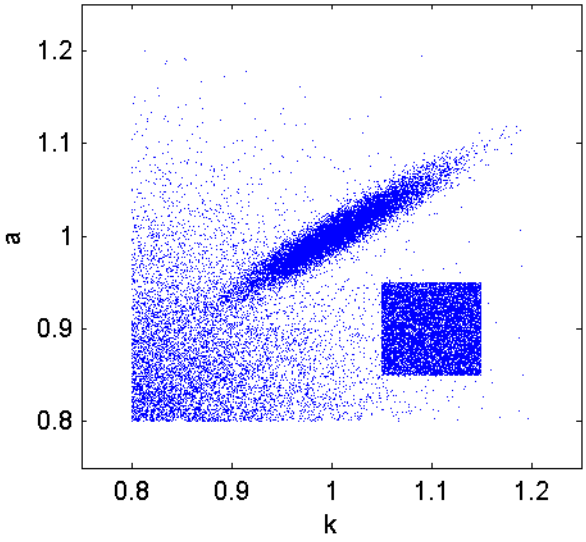
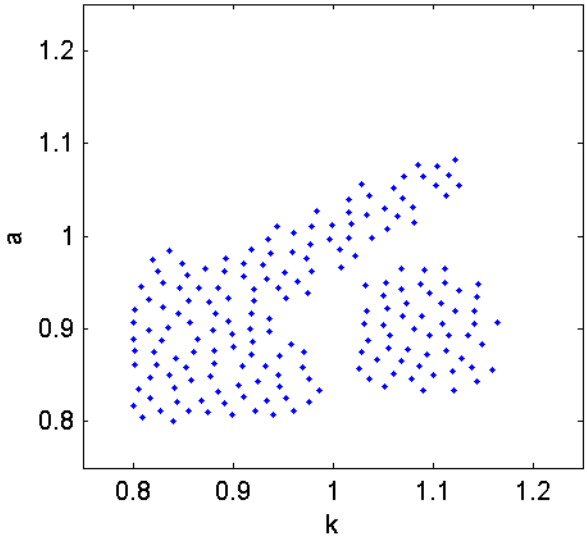


Figure 3f. EDS on simulated points



Tauchen and Hussey (1991) propose a related discretization technique that delivers an approximation to a continuous density function of a given stochastic process. Their key idea is to approximate a Markov process with a finite-state Markov chain. This discretization technique requires to specify the distribution function of the Markov process explicitly and is primarily useful for forming discrete approximations of density functions of exogenous variables. In contrast, the EDS discretization technique builds on stochastic simulation and does not require to known the distribution function. It can be applied to both exogenous and endogenous variables.

There are cases in which the EDS grid is a good choice. First, focusing on a high-probability set may not have advantages relatively to a hypercube; for example, if a vector $x \in \mathbb{R}^d$ is drawn from a multivariate uniform distribution, $x \sim [0,1]^d$, then an essentially ergodic set coincides with the hypercube $[0,1]^d$, and no saving in cost is possible. Second, in some applications, one may need to know a solution outside the high-probability set, for example, when analyzing a transition path of a developing economy with a low initial endowment. Finally, there are scenarios in which EDSs have large discrepancy as shows the worse-case analysis of Section 2. We did not observe the worst-case outcomes in our experiments. However, if the density of simulated points is highly uneven (e.g., Dirac point masses), the distribution of points in an EDS may be also uneven. If we know that we are in those cases, we may opt for other grids such as a grid on a multidimensional hypercube.

3.2 General description of the EDS algorithm

In this section, we develop a projection algorithm that uses the EDS grid for solving equilibrium problems. It is also possible to use the EDS grid in the context of dynamic-programming problems (we show an application in Section 4.3).

3.2.1 An equilibrium problem

We study an equilibrium problem in which a solution is characterized by the set of equilibrium conditions for $t = 0, 1, \ldots, \infty$,

$$E_t \left[G \left(s_t, z_t, y_t, s_{t+1}, z_{t+1}, y_{t+1} \right) \right] = 0, \tag{6}$$

$$z_{t+1} = Z \left(z_t, \epsilon_{t+1} \right), \tag{7}$$

where (s_0, z_0) is given; E_t denotes the expectations operator conditional on information available at t; $s_t \in \mathbb{R}^{d_s}$ is a vector of endogenous state variables at t; $z_t \in \mathbb{R}^{d_z}$ is a vector of exogenous (random) state variables at t; $y_t \in \mathbb{R}^{d_y}$ is a vector of non-state variables – prices, consumption, labor supply, etc. – also called non-predetermined

variables; G is a continuously differentiable vector function; $\epsilon_{t+1} \in \mathbb{R}^p$ is a vector of shocks. A solution is given by a set of equilibrium functions $s_{t+1} = S(s_t, z_t)$, and $y_t = Y(s_t, z_t)$ that satisfy (6), (7) in the relevant area of the state space. In terms of notations of Section 2.1, we have $\varphi = (S, Y)$, $x_t = (s_t, z_t)$ and $d = d_s + d_z$. The solution (S, Y) is assumed to satisfy the assumptions of Section 2.1.

3.2.2 A projection algorithm based on the EDS grid

Our construction of the EDS grid in Section 2.2 is based on the assumption that the stochastic process (1) for the state variables is known. However, the law of motion for endogenous state variables is unknown before the model is solved: it is precisely our goal to approximate this law of motion numerically. We therefore proceed iteratively: guess a solution, simulate the model, construct an EDS grid, solve the model on that grid using a projection method, and iterate on these steps until the grid converges. Below, we elaborate a description of this procedure for the equilibrium problem (6), (7).

(EDS algorithm): A projection algorithm for equilibrium problems.
Step 0. Initialization.
a. Choose (s_0, z_0) and simulation length, T.
b. Draw $\{\epsilon_{t+1}\}_{t=0,...,T-1}$. Compute and fix $\{z_{t+1}\}_{t=0,...,T-1}$ using (7).
c. Choose approximating functions $S \approx \widehat{S}(\cdot; b^s)$ and $Y \approx \widehat{Y}(\cdot; b^y)$.
d. Make an initial guess on b^s and b^y.
e. Choose integration nodes, ϵ_j, and weights, ω_j, $j = 1, ..., J$.
Step 1. Construction of an EDS grid.
a. Use $\widehat{S}(\cdot; b^s)$ to simulate $\{s_{t+1}\}_{t=0,...,T-1}$.
b. Construct an EDS grid, $\Gamma \equiv \{s_m, z_m\}_{m=1,...,M}$.
Step 2. Computation of a solution on EDS grid using a projection method.
a. At iteration i, for $m = 1, ..., M$, compute
– $y_m \equiv \widehat{Y}(s_m, z_m; b^y)$, $s'_m \equiv \widehat{S}(s_m, z_m; b^s)$, $z'_{m,j} \equiv Z(z_m, \epsilon_j)$;
– $y'_{m,j} \equiv \widehat{Y}\left(s'_m, z'_{m,j}; b^y\right)$;
– $\sum_{j=1}^{J} \omega_j \cdot G\left(s_m, z_m, y_m, s'_m, z'_{m,j}, y'_{m,j}\right) = 0.$
b. Find b^s and b^y that solve the system in Step 2a.
Iterate on Steps 1, 2 until convergence of the EDS grid.

3.2.3 Choices related to the construction of the EDS grid

We construct the EDS grid as described in Section 2.2. We guess the equilibrium rule $\widehat{S}$, simulate the solution for T periods, construct a sample of n points by selecting each κth observation, estimate the density function, remove a fraction δ of the sample with the lowest density, and construct an EDS grid with a target number of points $\overline{M}$ using a bisection method. Below, we discuss some of the choices related to the construction of the EDS grids.

Initial guess on b^s. To insure that the EDS grid covers the right area of the state space, we need a sufficiently accurate initial guess about the equilibrium rules. Furthermore, the equilibrium rules used must lead to nonexplosive simulated series. For many problems in economics, linear solutions can be used as an initial guess; they are sufficiently accurate, numerically stable and readily available from automated perturbation software (we use Dynare solutions; see Adjemian et al., 2011).

The choices of n and T. Our construction of an EDS relies on the assumption that simulated points are sufficiently dense on the essentially ergodic set. Namely, in Proposition 2, we assume that each ball $B(x;\varepsilon)$ inside $\mathcal{A}^\eta$ contains at least one simulated point. The probability $\Pr(0)$ of having no points in a ball $B(x;\varepsilon)$ inside $\mathcal{A}^\eta$ after n draws satisfies $\Pr(0) \leq (1-p_\varepsilon)^n$ where $p_\varepsilon \equiv \int_{B(x;\varepsilon)} \eta dx \approx \lambda_d \varepsilon^d \eta$ and λ_d is the volume of a d-dimensional unit ball. (Note that on the boundary of $\mathcal{A}^\eta$ where $g = \eta$, we have $\Pr(0) = (1-p_\varepsilon)^n$). Thus, given ε and η, we must choose n and $T = n\kappa$ so that $\Pr(0)$ is sufficiently small. We use $T = 100,000$ and $\kappa = 10$, so that our sample has $n = 10,000$ points, and we choose η to remove 1% of points with the lowest density.

The choices of ε and $\overline{M}$. We need to have at least as many grid points in the EDS as the parameters b^s and b^y in $\widehat{S}$ and $\widehat{Y}$ (to identify these parameters). Conventional projection methods rely on collocation, when the number of grid points is the same as the number of the parameters to identify. The collocation is a useful technique in the context of orthogonal polynomial constructions but is not applicable to our case (because our bisection method does not guarantee that the number of grid points is exactly equal to the target number $\overline{M}$). Hence, we target slightly more points than the parameters, which also helps to increase both accuracy and numerical stability.

Convergence of the EDS grid. Under Assumptions 1 and 2, the convergence of the equilibrium rules implies the convergence of the time-series solution; see Peralta-

Alva and Santos (2005). Therefore, we are left to check that the EDS grid constructed on the simulated series also converges. Let $\Gamma' \equiv \{x_i'\}_{i=1,...,M'}$ and $\Gamma'' \equiv \{x_j''\}_{j=1,...,M''}$ be the EDS grids constructed on two different sets of simulated points. Our criteria of convergence is $\sup_{x_j'' \in \Gamma''} \inf_{x_i' \in \Gamma'} D\left(x_i', x_j''\right) < 2\varepsilon$. That is, each grid point of Γ'' has a grid point of Γ' at the distance smaller than 2ε (this value is used because it matches the maximum distance between the grid points on the essentially ergodic set; see Proposition 3).

3.2.4 Other choices in the EDS algorithm

The innovative feature of the projection algorithm of Section 3.2.2 is the EDS grid. The rest of the choices (such as a family of approximating functions, integration method, fitting method, etc.) should be made in a way that is suitable for a specific application. In making these choices, we must take into account the trade-off between accuracy, computational cost, numerical stability and programming efforts. Below, we identify several techniques that are particularly effective for high-dimensional applications (in conjunction with the EDS grid) and that are very simple to implement.

First, we approximate the unknown equilibrium rules using standard ordinary polynomials. Such polynomials are cheap to construct and can be fitted to the data using simple and reliable linear approximation methods.

Second, we rely on deterministic integration methods such as the Gauss-Hermite quadrature and monomial integration methods. Deterministic methods dominate in accuracy the Monte Carlo method by orders of magnitude in the context of the studied numerical examples; see Judd et al. (2011) for comparison results.[6] The cost of Gaussian product rules is prohibitive but monomial formulas are tractable in problems with very high dimensionality (even in models with hundreds of state variables). In particular, we use monomial formulas whose cost grows only linearly or quadratically with the number of shocks.

Finally, we solve the system of equations in Step 2 using a fixed-point iteration method, namely, we find the parameters vector in approximating functions on iteration $i+1$ as $b^{(i+1)} = \xi \widehat{b}^{(i)} + (1-\xi)\, b^{(i)}$, where $\widehat{b}^{(i)}$ and $b^{(i)}$ are the parameters vectors

[6] For example, assume that a Monte Carlo method is used to approximate an expectation of $y \sim \mathcal{N}(0, \sigma_y)$ with n random draws. The distribution of $\overline{y} = \sum_{i=1}^{n} y_i$ is $\overline{y} \sim \mathcal{N}\left(0, \frac{\sigma_y}{\sqrt{n}}\right)$. If $\sigma_y = 1\%$ and $n = 10,000$, we have approximation errors of order $\frac{\sigma_y}{\sqrt{n}} = 10^{-4}$. To bring the error to the level of 10^{-8}, which we attain using quadrature methods, we need to have $n = 10^{12}$. That is, a slow $\sqrt{n}$-rate of convergence makes it very expensive to obtain highly accurate solutions using stochastic simulation.

in the beginning and the end of iteration i and $\xi \in (0,1]$. For a fixed EDS grid, we consider the solution as being converged when the difference between the approximating functions on two consecutive iterations is smaller then a given value. The cost of fixed-point iteration does not practically increase with the dimensionality of the problem.

3.2.5 Evaluating the accuracy of solutions

Provided that the EDS algorithm succeeds in producing a candidate solution, we subject such a solution to a tight accuracy check. We specifically generate a set of points within the domain on which we want the solution to be accurate, and we compute residuals in all equilibrium conditions.

(Evaluation of accuracy): Residuals in equilibrium conditions

a. Choose a set of points $\{s_\tau, z_\tau\}_{\tau=1,...,\mathcal{T}^{\text{test}}}$ for evaluating the accuracy.

b. For $\tau = 1, ..., \mathcal{T}^{\text{test}}$, compute the size of the residuals:

$$\mathcal{R}(s_\tau, z_\tau) \equiv \sum_{j=1}^{J^{\text{test}}} \omega_j^{\text{test}} \cdot \left[G\left(s_\tau, z_\tau, y_\tau, s'_\tau, z'_{\tau,j}, y'_{\tau,j}\right)\right],$$

where $y_\tau = \widehat{Y}(s_\tau, z_\tau; b^y)$, $s'_\tau = \widehat{S}(s_\tau, z_\tau; b^s)$,

$z'_{\tau,j} = Z\left(z_\tau, \epsilon_j^{\text{test}}\right)$, $y'_{\tau,j} = \widehat{Y}\left(y_\tau, z'_{\tau,j}; b^y\right)$,

ϵ_j^{test} and ω_j^{test} are the integration nodes and weights.

c. Find a mean and/or maximum of the residuals $\mathcal{R}(s_\tau, z_\tau)$.

If the quality of a candidate solution is economically inacceptable, we modify the choices made in the EDS algorithm (i.e., simulation length, number of grid points, approximating functions, integration method) and recompute the solution. In the paper, we evaluate the accuracy on a set of simulated points. This new set of points is different from that used in the solution procedure: it is constructed under a different sequence of shocks. Other possible accuracy checks include an evaluation of residuals in the model's equations on a given set of points in the state space (Judd, 1992), and testing the orthogonality of residuals in the optimality conditions (Den Haan and Marcet, 1994); see Santos (2000) for a discussion.

3.2.6 Potential problems

An EDS grid is an effective grid for high-dimensional problems, however, the EDS algorithm shares the limitations of projection methods. In particular, the combination of computational techniques described above does not guarantee the convergence; and even if the algorithm converges, there is no guarantee that it converges to the

true solution. Below, we outline some problems that may arise in applications and discuss how these problems can be addressed.

First, stochastic simulation can produce explosive time series under high-degree polynomials. To restore the numerical stability, we may try to use other better behaved families of approximating functions or to restrict the equilibrium rules to satisfy certain growth limits (e.g., concavity, monotonicity, non-negativity).

Second, local perturbation solutions may be not a sufficiently accurate initial guess, and the resulting EDS grid may not represent correctly the essentially ergodic set. Finding a sufficiently good initial guess can be a non-trivial issue in some applications, and techniques from learning literature can be useful here; see Bertsekas and Tsitsiklis (1996) for a discussion.

Third, fixed-point iteration may fail to converge. Newton methods are more reliable in the given context but they also do not guarantee finding a solution. Global-search techniques are a useful but expensive alternative.

Finally, small residuals in the equilibrium conditions do not necessarily mean that we found the solution. For certain models, equilibrium may fail to exist; see Santos (2002) for examples. Furthermore, a candidate solution may be a minimum or just a regular point rather than a maximum, and the analysis of second-order conditions may be needed. Also, there could be multiple equilibria; see Feng et al. (2009) for related examples and their treatment.

In all our examples, the projection method based on the EDS grid was highly accurate and reliable, however, the reader must be aware of the existence of the above potential problems and must be ready to detect and to address such problems if they arise in applications.

4 Neoclassical stochastic growth model

In this section, we use the EDS approach to solve the standard neoclassical stochastic growth model. We discuss some relevant computational choices and assess the performance of the algorithm in one- and multi-agent setups.

4.1 The set up

The representative agent solves

$$\max_{\{k_{t+1},c_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \tag{8}$$

$$\text{s.t. } c_t + k_{t+1} = (1-\delta) k_t + a_t A f(k_t), \tag{9}$$

$$\ln a_{t+1} = \rho \ln a_t + \epsilon_{t+1}, \qquad \epsilon_{t+1} \sim \mathcal{N}\left(0, \sigma^2\right), \tag{10}$$

where (k_0, a_0) is given; E_t is the expectation operator conditional on information at time t; c_t, k_t and a_t are consumption, capital and productivity level, respectively; $\beta \in (0,1)$; $\delta \in (0,1]$; $A > 0$; $\rho \in (-1,1)$; $\sigma \geq 0$; u and f are the utility and production functions, respectively, both of which are strictly increasing, continuously differentiable and concave. Under our assumptions, this model has a unique solution; see, e.g., Stokey and Lucas with Prescott (1989, p. 392). In particular, under $u(c) = \ln(c)$, $\delta = 1$ and $f(k) = k^\alpha$, the model admits a closed-form solution $k_{t+1} = \alpha\beta a_t A k_t^\alpha$ (this solution was used to produce Figures 1 and 2).

4.2 An EDS algorithm iterating on the Euler equation

We describe an example of the EDS method that iterates on the Euler equation. For the model (8)–(10), the Euler equation is

$$u_1(c) = \beta E\left[u_1(c')\left(1 - \delta + a' A f_1(k')\right)\right], \tag{11}$$

where primes on variables denote next-period values, and F_i denotes a first-order derivative of a function f with respect the ith argument. Under the Euler equation approach, we must solve for equilibrium rules $c = C(k,a)$ and $k' = K(k,a)$ that satisfy (9)–(11).

In Table 1, we provide the results for the Euler equation EDS algorithm under the target number of grid points $\overline{M} = 25$ points.

The accuracy of solutions delivered by the EDS algorithm is comparable to the highest accuracy attained in the related literature. The residuals in the optimality conditions decrease with each polynomial degree by one or more orders of magnitude. For the fifth-degree polynomials, the largest unit-free residual corresponding to our least accurate solution is still less than 10^{-6} (see the experiment with a high degree of risk aversion $\gamma = 5$). Most of the cost of the EDS algorithm comes from the construction of the EDS grid (the time for constructing the EDS grids is included in

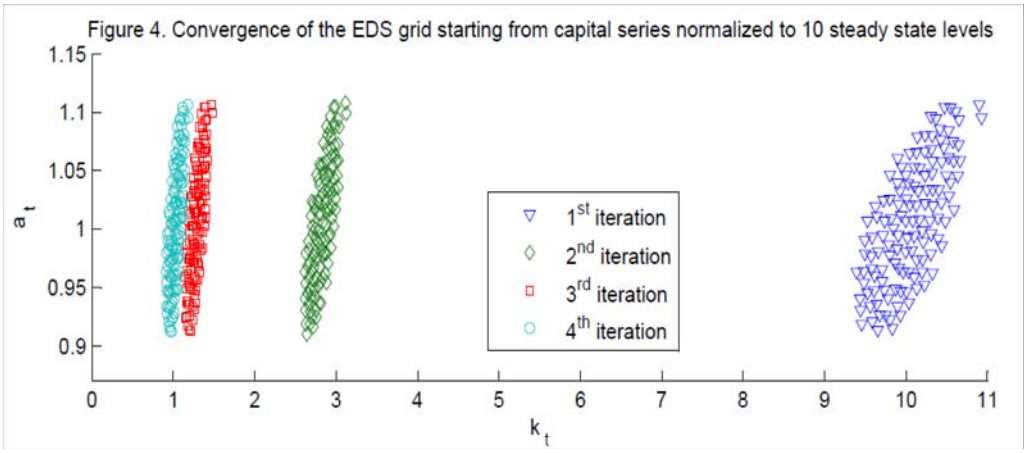
Table 1: Accuracy and speed of the Euler-equation EDSA algorithm in the one-agent model.[a]

Polynomial degree	$\gamma = 1/5$			$\gamma = 1$			$\gamma = 5$		
	$M(\varepsilon) = 21$			$M(\varepsilon) = 27$			$M(\varepsilon) = 25$		
	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
1st	-4.74	-3.81	25.5	-4.29	-3.31	24.7	-3.29	-2.35	23.6
2nd	-6.35	-5.26	1.8	-5.94	-4.87	0.8	-4.77	-3.60	0.4
3rd	-7.93	-6.50	1.9	-7.26	-6.04	0.9	-5.97	-4.47	0.4
4th	-9.37	-7.60	2.0	-8.65	-7.32	0.9	-7.05	-5.26	0.4
5th	-9.82	-8.60	14.25	-9.47	-8.24	5.5	-7.89	-6.46	2.8

[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); γ is the coefficient of risk aversion; $M(\varepsilon)$ is the realized number of points in the EDS grid (the target number of grid points is $\overline{M}$=25).

the total time for computing the polynomial solution of degree 1). Computing high-degree polynomial solutions is relatively fast (a few seconds) for a given grid. We perform sensitivity experiments in which we vary the target number of grid points and find that the results are robust to the modifications considered. We also vary the number of nodes in the Gauss-Hermite quadrature rule, and we find that even the 2-node rule leads to essentially the same accuracy levels as the 10-node rule (except the fourth and fifth-degree polynomials under which the accuracy is somewhat lower). This result is in line with the finding of Judd (1992) that in the context of the standard growth model, even few quadrature nodes lead to very accurate solutions.

Autocorrection of the EDS grid. Suppose our initial guess about the essentially ergodic set is poor. To check whether the EDS grid is autocorrecting, we perform the following experiment. We scale up the time-series solution for capital by a factor of 10, and use the resulting series for constructing the first EDS grid (thus, the capital values in this grid are spread around 10 instead of 1). We solve the model on this grid and use the solution to construct the second EDS grid. We repeat this procedure two more times. Figure 3 shows that the EDS grid converges rapidly.



We tried out various initial guesses away from the essentially ergodic set, and we observed autocorrection of the EDS grid in all the experiments performed. Furthermore, the EDS grid approach was autocorrecting in our challenging applications such as a multi-agent neoclassical growth model and a new Keynesian model with a zero lower bound on nominal interest rates. Note that the property of autocorrection of the grid is a distinctive feature of the EDS algorithm. Conventional projection methods operate on fixed domains and have no built-in mechanism for correcting their domains if the choices of their domains are inadequate.

EDS grid versus Smolyak grid. Krueger and Kubler (2004), and Malin, Krueger and Kubler (2011) develop a projection method that relies on a Smolyak sparse grid. Like conventional projection methods, Smolyak's method operates on a hypercube domain (and, hence, the size of the domain grows exponentially with the dimensionality of the state space). However, it uses a specific discretization of the hypercube domain which yields a sparse grid of carefully selected points (the number of points in the Smolyak grid grows only polynomially with the dimensionality of the state space).

We now compare the accuracy of solutions under the Smolyak and EDS grids.[7] We construct the Smolyak grid as described in Malin et al. (2011), namely, we use the

[7] Also, the Smolyak and EDS methods differ in the number of grid points (collocation versus overidentification), the polynomial family (a subset of complete Chebyshev polynomials versus a set of complete ordinary polynomials), the interpolation procedure (Smolyak interpolation versus polynomial interpolation) and the procedure for finding fixed-point coefficients (time iteration versus fixed-point iteration). These differences are important; for example, time iteration is more expensive than fixed-point iteration, the collocation is less robust and stable than overidentification.

Table 2: Accuracy and speed in the one-agent model: Smolyak grid versus EDS grid.[a]

Polynomial degree	Accuracy on a simulation				Accuracy on a hypercube			
	Smolyak grid		EDS grid		Smolyak grid		EDS grid	
	L_1	L_∞	L_1	L_∞	L_1	L_∞	L_1	L_∞
1st	-3.31	-2.94	-4.23	-3.31	-3.25	-2.54	-3.26	-2.38
2nd	-4.74	-4.17	-5.89	-4.87	-4.32	-3.80	-4.41	-3.25
3rd	-5.27	-5.13	-7.19	-6.16	-5.39	-4.78	-5.44	-4.11

[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); γ is the coefficient of risk aversion; $M(\varepsilon)$ is the realized number of points in the EDS grid (the target number of grid points is $\overline{M}$=25).

interval for capital $[0.8, 1.2]$ and the interval for productivity $\left[\exp\left(-\frac{0.8}{1-\rho}\right), \exp\left(\frac{0.8}{1-\rho}\right)\right]$. The Smolyak grid has 13 points (see Figure 3c), so we use the same target number of points in the EDS grid. With 13 grid points, we can identify the coefficients in ordinary polynomials up to degree 3. In this case, we evaluate the accuracy of solutions not only on a stochastic simulation but also on a set of 100×100 points which are uniformly spaced on the same domain as the one used by the Smolyak method for finding a solution. The results are shown in Table 2.

In the test on a stochastic simulation, the EDS grid leads to considerably more accurate solutions than the Smolyak grid. This is because under the EDS grid, we fit a polynomial directly in the essentially ergodic set, while under the Smolyak grid, we fit a polynomial in a larger rectangular domain and face a trade-off between the fit inside and outside the ergodic set. However, in the test on the rectangular domain, however, the Smolyak grid produces significantly smaller maximum residuals than the EDS grid. This is because the EDS algorithm is designed to be accurate in the essentially ergodic set and its accuracy decreases more rapidly away from the essentially ergodic set than the accuracy of methods operating on larger hypercube domains. We repeated this experiment by varying the intervals for capital and productivity in the Smolyak grid, and we have the same regularities. These regularities are also observed in high-dimensional applications.[8]

[8] Kollmann et al. (2011) compare the accuracy of solutions produced by several solution methods, including the cluster-grid algorithm (CGA) introduced in the earlier version of the present paper and the Smolyak algorithm of Krueger and Kubler (2004) (see Maliar et al., 2011, and Malin et al., 2011, for implementation details of the respective methods in the context of those models). Their comparison is performed using a collection of 30 real-business cycle models with up to 10

4.3 An EDS algorithm iterating on the Bellman equation

We can also characterize a solution to the model (8)–(10) by using dynamic programming approach. We must solve for value function $V(k,a)$ that satisfies the Bellman equation,

$$V(k,a) = \max_{k',c} \{u(c) + \beta E[V(k',a')]\} \tag{12}$$

$$\text{s.t. } k' = (1-\delta)k + aAf(k) - c, \tag{13}$$

$$\ln a' = \rho \ln a + \epsilon', \qquad \epsilon' \sim \mathcal{N}\left(0, \sigma^2\right), \tag{14}$$

where $E[V(k',a')] \equiv E[V(k',a')|k,a]$ is an expectation of $V(k',a')$ conditional on state (k,a).

The results for the Bellman equation EDS algorithm are provided in Table 3.

Table 3: Accuracy and speed of Bellman equation EDSA algorithm in the one-agent model.[a]

Polynomial degree	$\gamma = 1/5$			$\gamma = 1$			$\gamma = 5$		
	$M(\varepsilon) = 21$			$M(\varepsilon) = 27$			$M(\varepsilon) = 25$		
	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
1st	-	-	-	-	-	-	-	-	-
2nd	-4.39	-3.51	0.2	-3.98	-3.11	0.5	-2.69	-2.12	1.3
3rd	-5.70	-4.52	0.2	-5.15	-4.17	0.4	-3.80	-2.75	0.7
4th	-6.79	-5.49	0.2	-6.26	-5.12	0.4	-4.64	-3.36	0.7
5th	-7.12	-5.45	0.2	-7.42	-5.93	0.4	-5.44	-3.94	1.0

[a] Notes: L_1 and L_∞ are, repectively, the average and maximum of absolute approximation errors 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); γ is the coefficient of risk aversion; $M(\varepsilon)$ is the realized number of points in the EDS grid (the target number of grid points is $\overline{M}$=25)

The EDS algorithm iterating on the Bellman equation is successful in finding polynomial solutions of degrees 2–5 (the polynomial solutions of degree 1 imply a degenerate case when the equilibrium rules are constant).

The comparison with Table 2 shows that the Bellman equation algorithm produces much larger approximation errors (of order $10^{-4} - 10^{-6}$) than the Euler equation algorithm. The accuracy of the Bellman equation method is lower because it

heterogeneous agents. Their findings are the same as ours: on a stochastic simulation and near the steady state, the CGA solutions are more accurate than the Smolyak solutions whereas the situation reverses for large deviations from the steady state.

parameterizes the value function (while the Euler equation method parameterizes the derivative of the value function) and effectively, loses one polynomial degree. Also, the Bellman equation is much faster.

4.4 EDS algorithm in problems with high dimensionality

We now explore the tractability of the EDS algorithm in problems with high dimensionality. We extend the one-agent model (8)–(10) to include multiple agents, which is a simple way to expand and to control the size of the problem. There are N agents, interpreted as countries, which differ in initial capital endowment and productivity levels. The countries' productivity levels are affected by both country-specific and worldwide shocks. We study the social planner's problem. We do not make use of the symmetric structure of the economy and approximate the planner's solution in the form of N capital equilibrium rules, each of which depends on $2N$ state variables (N capital stocks and N productivity levels). For each country, we use essentially the same computational procedure as that used in the representative-agent case. For a description of the multicountry model and details of the computational procedure, see Appendix D.

Determinants of cost in problems with high dimensionality. The cost of finding numerical solutions increases with the dimensionality of the problem for various reasons. There are more equations to solve and more equilibrium rules to approximate. The number of terms in an approximating polynomial function increases, and we need to increase the number of grid points to identify the polynomial coefficients. The number of nodes in integration formulas increases. Finally, operating with large data sets can slow down computations or can lead to a memory congestion. If a solution method relies on product-rule constructions (of grids, integration nodes, derivatives, etc.), the cost increases exponentially (curse of dimensionality) as is in the case of conventional projection methods such as a Galerkin method of Judd (1992). Below, we show that the cost of the EDS algorithm grows at a relatively moderate rate.

Accuracy and cost of solutions. We solve the model with N ranging from 2 to 200. The results about the accuracy and cost of solutions are provided in Table 4. We consider four alternative integration rules such as the Gauss-Hermite product rule with 2^N nodes, denoted by $Q\,(2)$, the monomial rule with $2N^2+1$ nodes, denoted by $M2$, the monomial rule with $2N$ nodes, denoted by $M1$, (see Judd, 1998, formulas 7.5.9–7.5.11), and the Gauss-Hermite rule with one node, denoted by $Q\,(1)$.

Table 4. Accuracy and speed in the multicountry model depending on the integration method used. [a]

Number of Countries	Polyn. Degree	Number of Coef. [b]	$\bar{M}$	$M(\varepsilon)$	Integration Method											
					2^N nodes			$2N^2+1$ nodes			$2N$ nodes			*1 node*		
					L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
N = 2	1st	5	300	292	-4.70	-3.17	0.6	-4.70	-3.17	0.9	-4.70	-3.17	0.7	-4.68	-3.18	0.8
	2nd	15		317	-6.01	-4.06	1.8	-6.01	-4.06	2.5	-6.01	-4.06	1.9	-5.74	-4.04	1.4
N = 4	1st	9	300	291	-4.73	-3.12	1.1	-4.73	-3.12	2.0	-4.73	-3.12	0.8	-4.71	-3.13	0.7
	2nd	45		300	-6.03	-4.19	4.8	-6.03	-4.19	8.6	-6.03	-4.19	3.4	-5.58	-4.34	1.5
N = 6	1st	13	300	290	-4.73	-3.05	3.7	-4.73	-3.05	3.9	-4.73	-3.05	1.1	-4.71	-3.06	0.7
	2nd	91		300	-6.00	-4.15	21	-6.00	-4.15	24	-6.00	-4.15	5.0	-5.53	-4.27	1.6
N = 8	1st	17	300	283	-4.74	-3.02	16	-4.74	-3.02	7.8	-4.74	-3.02	1.9	-4.72	-3.03	0.7
	2nd	153		322	-5,99	-4,24	159	-5,99	-4,24	63	-5,99	-4,24	7,0	-5,49	-4,23	1,8
N = 10	1st	21	400	398	-	-	-	-4,76	-3,05	18	-4,76	-3,05	3,0	-4,74	-3,06	1,0
	2nd	231		407	-	-	-	-5,94	-4,30	208	-5,94	-4,30	16	-5,53	-4,18	3,2
N = 12	1st	25	400	398	-	-	-	-4,76	-3,06	23	-4,76	-3,06	3,9	-4,75	-3,07	1,1
	2nd	325		405	-	-	-	-5,84	-4,27	1150	-5,84	-4,27	22	-5,52	-4,22	3,8
N = 16	1st	33	1000	1006	-	-	-	-	-	-	-4,78	-3,05	14	-4,77	-3,06	2,9
	2nd	561		973	-	-	-	-	-	-	-6,03	-4,34	113	-5,51	-4,35	18
N = 20	1st	41	1000	1006	-	-	-	-	-	-	-4,76	-3,05	21	-4,77	-3,06	3,1
	2nd	861		987	-	-	-	-	-	-	-5,88	-4,14	282	-5,50	-4,14	32
N = 30	1st	61	4000	4000	-	-	-	-	-	-	-	-	-	-4,79	-3,11	59
	2nd	1891		4030	-	-	-	-	-	-	-	-	-	-5,50	-4,16	605
N = 40	1st	81	4000	4000	-	-	-	-	-	-	-	-	-	-4.79	-3.09	89
	2nd	3321		3997	-	-	-	-	-	-	-	-	-	-5.48	-4.13	1463
N=100	1st	201	1000	1005	-	-	-	-	-	-	-	-	-	-4.74	-2.95	36
N=200	1st	401	1000	981	-	-	-	-	-	-	-	-	-	-4.66	-2.90	105

[a] L_1 and L_∞ are, respectively, the average and maximum absolute unit-free Euler equation residuals (in log10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in minutes); $\bar{M}$ *and M(ε)* are the target and realized numbers of points in the EDS grid, respectively; and 2^N, $2N^2+1$, $2N$ and *1* are the Gauss-Hermite product rule with 2 nodes for each shock, quadratic and linear monomial rules and 1-node rule, respectively;

[b] In the equilibrium rule of each country.

The accuracy of solutions here is similar to that we have for the one-agent model. For the polynomial approximations of degrees 1 and 2, the residuals are typically smaller than 0.1% and 0.01%, respectively. A specific integration method used plays only a minor role in the accuracy of solutions. For the polynomial approximation of degree 1, all the integration methods considered lead to virtually the same accuracy. For the polynomial approximation of degree 2, $Q(2)$, $M2$ and $M1$ lead to the residuals which are identical up to the fourth digit, while $Q(1)$ yields the residuals which are $5-10\%$ larger. These regularities are robust to variations in the model's parameters such as the volatility and persistence of shocks and the degrees of risk aversion (see Table 8 in Judd, Maliar and Maliar, 2010, a working-paper version of the present paper).

The running time ranges from 36 seconds to 24 hours depending on the number of countries, the polynomial degree and the integration technique used. In particular, the EDS algorithm is able to compute quadratic solutions to the models with up to 40 countries and linear solutions to the models with up to 200 countries when using inexpensive (monomial and one-node quadrature) integration rules. Thus, the EDS algorithm is tractable in much larger problems than those studied in the related literature. A proper coordination between the choices of approximating function and integration technique is critical in problems with high dimensionality. An example of such a coordination is a combination of a flexible second-degree polynomial with a cheap one-node Gauss-Hermite quadrature rule (as opposed to an inefficient combination of a rigid first-degree polynomial with expensive product integration formulas).

5 New Keynesian model with the ZLB

In this section, we use the EDS algorithm to solve a stylized new Keynesian model with Calvo-type price frictions and a Taylor (1993) rule.[9] Our setup builds on the models considered in Christiano, Eichenbaum and Evans (2005), Smets and Wouters (2003, 2007), Del Negro, Schorfheide, Smets and Wouters (2007). This literature estimates their models using the data on actual economies, while we use their parameter estimates and compute solutions numerically. We solve two versions of the model, one in which we allow for negative nominal interest rates and the other in which we impose a zero lower bound (ZLB) on nominal interest rates.[10] The studied

[9] In an earlier version of the present paper, Judd et al. (2011) use a cluster-grid algorithm (CGA) to solve a new Keynesian model which is identical to the one studied here except of parameterization.

[10] For the neoclassical growth model studied in Section 4, it would be also intersting to explore the case with occasionally binding borrowing constraints. Christiano and Fisher (2000) show how

model has eight state variables and is large-scale in the sense that it is expensive or even intractable under conventional global solution methods that rely on product rules.

The literature that finds numerical solutions to new Keynesian models typically relies on local perturbation solution methods or applies expensive global solution methods to low-dimensional problems. As for perturbation, most papers compute linear approximations, and some papers compute quadratic approximations (e.g., Kollmann, 2002, and Schmitt-Grohé and Uribe, 2007) or cubic approximations (e.g., Rudebusch and Swanson, 2008). Few papers use global solution methods; see, e.g., Adam and Billi (2006), Anderson, Kim and Yun (2010), and Adjemian and Juillard (2011). The above papers have at most 4 state variables and employ simplifying assumptions.[11] A recent paper of Fernández-Villaverde, Gordon, Guerrón-Quintana, and Rubio-Ramírez (2012) uses a global solution method (namely, Smolyak's method) to study the predictions of a model similar to ours, and it provides an extensive analysis of the economic significance of the ZLB.

5.1 The set up

The economy is populated by households, final-good firms, intermediate-good firms, monetary authority and government; see Galí (2008, Chapter 3) for a detailed description of the baseline new Keynesian model.

Households. The representative household solves

$$\max_{\{C_t,L_t,B_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma}-1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta}-1}{1+\vartheta}\right] \tag{15}$$

$$\text{s.t. } P_t C_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} + T_t = B_{t-1} + W_t L_t + \Pi_t, \tag{16}$$

where $\left(B_0, \eta_{u,0}, \eta_{L,0}, \eta_{B,0}\right)$ is given; C_t, L_t, and B_t are consumption, labor and nominal bond holdings, respectively; P_t, W_t and R_t are the commodity price, nominal wage and (gross) nominal interest rate, respectively; $\eta_{u,t}$ and $\eta_{L,t}$ are exogenous preference shocks to the overall momentary utility and disutility of labor, respectively; $\eta_{B,t}$ is an exogenous premium in the return to bonds; T_t is lump-sum taxes; Π_t is the

projection methods could be used to solve such a model.

[11] In particular, Adam and Billi (2006) linearize all the first-order conditions except for the nonnegativity constraint for nominal interest rates, and Adjemian and Juillard (2011) assume perfect foresight to implement an extended path method of Fair and Taylor (1984).

profit of intermediate-good firms; $\beta \in (0,1)$ is the discount factor; $\gamma > 0$ and $\vartheta > 0$ are the utility-function parameters. The processes for shocks are

$$\eta_{u,t+1} = \rho_u \eta_{u,t} + \epsilon_{u,t+1}, \qquad \epsilon_{u,t+1} \sim \mathcal{N}\left(0, \sigma_u^2\right), \tag{17}$$

$$\eta_{L,t+1} = \rho_L \eta_{L,t} + \epsilon_{L,t+1}, \qquad \epsilon_{L,t+1} \sim \mathcal{N}\left(0, \sigma_L^2\right), \tag{18}$$

$$\eta_{B,t+1} = \rho_B \eta_{B,t} + \epsilon_{B,t+1}, \qquad \epsilon_{B,t+1} \sim \mathcal{N}\left(0, \sigma_B^2\right), \tag{19}$$

where $\rho_u, \rho_L, \rho_B \in (-1,1)$, and $\sigma_u, \sigma_L, \sigma_B \geq 0$.

Final-good firms. Perfectly competitive final-good firms produce final goods using intermediate goods. A final-good firm buys $Y_t(i)$ of an intermediate good $i \in [0,1]$ at price $P_t(i)$ and sells Y_t of the final good at price P_t in a perfectly competitive market. The profit-maximization problem is

$$\max_{Y_t(i)} \quad P_t Y_t - \int_0^1 P_t(i) Y_t(i) \, di \tag{20}$$

$$\text{s.t. } Y_t = \left(\int_0^1 Y_t(i)^{\frac{\varepsilon-1}{\varepsilon}} \, di \right)^{\frac{\varepsilon}{\varepsilon-1}}, \tag{21}$$

where (21) is a Dixit-Stiglitz aggregator function with $\varepsilon \geq 1$.

Intermediate-good firms. Monopolistic intermediate-good firms produce intermediate goods using labor and are subject to sticky prices. The firm i produces the intermediate good i. To choose labor in each period t, the firm i minimizes the nominal total cost, TC (net of government subsidy v),

$$\min_{L_t(i)} \quad \text{TC}\left(Y_t(i)\right) = (1-v) W_t L_t(i) \tag{22}$$

$$\text{s.t. } Y_t(i) = \exp\left(\eta_{a,t}\right) L_t(i), \tag{23}$$

$$\eta_{a,t+1} = \rho_a \eta_{a,t} + \epsilon_{a,t+1}, \qquad \epsilon_{a,t+1} \sim \mathcal{N}\left(0, \sigma_a^2\right), \tag{24}$$

where $L_t(i)$ is the labor input; $\exp\left(\eta_{a,t}\right)$ is the productivity level; $\rho_a \in (-1,1)$, and $\sigma_a \geq 0$. The firms are subject to Calvo-type price setting: a fraction $1-\theta$ of the firms sets prices optimally, $P_t(i) = \widetilde{P}_t$, for $i \in [0,1]$, and the fraction θ is not allowed to change the price and maintains the same price as in the previous period, $P_t(i) = P_{t-1}(i)$, for $i \in [0,1]$. A reoptimizing firm $i \in [0,1]$ maximizes the current

value of profit over the time when $\widetilde{P}_t$ remains effective,

$$\max_{\widetilde{P}_t} \quad \sum_{j=0}^{\infty} \beta^j \theta^j E_t \left\{ \Lambda_{t+j} \left[\widetilde{P}_t Y_{t+j}(i) - P_{t+j} \mathrm{mc}_{t+j} Y_{t+j}(i) \right] \right\} \tag{25}$$

$$\text{s.t. } Y_t(i) = Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon}, \tag{26}$$

where (26) is the demand for an intermediate good i (follows from the first-order condition of (20), (21)); Λ_{t+j} is the Lagrange multiplier on the household's budget constraint (16); mc_{t+j} is the real marginal cost of output at time $t+j$ (which is identical across the firms).

Government. Government finances a stochastic stream of public consumption by levying lump-sum taxes and by issuing nominal debt. The government budget constraint is

$$T_t + \frac{B_t}{\exp(\eta_{B,t}) R_t} = P_t \frac{\overline{G} Y_t}{\exp(\eta_{G,t})} + B_{t-1} + v W_t L_t, \tag{27}$$

where $\overline{G}$ is the steady-state share of government spending in output; vW_tL_t is the subsidy to the intermediate-good firms; $\eta_{G,t}$ is a government-spending shock,

$$\eta_{G,t+1} = \rho_G \eta_{G,t} + \epsilon_{G,t+1}, \qquad \epsilon_{G,t+1} \sim \mathcal{N}(0, \sigma_G^2), \tag{28}$$

where $\rho_G \in (-1, 1)$ and $\sigma_G \geq 0$.

Monetary authority. The monetary authority follows a Taylor rule. When the ZLB is imposed on the net interest rate, this rule is $R_t = \max\{1, \Phi_t\}$ with Φ_t being defined as

$$\Phi_t \equiv R_* \left(\frac{R_{t-1}}{R_*} \right)^{\mu} \left[\left(\frac{\pi_t}{\pi_*} \right)^{\phi_\pi} \left(\frac{Y_t}{Y_{N,t}} \right)^{\phi_y} \right]^{1-\mu} \exp(\eta_{R,t}), \tag{29}$$

where R_t and R_* are the gross nominal interest rate at t and its long-run value, respectively; π_* is the target inflation; $Y_{N,t}$ is the natural level of output; and $\eta_{R,t}$ is a monetary shock,

$$\eta_{R,t+1} = \rho_R \eta_{R,t} + \epsilon_{R,t+1}, \qquad \epsilon_{R,t+1} \sim \mathcal{N}(0, \sigma_R^2), \tag{30}$$

where $\rho_R \in (-1, 1)$ and $\sigma_R \geq 0$. . When the ZLB is not imposed, the Taylor rule is $R_t = \Phi_t$.

Natural level of output. The natural level of output $Y_{N,t}$ is the level of output in an otherwise identical economy but without distortions. It is a solution to the following planner's problem

$$\max_{\{C_t,L_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma}-1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta}-1}{1+\vartheta}\right] \tag{31}$$

$$\text{s.t. } C_t = \exp\left(\eta_{a,t}\right) L_t - G_t, \tag{32}$$

where $G_t \equiv \frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)}$ is given, and $\eta_{u,t+1}$, $\eta_{L,t+1}$, $\eta_{a,t+1}$, and $\eta_{G,t}$ follow the processes (17), (18), (24), and (28), respectively. The FOCs of the problem (31), (32) imply that $Y_{N,t}$ depends only on exogenous shocks,

$$Y_{N,t} = \left[\frac{\exp\left(\eta_{a,t}\right)^{1+\vartheta}}{\left[\exp\left(\eta_{G,t}\right)\right]^{-\gamma} \exp\left(\eta_{L,t}\right)}\right]^{\frac{1}{\vartheta+\gamma}}. \tag{33}$$

5.2 Summary of equilibrium conditions

We summarize the equilibrium conditions below (the derivation of the first-order conditions is provided in Appendix E):

$$S_t = \frac{\exp\left(\eta_{u,t}+\eta_{L,t}\right)}{\exp\left(\eta_{a,t}\right)} L_t^{\vartheta} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}, \tag{34}$$

$$F_t = \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon-1} F_{t+1}\right\}, \tag{35}$$

$$\frac{S_t}{F_t} = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}, \tag{36}$$

$$\Delta_t = \left[(1-\theta)\left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{\varepsilon}{\varepsilon-1}} + \theta\frac{\pi_t^{\varepsilon}}{\Delta_{t-1}}\right]^{-1}, \tag{37}$$

$$C_t^{-\gamma} = \frac{\beta \exp\left(\eta_{B,t}\right) R_t}{\exp\left(\eta_{u,t}\right)} E_t\left[\frac{C_{t+1}^{-\gamma} \exp\left(\eta_{u,t+1}\right)}{\pi_{t+1}}\right], \tag{38}$$

$$Y_t = \exp\left(\eta_{a,t}\right) L_t \Delta_t, \tag{39}$$

$$C_t = \left(1 - \frac{\overline{G}}{\exp\left(\eta_{G,t}\right)}\right) Y_t, \tag{40}$$

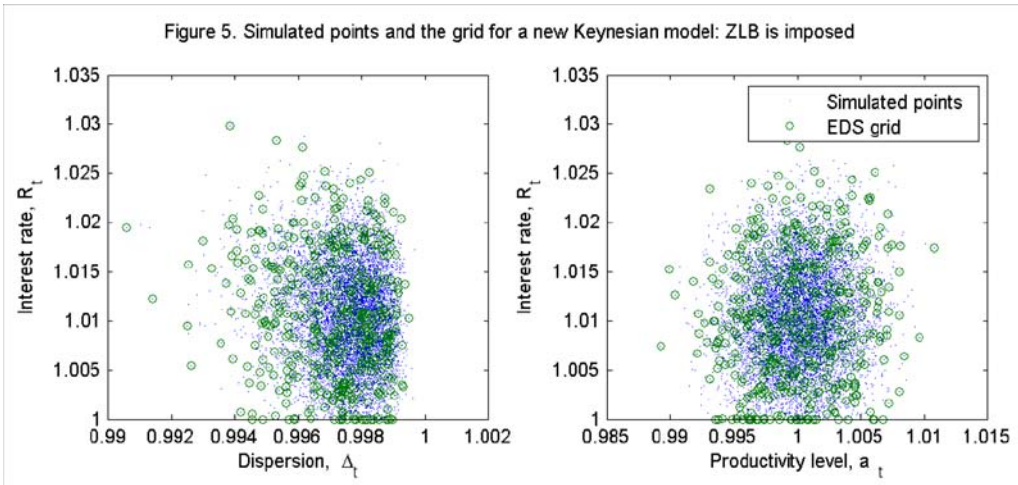
$$R_t = \max\{1,\ \Phi_t\}, \tag{41}$$

where Φ_t is given by (29); the variables S_t and F_t are introduced for a compact representation of the profit-maximization condition of the intermediate-good firm and are defined recursively; $\pi_{t+1} \equiv \frac{P_{t+1}}{P_t}$ is the gross inflation rate between t and $t+1$; Δ_t is a measure of price dispersion across firms (also referred to as efficiency distortion). The conditions (34)–(40) correspond, respectively, to (E17), (E18), (E23), (E33) and (E3) in Appendix E.

An interior equilibrium is described by 8 equilibrium conditions (34)–(41), and 6 processes for exogenous shocks, (17)–(19), (24), (30) and (28). The system of equations must be solved with respect to 8 unknowns $\{C_t, Y_t, L_t, \pi_t, \Delta_t, R_t, S_t, F_t\}$. There are 2 endogenous state variables, $\{\Delta_{t-1}, R_{t-1}\}$, and 6 exogenous state variables, $\{\eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{a,t}, \eta_{R,t}, \eta_{G,t}\}$.

5.3 Numerical analysis

Methodology. We use the estimates of Smets and Wouters (2003, 2007) and Del Negro et al. (2007) to assign values to the parameters. We approximate numerically the equilibrium rules $S_t = S(x_t)$, $F_t = F(x_t)$ and $C_t^{-\gamma} = \text{MU}(x_t)$ with $x_t = \{\Delta_{t-1}, R_{t-1}, \eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{a,t}, \eta_{R,t}, \eta_{G,t}\}$ using the Euler equations (34), (35) and (38), respectively. We solve for the other variables analytically using the remaining equilibrium conditions. We compute the polynomial solutions of degrees 2 and 3, referred to as EDS2 and EDS3, respectively. For comparison, we also compute first- and second-order perturbation solutions, referred to as PER1 and PER2, respectively (we use Dynare 4.2.1 software). When solving the model with the ZLB by the EDS algorithm, we impose the ZLB both in the solution procedure and in subsequent simulations (accuracy checks). Perturbation methods do not allow us to impose the ZLB in the solution procedure. The conventional approach in the literature is to disregard the ZLB when computing perturbation solutions and to impose the ZLB in simulations when running accuracy checks (that is, whenever R_t happens to be smaller than 1 in simulation, we set it to 1). A detailed description of the methodology of our numerical analysis is provided in Appendix E. We illustrate the EDS grid for the model with the ZLB in Figure 5 where we plot the time-series solution and the grids in two-dimensional spaces, namely, (R_t, Δ_t) and $\left(R_t, \exp\left(\eta_{a,t}\right)\right)$. We see that many points happen to be on the bound $R_t = 1$ and that the essentially ergodic set in the two figures is shaped roughly as a circle.



Accuracy and cost of solutions. Two parameters that play a special role in our analysis are the volatility of labor shocks σ_L and the target inflation rate π_*. Concerning σ_L, Del Negro et al. (2007) finds that shocks to labor must be as large as 40% to match the data, namely, they estimate the interval $\sigma_L \in [0.1821, 0.6408]$ with the average of $\sigma_L = 0.4054$. Concerning π_*, Del Negro et al. (2007) estimate the interval $\pi_* \in [1.0461, 1.0738]$ with the average of $\pi_* = 1.0598$, while Smets and Wouters (2003) use the value $\pi_* = 1$. The inflation rate affects the incidence of the ZLB: a negative net nominal interest rate is more likely to occur in a low- than a high-inflation economy.[12]

In Table 5, we show how the parameters σ_L and π_* affect the quality of numerical solutions. In the first experiment, we assume $\sigma_L = 0.1821$ (which is the lower bound of the interval estimated by Smets and Wouters, 2003), set $\pi_* = 1$ and allow for a negative net interest rate. Both the perturbation and EDS methods deliver reasonably accurate solutions. The maximum size of residuals in the equilibrium conditions is about 6% and 2% for PER1 and PER2, respectively ($10^{-1.21}$ and $10^{-1.64}$ in the table), and it is less than 1% and 0.2% for EDS2 and EDS2, respectively ($10^{-2.02}$ and $10^{-2.73}$ in the table). We also report the minimum and maximum values of R_t on a stochastic simulation, as well as a percentage number of periods in which $R_t < 1$. Here, R_t falls up to 0.9916, and the frequency of $R_t < 1$ is about 2%.

[12] Chung, Laforte, Reifschneider, and Williams (2011) provide estimates of the incidence of the ZLB in the US economy. Christiano, Eichenbaum and Rebelo (2009) study the economic significance of the ZLB in the context of a similar model. Also, Mertens and Ravn (2011) analyze the incidence of the ZLB in a model with sunspot equilibria.

Table 5. The new Keynesian model: the EDS algorithm versus perturbation algorithm. [a]

	$\pi^* = 1$ *and* $\sigma_L = 0.1821$				$\pi^* = 1.0598$ *and* $\sigma_L = 0.4054$				$\pi^* = 1$ *and* $\sigma_L = 0.1821$ with ZLB			
	PER1	PER2	EDS2	EDS3	PER1	PER2	EDS2	EDS3	PER1	PER2	EDS2	EDS3
M(ε)	-	-	496	496	-	-	496	496	-	-	494	494
Running time												
CPU	0.15		24.3	4.4	0.15		22.1	12.0	0.15		21.4	3.58
Absolute errors across optimality conditions												
L_1	-3.03	-3.77	-3.99	-4.86	-2.52	-2.90	-3.43	-4.00	-3.02	-3.72	-3.57	-3.65
L_∞	-1.21	-1.64	-2.02	-2.73	-0.59	-0.42	-1.31	-1.91	-1.21	-1.34	-1.58	-1.81
Properties of the interest rate												
R_{min}	0.9916	0.9929	0.9931	0.9927	1.0014	1.0065	1.0060	1.0060	1.0000	1.0000	1.0000	1.0000
R_{max}	1.0340	1.0364	1.0356	1.0358	1.0615	1.0694	1.0653	1.0660	1.0340	1.0364	1.0348	1.0374
Freq(R≤1).%	2.07	1.43	1.69	1.68	0	0	0	0	1.76	1.19	2.46	2.23
Difference between time series produced by the method in the given column and EDS3												
dif(*R), %*	0.17	0.09	0.05	0	0.63	0.39	0.25	0	0.33	0.34	0.34	0
dif(Δ), %	1.03	0.16	0.04	0	4.59	0.73	0.49	0	1.07	0.44	0.34	0
dif(*S), %*	5.45	1.14	0.75	0	17.94	5.83	4.15	0	5.60	9.87	5.04	0
dif(*F), %*	1.37	0.40	0.15	0	9.51	2.25	1.73	0	3.21	4.05	2.32	0
Rem dif(*C), %*	1.00	0.19	0.12	0	6.57	1.49	0.72	0	4.31	3.65	2.26	0
Rem dif(*Y), %*	1.00	0.19	0.12	0	6.57	1.48	0.72	0	4.33	3.65	2.26	0
Rem dif(*L), %*	0.65	0.33	0.16	0	3.16	1.30	0.54	0	3.37	3.17	2.45	0
Rem dif(π), %	0.30	0.16	0.11	0	1.05	0.79	0.60	0	1.17	1.39	0.79	0

[a] L_1 and L_∞ are, respectively, the average and maximum absolute percentage residuals (in log10 units) across all equilibrium conditions on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in minutes); *M(ε)* is the realized number of points in the EDS grid (the target number of grid points is $\bar{M}$ *=500)*; PER1 and PER2 are the 1st- and 2nd-order perturbation solutions, respectively; EDS2 and EDS3 are 2nd- and 3d-degree EDA polynomial solutions, respectively; R_{min} and R_{max} are, respectively, the minimum and maximum gross nominal interest rates across 10,000 simulated periods; *Freq(R≤1)* is a percentage number of periods in which *R≤1*; dif(*X),%* is the maximum absolute percentage difference between time series for variable *X* produced by the method in the given column and EDS3.

We design the next two experiments to separate the effect of the volatility of labor shocks σ_L and the inflation rate $\pi_* = 1$ on the quality of numerical solutions. In the second experiment, we consider a higher volatility of labor $\sigma_L = 0.4054$, and we set $\pi_* = 1.0598$, which is sufficient to preclude net nominal interest rates from being negative. The performance of the perturbation methods worsens dramatically. The residuals in the equilibrium conditions for the PER1 solution are as large as 25% ($10^{-0.59}$), and they are even larger for the PER2 solution, namely, 38% ($10^{-0.42}$). Thus, increasing the order of perturbation does not help increase the quality of approximation. The accuracy of the EDS solutions also decreases but less dramatically: the corresponding residuals for the EDS2 and EDS3 methods are less than 5% ($10^{-1.31}$) and 1.2% ($10^{-1.91}$), respectively. For the EDS method, high-degree polynomials do help increase the quality of approximation.

Finally, in the third experiment, we concentrate on the effect of the ZLB on equilibrium by setting $\pi_* = 1$ and by imposing the restriction $R_t \geq 1$ under the low-volatility $\sigma_L = 0.1821$ of labor shocks assumed in the first experiment. Again, we observe that the accuracy of the perturbation solutions decreases more than the accuracy of the global EDS solutions. In particular, the maximum residual for the PER2 solution is about 5%, while the corresponding residuals for the EDS2 and EDS3 solutions are less than 2.7% ($10^{-1.58}$) and 1.6% ($10^{-1.81}$), respectively.

To appreciate how much the equilibrium quantities differ across the methods, we report the maximum percentage differences between variables produced by EDS3 and those produced by the other methods on a simulation of 10,000 observations. The regularities are similar to those we observed for the residuals. The difference between the series produced by PER1 and EDS3 can be as large as 17%; the difference between the series produced by PER2 and EDS3 depends on the model: it is about 1% when the ZLB is not imposed in the model with $\sigma_L = 0.1821$ but can reach 10% when the ZLB is imposed; and finally, the difference between the series produced by EDS2 and EDS3 is smaller in all cases (5.04% at most for the model in which the ZLB is imposed). Generally, the supplementary variables S_t and F_t differ more across methods than such economically relevant variables as Y_t, L_t and C_t.

Economic importance of the ZLB. Figures 6a and 6b plot fragments of a stochastic simulation when the ZLB is not imposed and imposed, respectively, for the model parameterized by $\sigma_L = 0.1821$ and $\pi_* = 1$. When the ZLB is not imposed, both the perturbation and EDS methods predict 5 periods of negative (net) interest rates (see periods 4, 6-9 in Figure 6a). When the ZLB is imposed, The EDS method, EDS2 and EDS3, predict a zero interest rate in those 5 periods, while the perturbation methods, PER1 and PER2, predict a zero interest rate just in 3 periods (see

periods 4, 6 and 7 in Figure 6b).

Figure 6a. A time-series solution to a new Keynesian model: ZLB is not imposed

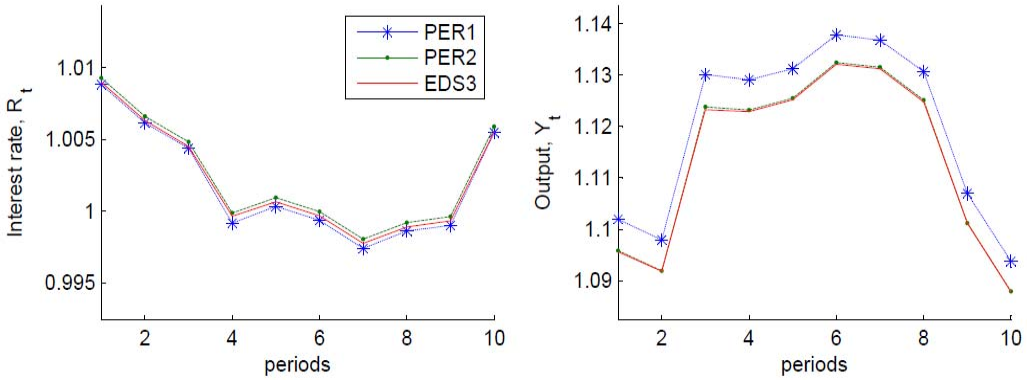
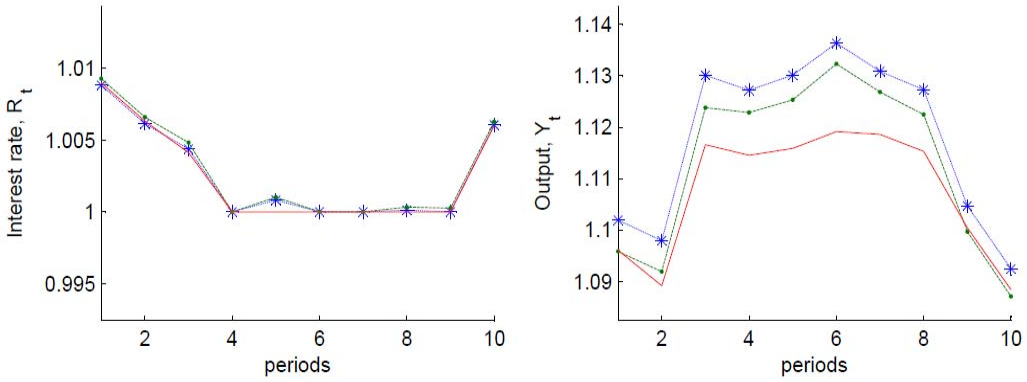


Figure 6b. A time-series solution to a new Keynesian model: ZLB is imposed



The way we deal with the ZLB in the perturbation solution misleads the agents about the true state of the economy. To be specific, when we chop the interest rate at zero in the simulation procedure, agents perceive the drop in the interest rate as being small and respond by an immediate recovery. In contrast, under the EDS algorithm, agents accurately perceive the drop in the interest rate as being large and respond by 5 periods of a zero net interest rate (which correspond to 5 periods of negative net interest rates predicted in the case when the ZLB is not imposed). The output differences between PER2 and EDS3 are relatively small when the ZLB is not imposed but they become quantitatively important when the ZLB is imposed and can be as large as 2%.

Lessons. The studied new Keynesian model is a challenging application for any numerical method. First, the dimensionality of the state space is large; second, the volatility of exogenous variables is high; and finally, there is a kink in the equilibrium rules due to the ZLB. We choose this application in order to subject the EDS method to a tight test that makes it possible to see its limitations.

Our results indicate that the EDS method is able to confront the above challenges. First, the running time for the EDS method ranges from 4 to 25 minutes; the EDS method would be tractable in much larger applications, as our results for the multi-country model suggest. Second, the EDS method produces very accurate solutions if the volatility of shocks is not excessively high, and its accuracy can be increased using polynomial functions of higher degrees, unlike the accuracy of the perturbation methods. Finally, in the presence of the ZLB, the perturbation and EDS methods may produce qualitatively different results. The accuracy of both the EDS and perturbation methods is limited by their use of polynomials, which are not well suitable for approximating equilibrium rules with kinks. The EDS algorithm may do better in terms of accuracy if we use functional families that can accommodate kinks. However, this lies beyond the scope of the present paper.

6 Conclusion

We introduce a projection algorithm that operates on a high-probability area of the ergodic set of an economic model. The EDS algorithm is tractable in problems with much higher dimensionality than those studied in the related literature. In particular, we are able to compute accurate quadratic solutions to a multicountry growth model with up to 80 state variables. Furthermore, we are able to compute an accurate global solution to a new Keynesian model. This model is of particular interest to the literature as it is used by governments and financial institutions all over the world for the policy analysis. We find that perturbation methods are not reliable in the context of new Keynesian models, and we show examples where perturbation and global solution methods produce qualitatively different predictions. We emphasize that all the numerical results in the paper are obtained using a standard desktop computer and serial MATLAB software. The speed of the EDS algorithm can be increased by far using more powerful hardware and software, as well as parallelization techniques.

References

[1] Adam, K. and R. Billi, (2006). Optimal monetary policy under commitment with a zero bound on nominal interest rates. Journal of Money, Credit, and Banking 38 (7), 1877-1905.

[2] Adjemian, S., H. Bastani, M. Juillard, F. Mihoubi, G. Perendia, M. Ratto and S. Villemot, (2011). Dynare: Reference Manual, Version 4. Dynare Working Papers 1, CEPREMAP.

[3] Adjemian, S. and M. Juillard, (2011). Accuracy of the extended path simulation method in a new Keynesian model with zero lower bound on the nominal interest rate. Manuscript.

[4] Anderson, G., J. Kim and T. Yun, (2010). Using a projection method to analyze inflation bias in a micro-founded model. Journal of Economic Dynamics and Control 34 (9), 1572-1581.

[5] 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.

[6] Baryshnikov, Yu., Eichelbacker, P., Schreiber, T., and J.E. Yukich, (2008). Moderate deviations for some point measures in geometric probability. Annales de l'Institut Henri Poincaré - Probabilités et Statistiques 44, 422-446.

[7] Bertsekas, D. and J. Tsitsiklis (1996). Neuro-Dynamic Programming. Optimization and Neural Computation Series. Athena Scientific: Belmont, Massachusetts.

[8] Christiano, L., M. Eichenbaum and C. Evans, (2005). Nominal rigidities and the dynamic effects of a shock to monetary policy. Journal of Political Economy 113/1, 1-45.

[9] Christiano, L., M. Eichenbaum, and S. Rebelo, (2011). When is the government spending multiplier large? Journal of Political Economy 119(1), 78-121.

[10] Christiano, L. and D. Fisher, (2000). Algorithms for solving dynamic models with occasionally binding constraints. Journal of Economic Dynamics and Control 24, 1179-1232.

[11] Chung, H., J.-P. Laforte, D. Reifschneider and J. Williams, (2011). Have we underestimated the probability of hitting the zero lower bound? Federal Reserve Bank of San Francisco. Working paper 2011-01.

[12] Collard, F., and M. Juillard, (2001). Accuracy of stochastic perturbation methods: the case of asset pricing models, Journal of Economic Dynamics and Control, 25, 979-999.

[13] Del Negro, M., F. Schorfheide, F. Smets and R. Wouters, (2007). On the fit of new Keynesian models. Journal of Business and Economic Statistics 25 (2), 123-143.

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

[15] Den Haan, W., and A. Marcet (1994). Accuracy in Simulations. Review of Economic Studies 61, 3–18.

[16] Den Haan, W., (2010), Comparison of solutions to the incomplete markets model with aggregate uncertainty. Journal of Economic Dynamics and Control 34, 4–27.

[17] Fair, R. and J. Taylor, (1983). Solution and maximum likelihood estimation of dynamic nonlinear rational expectation models. Econometrica 51, 1169-1185.

[18] 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.

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

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

[21] Fudenberg D. ad D. Levine, (1993). Self-confirming equilibrium. Econometrica 61, 523-545.

[22] Galí, J., (2008). Monetary Policy, Inflation and the Business Cycles: An Introduction to the New Keynesian Framework. Princeton University Press: Princeton, New Jersey.

[23] Gaspar, J. and K. Judd, (1997). Solving large-scale rational-expectations models. Macroeconomic Dynamics 1, 45-75.

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

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

[26] Judd, K. and S. Guu, (1993). Perturbation solution methods for economic growth models, in: H. Varian, (Eds.), Economic and Financial Modeling with Mathematica, Springer Verlag, pp. 80-103.

[27] Judd, K., L. Maliar and S. Maliar, (2010). A cluster-grid projection method: solving problems with high dimensionality. NBER working paper 15965.

[28] Judd, K., L. Maliar and S. Maliar, (2011). A cluster-grid projection algorithm: solving problems with high dimensionality. Manuscript available at http://stanford.edu/~maliars.

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

[30] 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.

[31] Kiefer, J., (1961). On large deviations of the empiric D.F. of vector change variables and a law of the iterated logarithm. Pacific Journal of Mathematics 11, 649-660.

[32] Kollmann, R., (2002). Monetary policy rules in the open economy: effects on welfare and business cycles. Journal of Monetary Economics 49, 989-1015.

[33] 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.

[34] Kollmann, R., S. Maliar, B. 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.

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

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

[37] 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.

[38] 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.

[39] Marcet, A. (1988), Solving non-linear models by parameterizing expectations. Unpublished manuscript, Carnegie Mellon University, Graduate School of Industrial Administration.

[40] 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.

[41] Marcet, A. and T. Sargent, (1989). Convergence of least-squares learning in environments with hidden state variables and private information. Journal of Political Economy 97, 1306-1322.

[42] Marimon, R. and A. Scott, (1999). Computational Methods for Study of Dynamic Economies, Oxford University Press, New York.

[43] Mertens, K. and M. Ravn, (2011). Credit channels in a liquidity trap. CEPR discussion paper 8322.

[44] Niederreiter, H., (1992). Random Number Generation and Quasi-Monte Carlo Methods. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

[45] Pakes, A. and P. McGuire, (2001). Stochastic algorithms, symmetric Markov perfect equilibria, and the 'curse' of dimensionality. Econometrica 69, 1261-1281.

[46] Peralta-Alva, A. and M. Santos, (2005). Accuracy of simulations for stochastic dynamic models. Econometrica 73, 1939-1976.

[47] Pichler, P., (2010). Solving the multi-country real business cycle model using a monomial rule Galerkin method. Journal of Economic Dynamics and Control 35, 240-251.

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

[49] Rényi, A., (1958). On a one-dimensional space-filling problem. Publication of the Mathematical Institute of the Hungarian Academy of Sciences 3, 109-127.

[50] Rudebusch, G. and E. Swanson, (2008). Examining the bond premium puzzle with a DSGE model. Journal of Monetary Economics 55, S111-S126.

[51] Rust, J., (1996). Numerical dynamic programming in economics, in: H. Amman, D. Kendrick and J. Rust (Eds.), Handbook of Computational Economics, Amsterdam: Elsevier Science, 619-722.

[52] Rust, J., (1997). Using randomization to break the curse of dimensionality. Econometrica 65, 487-516.

[53] 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.

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

[55] Santos, M., (2002). On non-existance of Markov equilibria in competitive market economies. Journal of Economic Theory 105, 73-98.

[56] Schmitt-Grohé S. and M. Uribe, (2007). Optimal simple and implementable monetary fiscal rules. Journal of Monetary Economics 54, 1702-1725.

[57] Scott, D. and S. Sain, (2005). Multidimensional density estimation, in: C. Rao, E. Wegman and J. Solka (Eds.), Handbook of Statistics, vol. 24, Amsterdam: Elsevier B. V, pp. 229-261.

[58] Smets, F. and R. Wouters, (2003). An estimated dynamic stochastic general equilibrium model of the Euro area. Journal of the European Economic Association 1(5), 1123-1175.

[59] Smets, F. and R. Wouters, (2007). Shocks and frictions in US business cycles: a Bayesian DSGE approach. American Economic Review 97 (3), 586-606.

[60] Smith, A., (1993). Estimating nonlinear time-series models using simulated vector autoregressions. Journal of Applied Econometrics 8, S63-S84.

[61] Stachursky, J., (2009). Economic Dynamics: Theory and Computations. Cambridge, MA: MIT Press.

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

[63] Tauchen, G. and R. Hussey, (1991). Quadrature-based methods for obtaining approximate solutions to nonlinear asset pricing models. Econometrica 59, 371-396.

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

[65] Taylor, J., (1993). Discretion versus policy rules in practice. Carnegie-Rochester Conference Series on Public Policy 39, 195-214.

[66] Temlyakov, V., (2011). Greedy approximation. Cambridge University Press, Cambridge.

[67] Tsitsiklis, J., (1994). Asynchronous stochastic approximation and Q-Learning, Machine Learning 16, 185-202.

[68] Wright, B. and J. Williams, (1984). The welfare effects of the introduction of storage. Quarterly Journal of Economics 99, 169-192.

Supplement to "Merging Simulation and Projection Approaches to Solve High-Dimensional Problems": Appendices

Kenneth L. Judd
Lilia Maliar
Serguei Maliar

Appendix A. A cheaper version of the EDS technique

We show a version of the EDS technique which is cheaper than that described in Section 2.2. The idea is to invert Steps 1 and 2: we first use Algorithm P^{ε} to construct an EDS on all simulated points, and we then use Algorithm $\mathcal{A}^{\eta}$ to remove from the EDS the low-density points. To be more specific, in Step 1, we produce a set P of n points $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$ by simulation of (1) and select an EDS P^{ε} of M points, $x_1^{\varepsilon}, ..., x_M^{\varepsilon}$; and in Step 2, we estimate density function $\widehat{g}(x_i^{\varepsilon}) \approx g(x_i^{\varepsilon})$ for all $x_i^{\varepsilon} \in P^{\varepsilon}$ and remove from P^{ε} a subset of points that represents a fraction of the sample δ which has the lowest density. To control the fraction of the sample removed, we use the estimated density function $\widehat{g}$. Note that when eliminating a point $x_j^{\varepsilon} \in P^{\varepsilon}$, we remove $\frac{\widehat{g}(x_j^{\varepsilon})}{\sum_{i=1}^{M} \widehat{g}(x_i^{\varepsilon})}$ of the original sample. We therefore can proceed with eliminations of points from the EDS one by one until their cumulative mass reaches the target value δ.

An advantage of this version of the EDS technique compared to the baseline one is its lower cost. In our baseline version, the kernel algorithm computes pairwise distances between all n simulated points which has the cost of order $O(n^2)$. In the modified version, the kernel algorithm estimates the density function only in M points contained in the EDS, and thus it computes the distances from M points to n points with the cost of order $O(nM)$. If $M \ll n$, this may be far smaller than $O(n^2)$, the complexity of the baseline two-step procedure of Section 2.2.

An illustration of the cheaper version of the EDS technique. To illustrate the application of the cheaper version of the EDS technique, we use the example of

Figure 7a. Simulated points

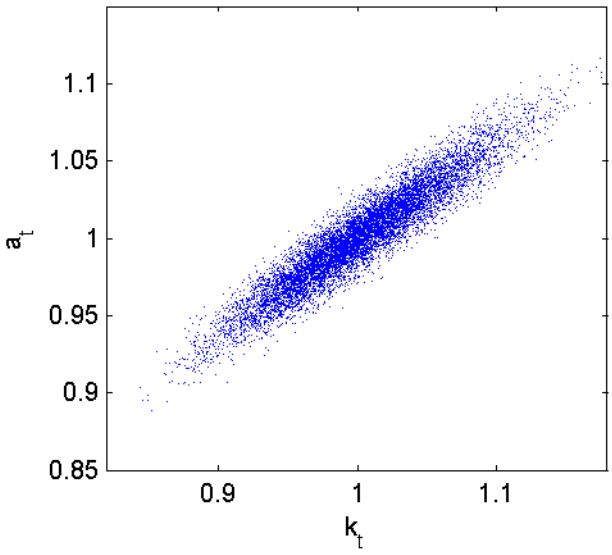


Figure 7b. Principal components (PCs)

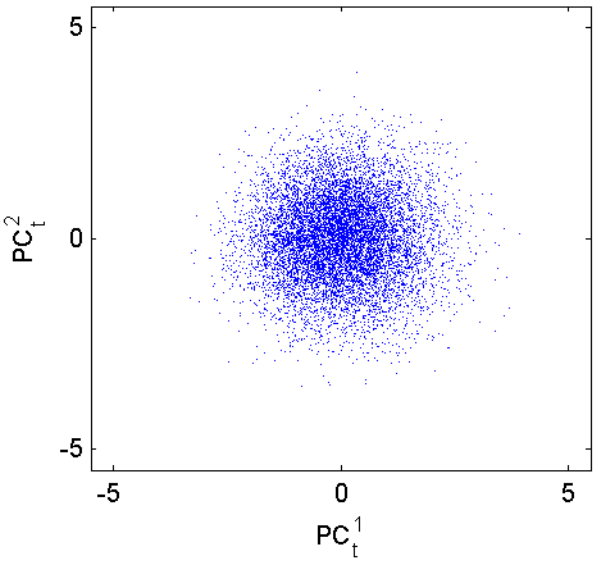


Figure 7c. EDS on PCs

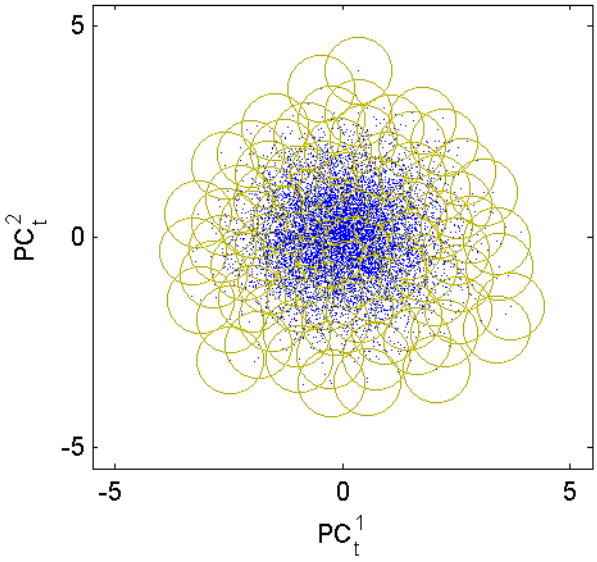


Figure 7d. Density levels on EDS

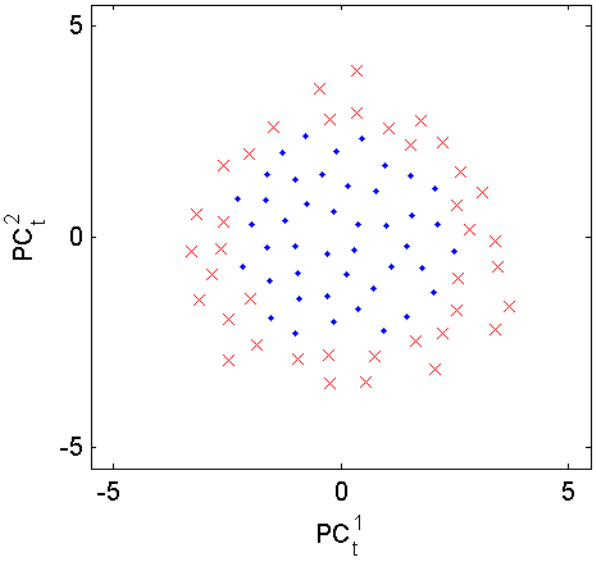


Figure 7e. EDS on PCs

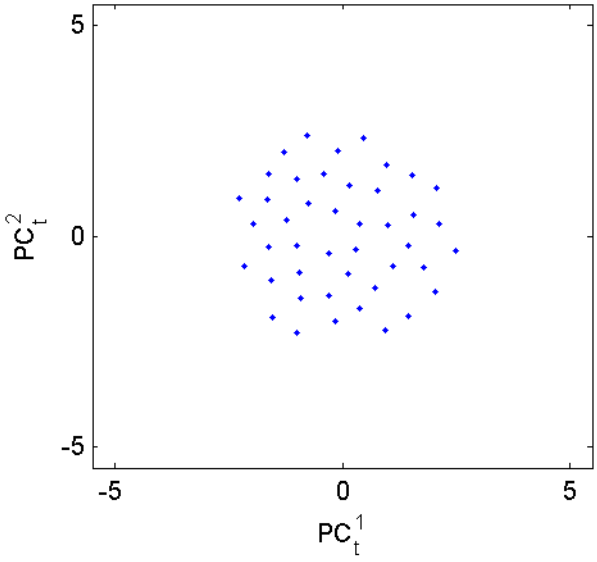
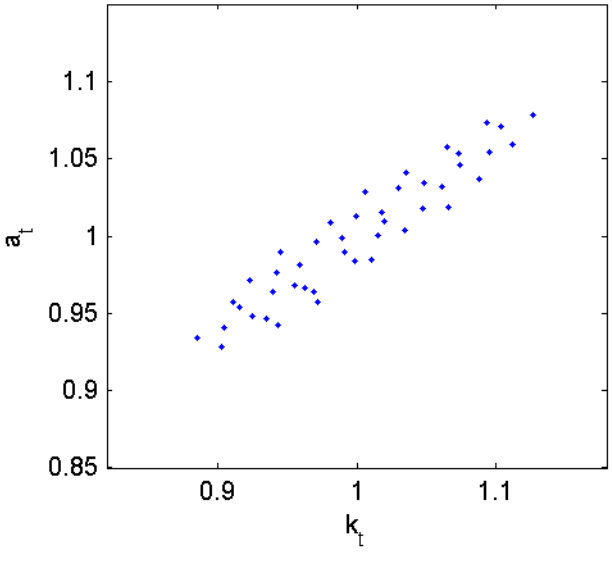


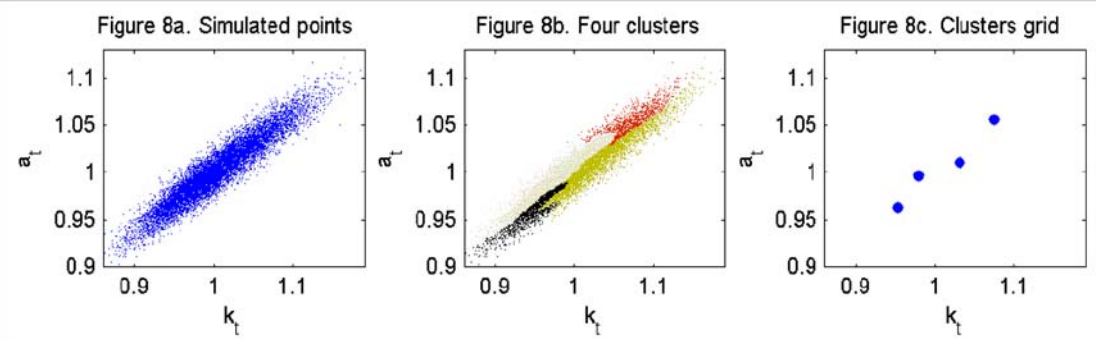
Figure 7f. EDS on original data



the neoclassical growth model with a closed-form solution studied in Section 2.2.4; see Figures 7a-7f. We first compute the normalized PCs of the original sample; see Figure 7b (this step in the same as in Figure 2b). We compute an EDS P^{ε} on the normalized PCs; see Figure 7c. We then estimate the density function in all points of P^{ε} using the kernel density algorithm. We finally removed from P^{ε} a set of points that has the lowest density function and that represents 5% of the sample. The removed points are represented with crosses in Figure 7d. The resulting EDS is shown in Figure 7e. Finally, we plot the EDS grid in the original coordinates in Figure 7f.

Appendix B. Clustering algorithms

Instead of the EDS technique, we can use methods from cluster analysis to construct a grid for our projection algorithm; see Everitt et al. (2011) for a review of clustering methods. In this case, we partition the simulated data into clusters (groups of closely-located points), and we replace each cluster with just one representative point. In Figures 8a-8c, we show an example in which we partition a set of simulated points into 4 clusters and construct 4 representative points. A representative point is the closest point to the cluster center (computed as the average of all observations in the given cluster).



In this section, we discuss two clustering methods that can be used in the context of our analysis, an agglomerative hierarchical and K-means ones. The advantage of clustering methods is that we can control the number of grid points directly (while the number of points in an EDS depends on ε). The drawbacks are that their complexity is higher (it is of order $O\left(n^3\right)$ and $O\left(n^{dM+1}\log n\right)$ for the agglomerative hierarchical and K-means algorithms, respectively), and that the properties of grids produced by clustering methods are hard to characterize analytically.

As is in the case of the EDS technique, two versions of the clustering technique can be constructed: we can first remove the low-density points and then construct representative points using clustering methods (this is parallel to the basic two-step EDS procedure of Section 2.2), or we can first construct clusters and then eliminate representative points in which the density function is the lowest (this is parallel to the cheap version of the two-step procedure described in Appendix A). Prior to an application of clustering methods, we preprocess the data by constructing the normalized PCs as we do when constructing an EDS grid in Section 2.2.

Appendix B1. Hierarchical clustering algorithm

A hierarchical agglomerative clustering algorithm which begins from individual objects (observations) and agglomerates them iteratively into larger objects – clusters.

Distance between groups of observations. As a measure of distance between two groups of observations (clusters), $A \equiv \{x_1, ..., x_I\}$ and $B \equiv \{y_1, ..., y_J\}$, we use Ward's measure of distance.[13] This measure shows how much the dispersion of observations changes when clusters A and B are merged together compared to the case when A and B are separate clusters.

Formally, we proceed as follows:

Step 1. Consider the cluster A. Compute the cluster's center $\overline{x} \equiv \left(\overline{x}^1, ..., \overline{x}^{\mathcal{L}}\right)$ as a simple average of the observations, $\overline{x}^\ell \equiv \frac{1}{I}\sum_{i=1}^{I} x_i^\ell$.

Step 2. For each $x_i \in A$, compute distance $D\left(x_i, \overline{x}\right)$ to its own cluster's center.

Step 3. Compute the dispersion of observations in cluster A as a squared sum of distances to its own center, i.e., $SSD\left(A\right) \equiv \sum_{i=1}^{I} \left[D\left(x_i, \overline{x}\right)\right]^2$.

Repeat Steps 1-3 for cluster B and for the cluster obtained by merging clusters A and B into a single cluster $A \cup B$.

Ward's measure of distance between A and B is defined as

$$\widetilde{D}\left(A, B\right) = SSD\left(A \cup B\right) - \left[SSD\left(A\right) + SSD\left(B\right)\right]. \tag{B1}$$

This measure is known to lead to spherical clusters of a similar size, see, e.g., Everitt et al. (2011, p. 79). This is in line with our goal of constructing a uniformly spaced grid that covers the essentially ergodic set. In our experiments, Ward's measure

[13] If a measure of distance between groups of observations does not fulfill the triangular inequality, it is not a distance in the conventional sense and is referred to in the literature as dissimilarity.

yielded somewhat more accurate solutions than the other measures of distance considered, such as the nearest neighbor, furthest neighbor, group average; see, e.g., Romesburg (1984) and Everitt et al. (2011) for reviews.

Steps of the agglomerative hierarchical clustering algorithm. The zero-order partition $\mathcal{P}^{(0)}$ is the set of singletons – each observation represents a cluster.

Step 0. Choose measures of distance between observations and clusters. Choose M, the number of clusters to be created.

Step 1. On iteration q, compute all pairwise distances between the clusters in partition $\mathcal{P}^{(q)}$.

Step 2. Merge a pair of clusters with the smallest distance into a new cluster. The resulting partition is $\mathcal{P}^{(q+1)}$.

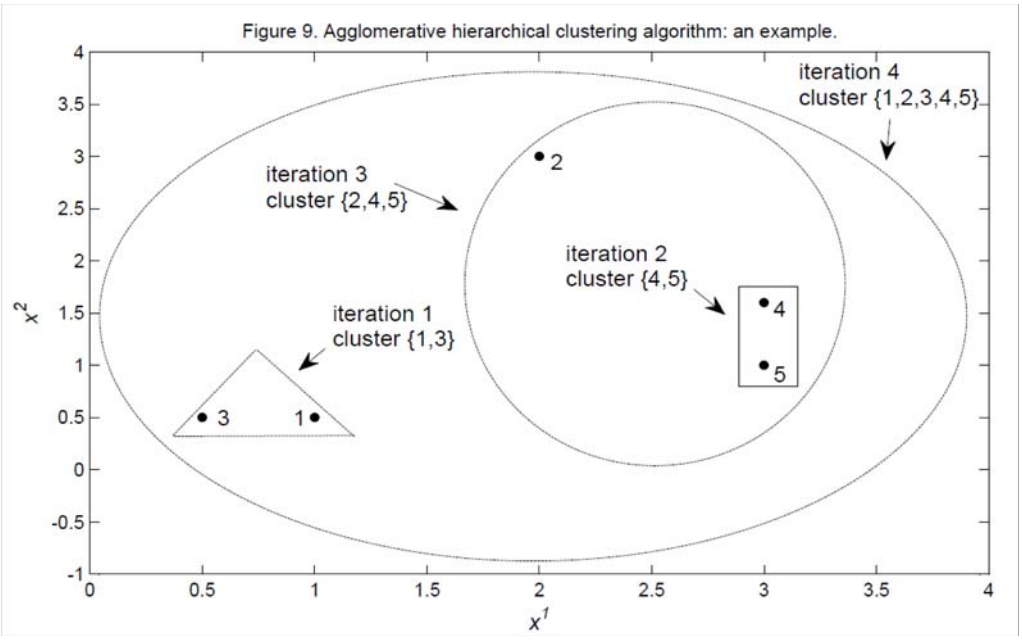
Iterate on Steps 1 and 2. Stop when the number of clusters in the partition is M. Represent each cluster with a simulated point which is closest to the cluster's center. Below we illustrate the operation of this algorithm by way of example.

A numerical example of implementing the agglomerative hierarchical clustering algorithm. We provide a numerical example that illustrates the construction of clusters under the agglomerative hierarchical algorithm. The sample data contains 5 observations for 2 variables, x^1 and x^2:

Observation	Variable x^1	Variable x^2
1	1	0.5
2	2	3
3	0.5	0.5
4	3	1.6
5	3	1

We will consider two alternative measures of distance between clusters, the nearest-neighbor (or single) and Ward's ones. Both measures lead to an identical set of clusters shown in Figure 9. On iteration 1, we merge observations 1 and 3 into a cluster $\{1, 3\}$; on iteration 2, we merge observations 4 and 5 into a cluster $\{4, 5\}$; on iteration 3, we merge observations 2 and $\{4, 5\}$ into a cluster $\{2, 4, 5\}$; and finally, on iteration 4, we merge clusters $\{1, 3\}$ and $\{2, 4, 5\}$ into one cluster that contains all observations $\{1, 2, 3, 4, 5\}$. Below, we describe computations performed by the clustering algorithm. We first consider the nearest-neighbor measure of distance which is simpler to understand (because the distance between clusters can be inferred from the distance between observations without additional computations). We then

show how to construct clusters using the Ward's distance measure, which is our preferred choice in numerical analysis.



Nearest-neighbor measure of distance. The nearest-neighbor measure of distance between the clusters A and B is the distance between the closest pair of observations $x_i \in A$ and $y_j \in B$, i.e., $\widetilde{D}(A,B) = \min\limits_{x_i \in A,\ y_j \in B} D(x_i, y_j)$. Let $D(x_i, y_j) = \left[\left(x_i^1 - y_j^1\right)^2 + \left(x_i^2 - y_j^2\right)^2\right]^{1/2} \equiv D_{ij}$ be the Euclidean distance.

Let us compute a matrix of distances between singleton clusters in which each entry ij corresponds to D_{ij},

$$S_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{pmatrix} 0 & & & & \\ 2.7 & 0 & & & \\ 0.5 & 2.9 & 0 & & \\ 2.3 & 1.7 & 2.7 & 0 & \\ 2.1 & 2.2 & 2.5 & 0.6 & 0 \end{pmatrix}$$

The smallest non-zero distance for the five observations in S_1 is $D_{13} = 0.5$. Thus, we merge observations (singleton clusters) 1 and 3 into one cluster and call the obtained cluster $\{1,3\}$. The distances for the four resulting clusters $\{1,3\}$, 2, 4, and 5, are

shown in a matrix S_2,

$$S_2 = \begin{matrix} \{1,3\} \\ 2 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & & & \\ 2.7 & 0 & & \\ 2.3 & 1.7 & 0 & \\ 2.1 & 2.2 & 0.6 & 0 \end{pmatrix}$$

where $\widetilde{D}(\{1,3\},2) = \min\{D_{12}, D_{32}\} = 2.7$, $\widetilde{D}(\{1,3\},4) = \min\{D_{14}, D_{34}\} = 2.3$, and $\widetilde{D}(\{1,3\},5) = \min\{D_{15}, D_{35}\} = 2.1$. Given that $\widetilde{D}(4,5) = D_{45} = 0.6$ is the smallest non-zero entry in S_2, we merge singleton clusters 4 and 5 into a new cluster $\{4,5\}$. The distances for three clusters $\{1,3\}$, $\{4,5\}$ and 2 are given in S_3,

$$S_3 = \begin{matrix} \{1,3\} \\ \{4,5\} \\ 2 \end{matrix} \begin{pmatrix} 0 & & \\ 2.1 & 0 & \\ 2.7 & 1.7 & 0 \end{pmatrix}$$

where $\widetilde{D}(\{1,3\},2) = \min\{D_{12}, D_{32}\} = 2.7$, $\widetilde{D}(\{4,5\},2) = \min\{D_{42}, D_{52}\} = 1.7$ and $\widetilde{D}(\{1,3\},\{4,5\}) = \min\{D_{14}, D_{15}, D_{34}, D_{35}\} = 2.1$. Hence, the smallest non-zero distance in S_3 is $\widetilde{D}(\{4,5\},2)$, so we merge clusters 2 and $\{4,5\}$ into a cluster $\{2,4,5\}$. The only two clusters left not merged are $\{1,3\}$ and $\{2,4,5\}$, so that the last step is to merge those two to obtain the cluster $\{1,2,3,4,5\}$. The procedure of constructing clusters is summarized below:

Iteration	Cluster Created	Clusters Merged		Shortest Distance
1	$\{1,3\}$	1	3	0.5
2	$\{4,5\}$	4	5	0.6
3	$\{2,4,5\}$	2	$\{4,5\}$	1.7
4	$\{1,2,3,4,5\}$	$\{1,3\}$	$\{2,4,5\}$	2.1

The algorithm starts from 5 singleton clusters, and after 4 iterations, it merges all observations into a single cluster (thus, the number of clusters existing, e.g., on iteration 2 is $5-2=3$).

Ward's measure of distance. We now construct clusters using Ward's measure of distance (B1). As an example, consider the distance between the singleton clusters 1 and 2, i.e., $\widetilde{D}(1,2)$. The center of the cluster $\{1,2\}$ is $\overline{x}_{\{1,2\}} = \left(\overline{x}^1_{\{1,2\}}, \overline{x}^2_{\{1,2\}}\right) = (1.5, 1.75)$, and $SSD(1) = SSD(2) = 0$. Thus, we have:

$$\widetilde{D}(1,2) = SSD(\{1,2\}) = (1-1.5)^2 + (2-1.5)^2 + (0.5-1.75)^2 + (3-1.75)^2 = 3.625.$$

In this manner, we obtain the following matrix of distances between singleton clusters on iteration 1

$$W_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{pmatrix} 0 & & & & \\ 3.625 & 0 & & & \\ 0.125 & 4.25 & 0 & & \\ 2.605 & 1.48 & 3.73 & 0 & \\ 2.125 & 2.5 & 3.25 & 1.8 & 0 \end{pmatrix}$$

Given that $\widetilde{D}(1,3) = 0.125$ is the smallest non-zero distance in W_1, we merge singleton clusters 1 and 3 into cluster $\{1,3\}$.

In the beginning of iteration 2, we have clusters (13), 2, 4 and 5. To illustrate the computation of distances between clusters that are not singletons, let us compute $\widetilde{D}(\{1,3\},2)$. The center of cluster $\{1,3\}$ is

$$\overline{x}_{\{1,3\}} = \left(\overline{x}^1_{\{1,3\}}, \overline{x}^2_{\{1,3\}}\right) = (0.75, 0.5),$$

and that of cluster $\{1,2,3\}$ is

$$\overline{x}_{\{1,2,3\}} = \left(\overline{x}^1_{\{1,2,3\}}, \overline{x}^2_{\{1,2,3\}}\right) = (7/6, 4/3).$$

We have

$$SSD(\{1,3\}) = (1-0.75)^2 + (0.5-0.75)^2 + (0.5-0.5)^2 + (0.5-0.5)^2 = 0.125,$$

$$\begin{aligned} SSD(\{1,2,3\}) &= (1-7/6)^2 + (2-7/6)^2 + (0.5-7/6)^2 \\ &+ (0.5-4/3)^2 + (3-4/3)^2 + (0.5-4/3)^2 = 16/3, \end{aligned}$$

and $SSD(2) = 0$. Thus, we obtain

$$\widetilde{D}(\{1,3\},2) = SSD(\{1,2,3\}) - [SSD(\{1,3\}) + SSD(2)] = 16/3 - 0.125 = 5.2083.$$

The distances obtained on iteration 2 are summarized in the matrix of distances W_2:

$$W_2 = \begin{array}{c} \{1,3\} \\ 2 \\ 4 \\ 5 \end{array} \begin{pmatrix} 0 & & & \\ 5.2083 & 0 & & \\ 4.1817 & 1.48 & 0 & \\ 3.5417 & 2.5 & 0.18 & 0 \end{pmatrix}$$

Given that $\widetilde{D}(\{4,5\}) = 0.18$ is the smallest non-zero distance in W_2, we merge singleton clusters 4 and 5 into cluster $\{4,5\}$.

On iteration 3, the matrix of distances is

$$W_3 = \begin{matrix} \{1,3\} \\ \{4,5\} \\ 2 \end{matrix} \begin{pmatrix} 0 & & \\ 5.7025 & 0 & \\ 5.2083 & 2.5933 & 0 \end{pmatrix}$$

which implies that clusters $\{4,5\}$ and 2 must be merged into $\{2,4,5\}$.

On the last iteration, $\{1,3\}$ and $\{2,4,5\}$ are merged into $\{1,2,3,4,5\}$. As we see, Ward's measure of distance leads to the same clusters as the nearest-neighbor measure of distance. Finally, in practice, it might be easier to use an equivalent representation of Ward's measure of distance in terms of the clusters' centers,

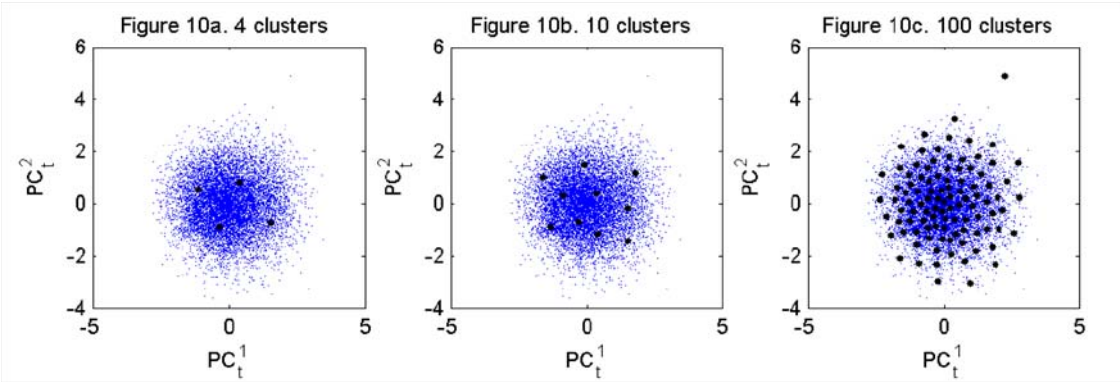
$$\widetilde{D}(A,B) = \frac{I \cdot J}{I+J} \sum_{\ell=1}^{\mathcal{L}} \left(\overline{x}^{\ell} - \overline{y}^{\ell}\right)^2,$$

where $A \equiv \{x_1, ..., x_I\}$, $B \equiv \{y_1, ..., y_J\}$, $\overline{x}^{\ell} \equiv \frac{1}{I}\sum_{i=1}^{I} x_i^{\ell}$ and $\overline{y}^{\ell} \equiv \frac{1}{J}\sum_{j=1}^{J} y_j^{\ell}$. For example, $\widetilde{D}(1,2)$ on iteration 1 can be computed as

$$\widetilde{D}(1,2) = \frac{1}{2}\left[(1-2)^2 + (0.5-3)^2\right] = 3.625,$$

where the centers of singleton clusters 1 and 2 are the observations themselves.

An illustration of the clustering techniques. In Figures 10a, 10b and 10c, we draw, respectively, 4, 10 and 100 clusters on the normalized PCs shown in Figure 2b. The constructed cluster grid is less uniform than the EDS grids: the density of points in the cluster grid depends on the density of simulated points.



6.1 Appendix B2. K-means clustering algorithm

A K-means clustering algorithm obtains a single partition of data instead of a cluster tree generated by a hierarchical algorithm. The algorithm starts with M random clusters, and then moves objects between those clusters with the goal to minimize variability within clusters and to maximize variability between clusters. The basic K-means algorithm proceeds as follows:

Step 0. Choose M, the number of clusters. Generate randomly M clusters to obtain initial partition $\mathcal{P}^{(0)}$.

Step 1. On iteration q, for each cluster $A \in \mathcal{P}^{(q)}$, determine the cluster's center $\overline{x} \equiv \left(\overline{x}^1, ..., \overline{x}^{\mathcal{L}}\right)$ as a simple average of the observations, $\overline{x}^\ell \equiv \frac{1}{I}\sum_{i=1}^{I} x_i^\ell$.

Step 2. For each x_i, compute the distance $D\left(x_i, \overline{x}\right)$ to all clusters' center.

Step 3. Assign each x_i to the nearest cluster center. Recompute the centers of the new M clusters. The resulting partition is $\mathcal{P}^{(q+1)}$.

Iterate on Steps 1–3 until convergence.

Unlike the hierarchical clustering algorithm, the K-means algorithm can give different results with each run. This is because the K-means algorithm is sensitive to initial random assignments of observations into clusters. In this respect, K-means algorithm is similar to Algorithm P^ε that can produce different EDSs depending on the order in which points are processed.

Appendix C. Neoclassical stochastic growth model

In this section, we present the two algorithms based on the EDS grid for solving the neoclassical growth model considered in Section 4, one iterating on the Euler equation and the other iterating on the Bellman equation.

An EDS algorithm iterating on the Euler equation. We parameterize K with a flexible functional form $\widehat{K}\left(\cdot; b\right)$ that depends on a coefficients vector b. Our goal is to find b that finds $\widehat{K} \approx K$ on the EDS grid given the functional form $\widehat{K}\left(\cdot; b\right)$. We compute b using fixed-point iteration (FPI). To implement FPI on $\widehat{K}$, we rewrite (11) in the form

$$k' = \beta E\left[\frac{u_1\left(c'\right)}{u_1\left(c\right)}\left(1 - \delta + a' A f_1\left(k'\right)\right)k'\right]. \tag{C2}$$

In the true solution, k' on both sides of (C2) takes the same values and thus, cancels out. In the FPI iterative process, k' on two sides of (C2) takes different values,

namely, we substitute $k' = \widehat{K}(\cdot; b)$ in the right side of (C2), and we compute the left side of (C2); the FPI iterations on b are performed until the two sides coincide.

(**Algorithm EE): An algorithm iterating on the Euler equation**
Step 0. Initialization.
a. Choose (k_0, a_0) and T.
b. Draw $\{\epsilon_{t+1}\}_{t=0,...,T-1}$. Compute and fix $\{a_{t+1}\}_{t=0,...,T-1}$ using (10).
c. Choose an approximating function $K \approx \widehat{K}(\cdot, b)$.
d. Make an initial guess on b.
e. Choose integration nodes, ϵ_j, and weights, ω_j, $j = 1, ..., J$.
Step 1. Construction of an EDS grid.
a. Use $\widehat{K}(\cdot, b)$ to simulate $\{k_{t+1}\}_{t=0,...,T-1}$.
b. Construct an EDS grid $\Gamma = \{k_m, a_m\}_{m=1,...,M}$.
Step 2. Computation of a solution for K.
a. At iteration i, for $m = 1, ..., M$, compute
– $k'_m = \widehat{K}\left(k_m, a_m; b^{(i)}\right)$ and $a'_{m,j} = a_m^{\rho} \exp(\epsilon_j)$ for all j;
– $k''_{m,j} = \widehat{K}\left(k'_m, a'_{m,j}; b^{(i)}\right)$ for all j;
– $c_m = (1-\delta) k_m + a_m A f(k_m) - k'_m$;
– $c'_{m,j} = (1-\delta) k'_m + a_m^{\rho} \exp(\epsilon_j) A f(k'_m) - k''_{m,j}$ for all j;
– $\widehat{k}'_m \equiv \beta \sum_{j=1}^{J} \omega_j \cdot \left[\frac{u_1(c'_{m,j})}{u_1(c_m)} \left[1 - \delta + a_m^{\rho} \exp(\epsilon_j) A f_1(k'_m)\right] k'_m\right]$.
b. Find b that solve the system in Step 2a.
– Run a regression to get: $\widehat{b} \equiv \arg\min_b \sum_{m=1}^{M} \left\| \widehat{k}'_m - \widehat{K}(k_m, a_m; b) \right\|$.
– Use damping to compute $b^{(i+1)} = (1-\xi) b^{(i)} + \xi \widehat{b}$.
– Check for convergence: end Step 2 if $\frac{1}{M} \sum_{m=1}^{M} \left\| \frac{(k'_m)^{(i+1)} - (k'_m)^{(i)}}{(k'_m)^{(i)}} \right\| < \varpi$.
Iterate on Steps 1, 2 until convergence of the EDS grid.

An EDS algorithm iterating on the Bellman equation. The envelope condition of (12)–(14) is

$$V_1(k, a) = u_1(c)\left[1 - \delta + a A f_1(k)\right]. \tag{C1}$$

We parameterize V with a flexible functional form $\widehat{V}(\cdot; b)$, that depends on a coefficients vector b, and we solve for b using FPI.

(Algorithm BE): An algorithm iterating on the Bellman equation

Step 0. Initialization.

a. Choose (k_0, a_0) and T.

b. Draw $\{\epsilon_{t+1}\}_{t=0,...,T-1}$. Compute and fix $\{a_{t+1}\}_{t=0,...,T-1}$ using (10).

c. Choose an approximating function $\widehat{V}(\cdot; b) \approx V$.

d. Make an initial guess on b.

e. Choose integration nodes, ϵ_j, and weights, ω_j, $j = 1, ..., J$.

Step 1. Construction of an EDS grid.

a. Find $\widehat{K}$ corresponding to $\widehat{V}(\cdot; b)$ to simulate $\{k_{t+1}\}_{t=0,...,T-1}$.

b. Construct an EDS grid $\Gamma = \{k_m, a_m\}_{m=1,...,M}$.

Step 2. Computation of a solution for V.

a. At iteration i, for $m = 1, ..., M$, compute

– $c_m \equiv u_1^{-1}\left[\frac{\widehat{V}_1\left(k_m, a_m; b^{(i)}\right)}{1-\delta+a_m A f_1(k_m)}\right]$;

– $k'_m = (1-\delta)\, k_m + a_m A f(k_m) - c_m$;

– $a'_{m,j} = a_m^{\rho} \exp(\epsilon_j)$ for all j;

– $\widehat{V}\left(k'_m, a'_{m,j}; b^{(i)}\right)$ for all j;

– $V_m \equiv u(c_m) + \beta \sum_{j=1}^{J} \omega_j \cdot \widehat{V}\left(k'_m, a'_{m,j}; b^{(i)}\right)$;

b. Find b that solve the system in Step 2a.

– Run a regression to get: $\widehat{b} \equiv \arg\min_b \sum_{m=1}^{M} \left\| V_m - \widehat{K}(k_m, a_m; b) \right\|$.

– Use damping to compute $b^{(i+1)} = (1-\xi)\, b^{(i)} + \xi \widehat{b}$.

– Check for convergence: end Step 2 if $\frac{1}{M} \sum_{m=1}^{M} \left| \frac{(k'_m)^{(i+1)} - (k'_m)^{(i)}}{(k'_m)^{(i)}} \right| < \varpi$.

Iterate on Steps 1, 2 until convergence of the EDS grid.

Computational choices. We parameterize the model (8)–(10) by assuming $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$ with $\gamma \in \left\{\frac{1}{5}, 1, 5\right\}$ and $f(k) = k^{\alpha}$ with $\alpha = 0.36$. We set $\beta = 0.99$, $\delta = 0.025$, $\rho = 0.95$ and $\sigma = 0.01$. We normalize the steady state of capital to one by assuming $A = \frac{1/\beta - (1-\delta)}{\alpha}$. The simulation length is $T = 100,000$, and we pick each 10th point so that $n = 10,000$. The damping parameter is $\xi = 0.1$, and the convergence parameter is $\varpi = 10^{-11}$. In Algorithm EE and Algorithm BE, we parameterize the capital equilibrium rule and value function, respectively, using complete ordinary polynomials of degrees up to 5. For example, for degree 2, we have $\widehat{K}(k, a; b) = b_0 + b_1 k + b_2 a + b_3 k^2 + b_4 ka + b_5 a^2$, where $b \equiv (b_0, ...b_5)$. We approximate conditional expectations with a 10-node Gauss-Hermite quadrature rule. We

compute the vector of coefficients b using an LS method based on QR factorization. To construct an initial EDS grid, we simulate the model under an (arbitrary) initial guess $k' = 0.95k + 0.05a$ (this guess matches the steady state level of capital equal to one).

After the solution was computed, we evaluate the quality of the obtained approximations on a stochastic simulation. We generate a new random draw of $10,200$ points and discard the first 200 points. At each point (k_i, a_i), we compute an Euler-equation residual in a unit-free form by using a 10-node Gauss-Hermite quadrature rule,

$$\mathcal{R}(k_i, a_i) \equiv \sum_{j=1}^{J^{\text{test}}} \omega_j^{\text{test}} \cdot \left[\beta \frac{u_1\left(c'_{i,j}\right)}{u_1(c_i)} \left[1 - \delta + a_i^{\rho} \exp\left(\epsilon_j^{\text{test}}\right) A f_1\left(k'_i\right)\right] \right] - 1,$$

where c_i and $c'_{i,j}$ are defined similarly to c_m and $c'_{m,j}$ in Step 2a of Algorithm EE, respectively; ϵ_j^{test} and ω_j^{test} are integration nodes and weights, respectively. We report the mean and maximum of absolute value of $\mathcal{R}(k_i, a_i)$.

Our code is written in MATLAB, version 7.6.0.324 (R2008a), and we run experiments using a desktop computer ASUS with Intel(R) Core(TM)2 Quad CPU Q9400 (2.66 GHz), RAM 4MB.

Appendix D. Multicountry model

In this section, we describe the multicountry model studied in Section 4.4 and elaborate a description of the numerical method used to solve this model.

The set up. We describe the multicountry model studied in Section 4.4. A social planner maximizes a weighted sum of expected lifetime utilities of N agents (countries),

$$\max_{\left\{c_t^h, k_{t+1}^h\right\}_{t=0,\ldots,\infty}^{h=1,\ldots,N}} E_0 \sum_{h=1}^{N} \lambda^h \left[\sum_{t=0}^{\infty} \beta^t u^h\left(c_t^h\right) \right] \tag{D1}$$

subject to the aggregate resource constraint,

$$\sum_{h=1}^{N} c_t^h + \sum_{h=1}^{N} k_{t+1}^h = \sum_{h=1}^{N} k_t^h (1 - \delta) + \sum_{h=1}^{N} a_t^h A f^h\left(k_t^h\right), \tag{D2}$$

where $\left\{k_0^h, a_0^h\right\}^{h=1,\ldots,N}$ is given; E_t is the operator of conditional expectation; c_t^h, k_t^h, a_t^h and λ^h are, respectively, consumption, capital, productivity level and welfare

weight of a country $h \in \{1, ..., N\}$; $\beta \in (0, 1)$ is the discount factor; $\delta \in (0, 1]$ is the depreciation rate; A is the normalizing constant in the production function. The utility and production functions, u^h and f^h, respectively, are increasing, concave and continuously differentiable. The process for the productivity level of country h is given by

$$\ln a_{t+1}^h = \rho \ln a_t^h + \epsilon_{t+1}^h, \tag{D3}$$

where ρ is the autocorrelation coefficient; $\epsilon_{t+1}^h \equiv \varsigma_{t+1}^h + \varsigma_{t+1}$ where $\varsigma_{t+1}^h \sim \mathrm{N}(0, \sigma^2)$ is specific to each country and $\varsigma_{t+1} \sim \mathcal{N}(0, \sigma^2)$ is identical for all countries.

We restrict our attention to the case in which the countries are characterized by identical preferences, $u^h = u$, and identical production technologies, $f^h = f$, for all h. The former implies that the planner assigns identical weights, $\lambda^h = 1$, and consequently, identical consumption $c_t^h = c_t$ to all agents. If an interior solution exists, it satisfies N Euler equations,

$$u'(c_t) = \beta E_t \left\{ u_1(c_{t+1}) \left[1 - \delta + a_{t+1}^h A f_1\left(k_{t+1}^h\right) \right] \right\}, \tag{D4}$$

where u_1 and f_1 denote the derivatives of u and f, respectively. Thus, the planner's solution is determined by the process for shocks (D3), the resource constraint (D2), and the set of Euler equations (D4).

Solution procedure. Our objective is to approximate N capital equilibrium rules, $k_{t+1}^h = K^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,...,N}\right)$, $h = 1, ..., N$. Since the countries are identical in their fundamentals (preferences and technology), their optimal equilibrium rules are also identical. We could have used the symmetry to simplify the solution procedure, however, we do not do so. Instead, we compute a separate equilibrium rule for each country, thus, treating the countries as completely heterogeneous. This approach allows us to assess the cost of finding solutions in general multidimensional setups in which countries have heterogeneous preferences and technology.

To solve the model, we parameterize the capital equilibrium rule of each country with a flexible functional form

$$K^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,...,N}\right) \approx \widehat{K}^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,...,N}; b^h\right),$$

where b^h is a vector of coefficients. We rewrite the Euler equation (D4) as

$$k_{t+1}^h = E_t\left\{ \beta \frac{u'(c_{t+1})}{u'(c_t)} \left[1 - \delta + a_{t+1}^h A f'\left(k_{t+1}^h\right) \right] k_{t+1}^h \right\}. \tag{D5}$$

For each country $h \in \{1, ..., N\}$, we need to compute a vector b^h such that, given the functional form of $\widehat{K}^h$, the resulting function $\widehat{K}^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,...,N}; b^h\right)$ is the best possible approximation of $K^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,...,N}\right)$ on the relevant domain.

The steps of the EDS algorithm here are similar to those described of Algorithm EE described in Appendix C for the one-agent model. However, we now iterate on N equilibrium rules of the heterogeneous countries instead of just one equilibrium rule of the representative agent. That is, we make an initial guess on N coefficients vectors $\left\{b^h\right\}^{h=1,...,N}$, approximate N conditional expectations in Step 2a and run N regressions in Step 2b. The damping parameter in $\left(b^h\right)^{(i+1)} = (1-\xi)\left(b^h\right)^{(i)} + \xi\widehat{b}^h$ is $\xi = 0.1$, and the convergence parameter is $\varpi = 10^{-8}$. In the accuracy check, we evaluate the size of Euler equation residuals on a stochastic simulation of length $T^{\text{test}} = 10,200$ (we discard the first 200 observations to eliminate the effect of initial condition). To test the accuracy of solutions, we use the Gauss-Hermite quadrature product rule $Q(2)$ for N up to 12, use the monomial rule $M2$ for N from 12 to 20, and use the monomial rule $M1$ for N larger than 20. We use the same values of the parameters in the multicountry model as in the one-agent model; in particular, we assume $\gamma = 1$.

Appendix E. New Keynesian model with the ZLB

In this section, we derive the first-order conditions (FOCs) and describe the details of our numerical analysis for the new Keynesian economy studied in Section 5.

Households. The FOCs of the household's problem (15)–(19) with respect to C_t, L_t and B_t are

$$\Lambda_t = \frac{\exp\left(\eta_{u,t}\right) C_t^{-\gamma}}{P_t}, \tag{E1}$$

$$\exp\left(\eta_{u,t} + \eta_{L,t}\right) L_t^{\vartheta} = \Lambda_t W_t, \tag{E2}$$

$$\exp\left(\eta_{u,t}\right) C_t^{-\gamma} = \beta \exp\left(\eta_{B,t}\right) R_t E_t \left[\frac{\exp\left(\eta_{u,t+1}\right) C_{t+1}^{-\gamma}}{\pi_{t+1}}\right], \tag{E3}$$

where Λ_t is the Lagrange multiplier associated with the household's budget constraint (16). After combining (E1) and (E2), we get

$$\exp\left(\eta_{L,t}\right) L_t^{\vartheta} C_t^{\gamma} = \frac{W_t}{P_t}. \tag{E4}$$

Final-good producers. The FOC of the final-good producer's problem (20), (21) with respect to $Y_t(i)$ yields the demand for the ith intermediate good

$$Y_t(i) = Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon}. \tag{E5}$$

Substituting the condition (E5) into (21), we obtain

$$P_t = \left(\int_0^1 P_t(i)^{1-\varepsilon} di \right)^{\frac{1}{1-\varepsilon}}. \tag{E6}$$

Intermediate-good producers. The FOC of the cost-minimization problem (22)–(24) with respect to $L_t(i)$ is

$$\Theta_t = \frac{(1-v) W_t}{\exp\left(\eta_{a,t}\right)}, \tag{E7}$$

where Θ_t is the Lagrange multiplier associated with (23). The derivative of the total cost in (22) is the nominal marginal cost, $\mathrm{MC}_t(i)$,

$$\mathrm{MC}_t(i) \equiv \frac{d\mathrm{TC}\left(Y_t(i)\right)}{dY_t(i)} = \Theta_t. \tag{E8}$$

Conditions (E7) and (E8) taken together imply that the real marginal cost is the same for all firms,

$$\mathrm{mc}_t(i) = \frac{(1-v)}{\exp\left(\eta_t^a\right)} \cdot \frac{W_t}{P_t} = \mathrm{mc}_t. \tag{E9}$$

The FOC of the reoptimizing intermediate-good firm with respect to $\widetilde{P}_t$ is

$$E_t \sum_{j=0}^{\infty} (\beta\theta)^j \Lambda_{t+j} Y_{t+j} P_{t+j}^{\varepsilon+1} \left[\frac{\widetilde{P}_t}{P_{t+j}} - \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+j} \right] = 0 \tag{E10}$$

From the household's FOC (E1), we have

$$\Lambda_{t+j} = \frac{\exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma}}{P_{t+j}}. \tag{E11}$$

Substituting (E11) into (E10), we get

$$E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma} Y_{t+j} P_{t+j}^{\varepsilon} \left[\frac{\widetilde{P}_t}{P_{t+j}} - \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+j} \right] = 0. \tag{E12}$$

Let us define $\chi_{t,j}$ such that

$$\chi_{t,j} \equiv \begin{cases} 1 \text{ if } j = 0 \\ \frac{1}{\pi_{t+j} \cdot \pi_{t+j-1} \cdots \pi_{t+1}} \text{ if } j \geq 1 \end{cases} . \tag{E13}$$

Then $\chi_{t,j} = \chi_{t+1,j-1} \cdot \frac{1}{\pi_{t+1}}$ for $j > 0$. Therefore, (E12) becomes

$$E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma} Y_{t+j} \chi_{t,j}^{-\varepsilon} \left[\widetilde{\widetilde{p}}_t \chi_{t,j} - \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+j}\right] = 0, \tag{E14}$$

where $\widetilde{\widetilde{p}}_t \equiv \frac{\widetilde{P}_t}{P_t}$. We express $\widetilde{\widetilde{p}}_t$ from (E14) as follows

$$\widetilde{\widetilde{p}}_t = \frac{E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma} Y_{t+j} \chi_{t,j}^{-\varepsilon} \frac{\varepsilon}{\varepsilon-1} \mathrm{mc}_{t+j}}{E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma} Y_{t+j} \chi_{t,j}^{1-\varepsilon}} \equiv \frac{S_t}{F_t}. \tag{E15}$$

Let us find recursive representations for S_t and F_t. For S_t, we have

$$\begin{aligned} S_t &\equiv E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma} Y_{t+j} \chi_{t,j}^{-\varepsilon} \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+j} \\ &= \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t \mathrm{mc}_t \\ &+ \beta\theta E_t \left\{ \sum_{j=1}^{\infty} (\beta\theta)^{j-1} \exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma} Y_{t+j} \left(\frac{\chi_{t+1,j-1}}{\pi_{t+1}}\right)^{-\varepsilon} \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+j} \right\} \\ &= \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t \mathrm{mc}_t \\ &+ \beta\theta E_t \left\{ \frac{1}{\pi_{t+1}^{-\varepsilon}} \sum_{j=0}^{\infty} (\beta\theta)^j \exp\left(\eta_{u,t+1+j}\right) C_{t+1+j}^{-\gamma} Y_{t+1+j} \chi_{t+1,j}^{-\varepsilon} \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+1+j} \right\} \\ &= \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t \mathrm{mc}_t \\ &+ \beta\theta E_t \left\{ \frac{1}{\pi_{t+1}^{-\varepsilon}} E_{t+1} \left(\sum_{j=0}^{\infty} (\beta\theta)^j \exp\left(\eta_{u,t+1+j}\right) C_{t+1+j}^{-\gamma} Y_{t+1+j} \chi_{t+1,j}^{-\varepsilon} \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+1+j} \right) \right\} \\ &= \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t \mathrm{mc}_t + \beta\theta E_t \left\{ \pi_{t+1}^{\varepsilon} S_{t+1} \right\}. \end{aligned}$$

Substituting mc_t from $(E9)$ into the above recursive formula for S_t, we have

$$S_t = \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t \frac{(1-v)}{\exp\left(\eta_{a,t}\right)} \cdot \frac{W_t}{P_t} + \beta\theta E_t \left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}. \tag{E16}$$

Substituting $\frac{W_t}{P_t}$ from (E4) into (E16), we get

$$S_t = \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) Y_t \frac{(1-v)}{\exp\left(\eta_{a,t}\right)} \cdot \exp\left(\eta_{L,t}\right) L_t^{\vartheta} + \beta\theta E_t \left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}. \tag{E17}$$

For F_t, the corresponding recursive formula is

$$F_t = \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t + \beta\theta E_t \left\{\pi_{t+1}^{\varepsilon-1} F_{t+1}\right\}. \tag{E18}$$

Aggregate price relationship. Condition (E6) can be rewritten as

$$P_t = \left(\int_0^1 P_t\left(i\right)^{1-\varepsilon} di\right)^{\frac{1}{1-\varepsilon}} =$$

$$\left[\int_{\text{reopt.}} P_t\left(i\right)^{1-\varepsilon} di + \int_{\text{non-reopt.}} P_t\left(i\right)^{1-\varepsilon} di\right]^{\frac{1}{1-\varepsilon}}, \tag{E19}$$

where "reopt." and "non-reopt." denote, respectively, the firms that reoptimize and do not reoptimize their prices at t.

Note that $\int_{\text{non-reopt.}} P_t\left(i\right)^{1-\varepsilon} di = \int_0^1 P\left(j\right)^{1-\varepsilon} \omega_{t-1,t}\left(j\right) dj$, where $\omega_{t-1,t}\left(j\right)$ is the measure of non-reoptimizers at t that had the price $P\left(j\right)$ at $t-1$. Furthermore, $\omega_{t-1,t}\left(j\right) = \theta\omega_{t-1}\left(j\right)$, where $\omega_{t-1}\left(j\right)$ is the measure of firms with the price $P\left(j\right)$ in $t-1$, which implies

$$\int_{\text{non-reopt.}} P_t\left(i\right)^{1-\varepsilon} di = \int_0^1 \theta P\left(j\right)^{1-\varepsilon} \omega_{t-1}\left(j\right) dj = \theta P_{t-1}^{1-\varepsilon}. \tag{E20}$$

Substituting (E20) into (E19) and using the fact that all reoptimizers set $\widetilde{P}_t^{1-\varepsilon}$, we get

$$P_t = \left[\left(1-\theta\right)\widetilde{P}_t^{1-\varepsilon} + \theta P_{t-1}^{1-\varepsilon}\right]^{\frac{1}{1-\varepsilon}}. \tag{E21}$$

We divide both sides of (E21) by P_t,

$$1 = \left[\left(1-\theta\right)\widetilde{\widetilde{p}}_t^{1-\varepsilon} + \theta\left(\frac{1}{\pi_t}\right)^{1-\varepsilon}\right]^{\frac{1}{1-\varepsilon}},$$

and express $\widetilde{\widetilde{p}}_t$

$$\widetilde{\widetilde{p}}_t = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}. \tag{E22}$$

Combining (E22) and (E15), we obtain

$$\frac{S_t}{F_t} = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}. \tag{E23}$$

Aggregate output. Let us define aggregate output

$$\overline{Y}_t \equiv \int_0^1 Y_t(i)\,di = \int_0^1 \exp\left(\eta_{a,t}\right) L_t(i)\,di = \exp\left(\eta_{a,t}\right) L_t, \tag{E24}$$

where $L_t = \int_0^1 L_t(i)\,di$ follows by the labor-market clearing condition. We substitute demand for $Y_t(i)$ from (E5) into (E24) to get

$$\overline{Y}_t = \int_0^1 Y_t \left(\frac{P_t(i)}{P_t}\right)^{-\varepsilon} di = Y_t P_t^{\varepsilon} \int_0^1 P_t(i)^{-\varepsilon}\,di. \tag{E25}$$

Let us introduce a new variable $\overline{P}_t$,

$$\left(\overline{P}_t\right)^{-\varepsilon} \equiv \int_0^1 P_t(i)^{-\varepsilon}\,di. \tag{E26}$$

Substituting (E24) and (E26) into (E25) gives us

$$Y_t \equiv \overline{Y}_t \left(\frac{\overline{P}_t}{P_t}\right)^{\varepsilon} = \exp\left(\eta_{a,t}\right) L_t \Delta_t, \tag{E27}$$

where Δ_t is a measure of price dispersion across firms, defined by

$$\Delta_t \equiv \left(\frac{\overline{P}_t}{P_t}\right)^{\varepsilon}. \tag{E28}$$

Note that if $P_t(i) = P_t(i')$ for all i and $i' \in [0,1]$, then $\Delta_t = 1$, that is, there is no price dispersion across firms.

Law of motion for price dispersion Δ_t. By analogy with (E21), the variable $\overline{P}_t$, defined in (E26), satisfies

$$\overline{P}_t = \left[(1-\theta)\,\widetilde{P}_t^{-\varepsilon} + \theta\left(\overline{P}_{t-1}\right)^{-\varepsilon}\right]^{-\frac{1}{\varepsilon}}. \tag{E29}$$

Using (E29) in (E28), we get

$$\Delta_t = \left(\frac{\left[(1-\theta)\,\widetilde{P}_t^{-\varepsilon} + \theta\left(\overline{P}_{t-1}\right)^{-\varepsilon}\right]^{-\frac{1}{\varepsilon}}}{P_t}\right)^{\varepsilon}. \tag{E30}$$

This implies

$$\Delta_t^{\frac{1}{\varepsilon}} = \left[(1-\theta)\left(\frac{\widetilde{P}_t}{P_t}\right)^{-\varepsilon} + \theta\left(\frac{\overline{P}_{t-1}}{P_t}\right)^{-\varepsilon}\right]^{-\frac{1}{\varepsilon}}. \tag{E31}$$

In terms of $\widetilde{\widetilde{p}}_t \equiv \frac{\widetilde{P}_t}{P_t}$, condition (E31) can be written as

$$\Delta_t = \left[(1-\theta)\,\widetilde{\widetilde{p}}_t^{-\varepsilon} + \theta\frac{\overline{P}_{t-1}^{-\varepsilon}}{P_t^{-\varepsilon}}\cdot\frac{P_{t-1}^{-\varepsilon}}{P_{t-1}^{-\varepsilon}}\right]^{-1}. \tag{E32}$$

By substituting $\widetilde{\widetilde{p}}_t$ from (E22) into (E32), we obtain the law of motion for Δ_t,

$$\Delta_t = \left[(1-\theta)\left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{-\frac{\varepsilon}{1-\varepsilon}} + \theta\frac{\pi_t^{\varepsilon}}{\Delta_{t-1}}\right]^{-1}. \tag{E33}$$

Aggregate resource constraint. Combining the household's budget constraint (16) with the government budget constraint (27), we have the aggregate resource constraint

$$P_tC_t + P_t\frac{\overline{G}Y_t}{\exp\left(\eta_{G,t}\right)} = (1-v)\,W_tL_t + \Pi_t. \tag{E34}$$

Note that the ith intermediate-good firm's profit at t is $\Pi_t(i) \equiv P_t(i)\,Y_t(i) - (1-v)\,W_tL_t(i)$. Consequently,

$$\Pi_t = \int_0^1 \Pi_t(i)\,di = \int_0^1 P_t(i)\,Y_t(i)\,di - (1-v)\,W_t\int_0^1 L_t(i)\,di = P_tY_t - (1-v)\,W_tL_t,$$

where $P_t Y_t = \int_0^1 P_t(i) Y_t(i) di$ follows by a zero-profit condition of the final-good firms. Hence, (E34) can be rewritten as

$$P_t C_t + P_t \frac{\overline{G}}{\exp\left(\eta_{G,t}\right)} Y_t = P_t Y_t. \tag{E35}$$

In real terms, the aggregate resource constraint (E35) becomes

$$C_t = \left(1 - \frac{\overline{G}}{\exp\left(\eta_{G,t}\right)}\right) Y_t. \tag{E36}$$

Equilibrium conditions. Condition (34) in the main text follows from (E17) under the additional assumption $\frac{\varepsilon}{\varepsilon-1}(1-v) = 1$ which ensures that the model admits a deterministic steady state (this assumption is commonly used in the related literature; see, e.g., Christiano et al. 2009). Conditions (35)–(40) in the main text correspond to conditions (E18), (E23), (E33), (E3), (E27) and (E36) in the present appendix.

Steady state. The steady state is determined by the following system of equations (written in the order we use to solve for the steady state values):

$$\begin{aligned}
Y_{N*} &= \left[\exp\left(\overline{G}\right)\right]^{\frac{\gamma}{\vartheta+\gamma}}, \\
Y_* &= Y_{N*}, \\
\Delta_* &= (1-\theta\pi_*^{\varepsilon}) \left[(1-\theta)\left(\frac{1-\theta\pi_*^{\varepsilon-1}}{1-\theta}\right)^{\frac{\varepsilon}{\varepsilon-1}}\right]^{-1}, \\
C_* &= \left(1-\overline{G}\right) Y_*, \\
F_* &= C_*^{-\gamma} Y_* + \beta\theta\pi_*^{\varepsilon-1} F_*, \\
S_* &= \frac{\left(1-\overline{G}\right)^{-1} C_*}{\Delta_*} + \beta\theta\pi_*^{\varepsilon} S_*, \\
R_* &= \pi_*/\beta,
\end{aligned}$$

where π_* (the target inflation) and $\overline{G}$ (the steady-state share of government spending in output) are given.

Calibration procedure. Most of the parameters are calibrated using the estimates of Del Negro et al. (2007, Table 1, column "DSGE posterior"); namely, we

assume $\gamma = 1$ and $\vartheta = 2.09$ in the utility function (15); $\phi_y = 0.07$, $\phi_\pi = 2.21$, and $\mu = 0.82$ in the Taylor rule (41); $\varepsilon = 4.45$ in the production function of the final-good firm (21); $\theta = 0.83$ (the fraction of the intermediate-good firms affected by price stickiness); $\overline{G} = 0.23$ in the government budget constraint (27); and $\rho_u = 0.92$, $\rho_G = 0.95$, $\rho_L = 0.25$, $\sigma_u = 0.54\%$, $\sigma_G = 0.38\%$, $\sigma_L = 18.21\%$ (the latter is a lower estimate of Del Negro et al., 2007, Table 1, column "DSGE posterior"), and $\sigma_L = 40.54\%$ (an average estimate of Del Negro et al., 2007) in the processes for shocks (17), (28) and (18). From Smets and Wouters (2007), we take the values of $\rho_a = 0.95$, $\rho_B = 0.22$, $\rho_R = 0.15$, $\sigma_a = 0.45\%$, $\sigma_B = 0.23\%$ and $\sigma_R = 0.28\%$ in the processes for shocks (24), (19) and (30). We set the discount factor at $\beta = 0.99$. To parameterize the Taylor rule (41), we use the steady-state interest rate $R_* = \frac{\pi_*}{\beta}$, and we consider two alternative values of the target inflation, $\pi_* = 1$ (a zero net inflation target) and $\pi_* = 1.0598$ (this estimate comes from Del Negro et al., 2007).

Solution procedure. The EDS method for the new Keynesian model is similar to the one described in Section 4.2 for the neoclassical growth model. We describe the algorithm below.

To approximate the equilibrium rules, we use the family of ordinary polynomials. To compute the conditional expectations in the Euler equations (34), (35) and (40), we use monomial formula $M1$ with $2N$ nodes.

We use the first-order perturbation solution delivered by Dynare as an initial guess (both for the coefficients of the equilibrium rules and for constructing an initial EDS grid). After the solution on the initial EDS solution is computed, we reconstruct the EDS grid and repeat the solution procedure (we checked that the subsequent reconstructions of the EDS grid do not improve the accuracy of solutions).

The simulation length is $T = 100,000$, and we pick each 10th point so that $n = 10,000$. The target number of grid points is $\overline{M} = 1000$. In Step 2b, the damping parameter is set at $\xi = 0.1$, and the convergence parameter is set at $\varpi = 10^{-7}$. We compute residuals on a stochastic simulation of $10,200$ observations (we eliminate the first 200 observations). In the test, we use monomial rule $M2$ with $2 \cdot 6^2 + 1$ nodes which is more accurate than monomial rule $M1$ used in the solution procedure. Dynare does not evaluate the accuracy of perturbation solutions itself. We wrote a MATLAB routine that simulates the perturbation solutions and evaluates their accuracy using the Dynare's representation of the state space which includes the current endogenous state variables $\{\Delta_{t-1}, R_{t-1}\}$, the past exogenous state variables $\{\eta_{u,t-1}, \eta_{L,t-1}, \eta_{B,t-1}, \eta_{a,t-1}, \eta_{R,t-1}, \eta_{G,t-1}\}$ and the current disturbances $\{\epsilon_{u,t}, \epsilon_{L,t}, \epsilon_{B,t}, \epsilon_{a,t}, \epsilon_{R,t}, \epsilon_{G,t}\}$.

(Algorithm EE-NK): An algorithm iterating on the Euler equation

Step 0. Initialization.

a. Choose $\left(\Delta_{-1}, R_{-1}, \eta_{u,0}, \eta_{L,0}, \eta_{B,0}, \eta_{a,0}, \eta_{R,0}, \eta_{G,0}\right)$ and T.

b. Draw $\left\{\epsilon_{u,t+1}, \epsilon_{L,t+1}, \epsilon_{B,t+1}, \epsilon_{a,t+1}, \epsilon_{R,t+1}, \epsilon_{G,t+1}\right\}_{t=0,...,T-1}$.
Compute and fix $\left\{\eta_{u,t+1}, \eta_{L,t+1}, \eta_{B,t+1}, \eta_{a,t+1}, \eta_{R,t+1}, \eta_{G,t+1}\right\}_{t=0,...,T-1}$.

c. Choose approximating functions $S \approx \widehat{S}\left(\cdot; b^S\right)$, $F \approx \widehat{F}\left(\cdot; b^F\right)$, $\text{MU} \approx \widehat{\text{MU}}\left(\cdot; b^{\text{MU}}\right)$.

d. Make initial guesses on b^S, b^F, b^{MU}.

e. Choose integration nodes, $\left\{\epsilon_{u,j}, \epsilon_{L,j}, \epsilon_{B,j}, \epsilon_{a,j}, \epsilon_{R,j}, \epsilon_{G,j}\right\}_{j=1,...,J}$ and weights, $\left\{\omega_j\right\}_{j=1,...,J}$.

Step 1. Construction of an EDS grid.

a. Use $\widehat{S}\left(\cdot; b^S\right)$, $\widehat{F}\left(\cdot; b^F\right)$, $\widehat{\text{MU}}\left(\cdot; b^{\text{MU}}\right)$ to simulate $\left\{S_t, F_t, C_t^{-\gamma}\right\}_{t=0,...,T-1}$.

b. Construct $\Gamma = \left\{\Delta_m, R_m, \eta_{u,m}, \eta_{L,m}, \eta_{B,m}, \eta_{a,m}, \eta_{R,m}, \eta_{G,m}\right\}_{m=1,...,M} \equiv \left\{x_m\right\}_{m=1,...,M}$.

Step 2. Computation of a solution for S, F, MU.

a. At iteration i, for $m = 1, ..., M$, compute

– $S_m = \widehat{S}\left(x_m; b^S\right)$, $F_m = \widehat{F}\left(x_m; b^F\right)$, $C_m = \left[\widehat{\text{MU}}\left(x_m; b^{\text{MU}}\right)\right]^{-1/\gamma}$;

– π_m from $\frac{S_m}{F_m} = \left[\frac{1-\theta\pi_m^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}$ and $\Delta'_m = \left[(1-\theta)\left[\frac{1-\theta\pi_m^{\varepsilon-1}}{1-\theta}\right]^{\frac{\varepsilon}{\varepsilon-1}} + \theta\frac{\pi_m^{\varepsilon}}{\Delta_m}\right]^{-1}$;

– $Y_m = \left(1 - \frac{\overline{G}}{\exp\left(\eta_{G,m}\right)}\right)^{-1} C_m$, and $L_m = Y_m\left[\exp\left(\eta_{a,m}\right)\Delta'_m\right]^{-1}$;

– $Y_{N,m} = \left[\frac{\exp\left(\eta_{a,m}\right)^{1+\vartheta}}{\left[\exp\left(\eta_{G,m}\right)\right]^{-\gamma}\exp\left(\eta_{L,m}\right)}\right]^{\frac{1}{\vartheta+\gamma}}$;

– $R'_m = \max\left\{1, \Phi_m\right\}$, $\Phi_m = R_*\left(\frac{R_m}{R_*}\right)^{\mu}\left[\left(\frac{\pi_m}{\pi_*}\right)^{\phi_\pi}\left(\frac{Y_m}{Y_{N,m}}\right)^{\phi_y}\right]^{1-\mu}\exp\left(\eta_{R,m}\right)$;

– $x'_{m,j} = \left(\Delta'_m, R'_m, \eta'_{u,m,j}, \eta'_{L,m,j}, \eta'_{B,m,j}, \eta'_{a,m,j}, \eta'_{R,m,j}, \eta'_{G,m,j}\right)$ for all j;

– $S'_{m,j} = \widehat{S}\left(x'_{m,j}; b^S\right)$, $F'_{m,j} = \widehat{F}\left(x'_{m,j}; b^F\right)$, $C'_{m,j} = \left[\widehat{\text{MU}}\left(x'_{m,j}; b^{\text{MU}}\right)\right]^{-1/\gamma}$;

– $\pi'_{m,j}$ from $\frac{S'_{m,j}}{F'_{m,j}} = \left[\frac{1-\theta\left(\pi'_{m,j}\right)^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}$;

– $\widehat{S}_m = \frac{\exp\left(\eta_{u,m}+\eta_{L,m}\right)}{\exp\left(\eta_{a,m}\right)}L_m^{\vartheta}Y_m + \beta\theta\sum_{j=1}^{J}\omega_j\cdot\left\{\left(\pi'_{m,j}\right)^{\varepsilon}S'_{m,j}\right\}$,

– $\widehat{F}_m = \exp\left(\eta_{u,m}\right)C_m^{-\gamma}Y_m + \beta\theta\sum_{j=1}^{J}\omega_j\cdot\left\{\left(\pi'_{m,j}\right)^{\varepsilon-1}F'_{m,j}\right\}$,

– $\widehat{C}_m^{-\gamma} = \frac{\beta\exp\left(\eta_{B,m}\right)R'_m}{\exp\left(\eta_{u,m}\right)}\sum_{j=1}^{J}\omega_j\cdot\left[\frac{\left(C'_{m,j}\right)^{-\gamma}\exp\left(\eta'_{u,m,j}\right)}{\pi'_{m,j}}\right]$.

b. Find b^S, b^F, b^{MU} that solve the system in Step 2a.

– Get: $\widehat{b}^S \equiv \underset{b^S}{\text{argmin}}\sum_{m=1}^{M}\left\|\widehat{S}_m - \widehat{S}\left(x_m; b^S\right)\right\|$. Similarly, get $\widehat{b}^F$ and $\widehat{b}^{\text{MU}}$.

– Use damping to compute $b^{(i+1)} = (1-\xi)\,b^{(i)} + \xi\widehat{b}$, where $b \equiv \left(\widehat{b}^S, \widehat{b}^F, \widehat{b}^{\text{MU}}\right)$.

– Check for convergence: end Step 2 if

$$\frac{1}{M}\max\left\{\sum_{m=1}^{M}\left|\frac{(S_m)^{(i+1)}-(S_m)^{(i)}}{(S_m)^{(i)}}\right|, \sum_{m=1}^{M}\left|\frac{(F_m)^{(i+1)}-(F_m)^{(i)}}{(F_m)^{(i)}}\right|, \sum_{m=1}^{M}\left|\frac{(\text{MU}_m)^{(i+1)}-(\text{MU}_m)^{(i)}}{(\text{MU}_m)^{(i)}}\right|\right\} < \varpi.$$

Iterate on Steps 1, 2 until convergence of the EDS grid.

References

[1] Christiano, L., M. Eichenbaum, and S. Rebelo, (2011). When is the government spending multiplier large? Journal of Political Economy 119 (1), 78-121.

[2] Del Negro, M., F. Schorfheide, F. Smets, and R. Wouters, (2007). On the fit of new Keynesian models. Journal of Business and Economic Statistics 25 (2), 123-143.

[3] Everitt, B., S. Landau, M. Leese and D. Stahl, (2011). Cluster Analysis. Wiley Series in Probability and Statistics. Wiley: Chichester, United Kingdom.

[4] Romesburg, C., (1984). Cluster Analysis for Researchers. Lifetime Learning Publications: Belmont, California.

[5] Smets, F. and R. Wouters, (2003). An estimated dynamic stochastic general equilibrium model of the Euro area. Journal of the European Economic Association 1 (5), 1123-1175.

[6] Smets, F. and R. Wouters, (2007). Shocks and frictions in US business cycles: a Bayesian DSGE approach. American Economic Review 97, 586-606.