Merging Simulation and Projection Approaches to Solve High-Dimensional Problems with an Application to a New Keynesian model

Kenneth L. Judd, Lilia Maliar and Serguei Maliar

December 21, 2013

Abstract

We introduce a numerical 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 cover the support of the constructed ergodic measure with a fixed grid, and we use projection techniques to accurately solve the model on that grid. The construction of the grid is the key novel piece of our analysis: we replace a large cloud of simulated points with a small set of "representative" points. We present three alternative techniques for constructing representative points: a clustering method, an epsilon-distinguishable set method, and a locally-adaptive variant of the epsilon-distinguishable set method. As an illustration, we solve one- and multi-agent neoclassical growth models and a large-scale new Keynesian model with a zero lower bound on nominal interest rates. The proposed solution algorithm is tractable in problems with high dimensionality (hundreds of state variables) on a desktop computer.

JEL classification : C61, C63, C68, E31, E52

Key Words : ergodic set; ε-distinguishable set; clusters; adaptive grid; discrepancy; large-scale model; new Keynesian model; ZLB; stochastic simulation

1 Introduction

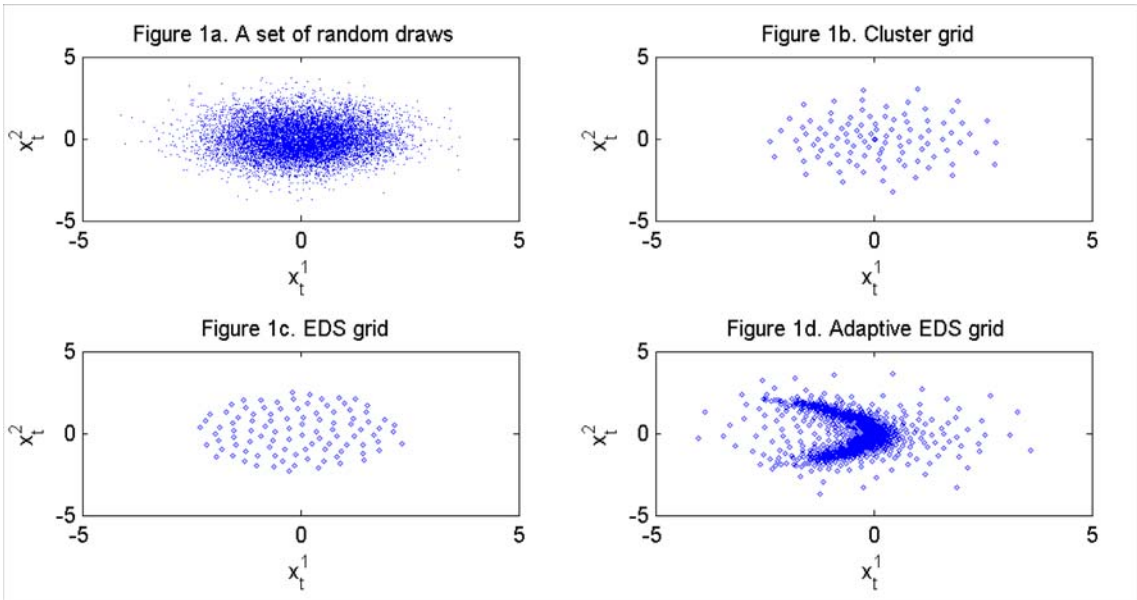
We introduce a numerical algorithm for solving dynamic economic models that merges stochastic simulation and projection approaches: we use simulation to approximate the ergodic measure of the solution, we cover the support of the constructed ergodic measure with a fixed grid, and we use projection techniques to accurately solve the model on that grid. The construction of the grid is the key novel piece of our analysis: we replace a large cloud of simulated points with a small set of "representative" points. We present three alternative techniques for constructing representative points: a clustering method, an epsilon-distinguishable set method, and a locally-adaptive variant of the epsilon-distinguishable set method. As an illustration, we solve one- and multi-agent neoclassical growth models and a large-scale new Keynesian model with a zero lower bound on nominal interest rates. The proposed solution algorithm is tractable in problems with high dimensionality (hundreds of state variables) on a desktop computer.

One broad class of numerical methods for solving dynamic economic models builds on stochastic simulation. First, this class includes iterative methods for solving rational expectations models; see, e.g., Marcet (1988), Smith (1993), Maliar and Maliar (2005) and Judd et al. (2011a). Second, it includes learning-based analysis; see, e.g., Marcet and Sargent (1989), Bertsekas and Tsitsiklis (1996), Pakes and McGuire (2001) and Powell (2011). Finally, it includes methods that use simulation to reduce information sets of decision makers; see, e.g., Krusell and Smith (1998), and Benkard et al. (2008). The key advantage of stochastic simulation methods is that the geometry of the set on which the solution is computed is adaptive. Namely, these methods solve dynamic economic models on a set of points produced by stochastic simulation, avoiding thus the cost of finding solutions in areas of the state space that are effectively never visited in equilibrium; see Judd et al. (2011a) for a discussion. 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 Monte Carlo integration is low).

Another broad class of numerical methods for solving dynamic economic models relies on projection techniques; see, e.g., Wright and Williams (1984), Judd (1992), Christiano and Fisher (2000), Krueger and Kubler (2004), Aruoba et al. (2006), Anderson et al. (2010), Malin et al. (2011), Pichler (2011) and Judd et al. (2013). Projection methods use efficient discretizations of the state space and effective deterministic integration methods, and they deliver very accurate solutions. However, a conventional projection method is limited to a fixed geometry 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 of happening in equilibrium. Moreover, the fraction of the irrelevant areas of a hypercube domain grows rapidly with the dimensionality of the problem.

The solution method introduced in this paper combines best features of stochastic simulation and projection methods, namely, it combines the adaptive geometry of stochastic simulation methods with efficient discretization techniques of projection methods. As an

example, in Figure 1a, we show a set of points that is obtained by simulating two state variables x_t^1 and x_t^2 of a typical dynamic economic model; this set of points identifies a high-probability area of the state space. In Figures 1b, 1c and 1d, we distinguish three different subsets of the simulated points – we call them "grids".



The grid shown in Figure 1b is constructed using methods from clustering analysis: we partition the simulated data into clusters, and we compute the centers of the clusters. The resulting *cluster grid* mimics the density function by placing more points in regions where the cloud of simulated points is more dense and fewer points where it is less dense. The grid shown in Figure 1c is produced by an ε-*distinguishable set* (EDS) technique: we select a subset of points that are situated at the distance at least ε from one another, where $\varepsilon > 0$ is a parameter. The EDS grid is roughly uniform. Finally, in Figure 1d, we show an example of a locally-adaptive EDS grid: instead of using a constant ε, we allow it to vary across the domain, i.e., for each point (x_t^1, x_t^2), we have a different ε, i.e., it is a function $\varepsilon = \varepsilon(x_t^1, x_t^2)$ (in this specific example, we decrease the value of ε as we approach a hyperbola $x_t^1 = [x_t^2]^2$). This kind of grid construction enables us to control the density of grid points (and hence, the quality of approximation) over the solution domain.

An important question is: Which of these techniques delivers the best grid of points to be used within a projection method? Crude simulation shown in Figure 1a is not an efficient choice: having many closely-located grid points does not increase accuracy but cost. Cluster grids tend to produce a good fit in a high-probability area of the state space, but may result in larger errors in low-probability areas of the state space. EDS grids with a constant ε tend to deliver more uniform accuracy. Finally, locally-adaptive EDS grids allow us to automate the control of accuracy over the solution domain using the following two-step procedure: (i) compute a solution using an EDS grid with a constant ε and evaluate the quality of approximation; (ii) define ε to be a decreasing function of the size of approximation errors, construct an EDS grid with a space-dependent ε and recompute

the solution; iterate on steps (i) and (ii) if necessary. Thus, in those areas in which errors are large, we use a smaller ε; this increases the density of grid points and hence, augments the accuracy.

An important role in our analysis plays the choice of an interpolation method, i.e. the way in which we approximate functions off the EDS grid. We consider two kinds of interpolant. One is a global polynomial function that approximates a given decision function on the whole solution domain. The other is a combination of local polynomial bases each of which approximates a decision function only in an a small neighborhood of a given EDS grid point; a global approximation is then obtained by tying up local approximations together. There are many ways to construct local approximations and to tie them up into a global approximation. Our baseline technique is as follows: For each grid point x in the EDS grid, we construct a hypercube centered at that specific point, we populate this hypercube with low-discrepancy points (namely, Sobol points), and we solve the model using those points as a grid; see Niederreiter (1992) for a review of low-discrepancy methods. Here, we compute a solution to the model as many times as the number of points in the EDS grid (since we construct a Sobol grid around each EDS grid point). Finally, to simulate a solution, we use a nearest-neighbor approach.

We next incorporate the EDS and cluster-grid techniques into a numerical method for solving dynamic economic models. Our solution method requires some initial guess about the true solution to the model at the initialization stage, such as a log-linearized solution. In particular, we need an initial guess to produce simulated points which we can use for constructing a grid. We therefore proceed iteratively: guess a solution, simulate the model, construct a grid, solve the model on that grid using a projection method, and perform few iterations on these steps until the grid converges. We complement the efficient grid construction 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 equilibrium rules.[1] Taken together, these techniques make our solution 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 a 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 focus on equilibrium in which target inflation coincides with actual inflation in the steady state. We parameterize the model using the estimates of Smets and Wouters (2003,

[1] In the present paper, we focus on equilibrium problems in which solutions are characterized by Euler equations, however, in a working paper version of the paper, Judd et al. (2012), we also show applications of the EDS technique to dynamic programming.

2007), and Del Negro et al. (2007). The EDS method is tractable for global polynomial approximations of degrees 2 and 3 (at least): the running time is less than 25 minutes in all the 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 rates, 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. Namely, when the ZLB is active, the perturbation method considerably understates the duration of the ZLB episodes and the magnitude of the crises. We also solve the new Keynesian model with an active ZLB using a locally-adaptive EDS method, and we find that consecutive refinements of the EDS grid can considerably increase the quality of approximation.

The mainstream of the literature on new Keynesian models relies on local perturbation solution methods.[2] However, recent developments in numerical analysis triggered a quickly growing body of literature that computes non-linear solutions to medium- and large-scale new Keynesian models; see Judd et al. (2011b), Braun et al. (2012), Coibion et al. (2012), Fernández-Villaverde et al. (2012), Gust et al. (2012), Schmitt-Grohé and Uribe (2012), Aruoba and Schorfheide (2013), Gavion et al. (2013), Mertens and Ravn (2013), Richter and Throckmorton (2013). As is argued in Judd et al. (2011b), Fernández-Villaverde et al. (2012) and Braun et al. (2012), perturbation methods, which were traditionally used in this literature, do not provide accurate approximation in the context of new Keynesian models with the ZLB. Moreover, recent papers of Schmitt-Grohé and Uribe (2012), Mertens and Ravn (2013) and Aruoba and Schorfheide (2013) argue that new Keynesian economies may have multiple equilibria in the presence of ZLB. In particular, the last paper accurately computes a deflation and sunspot equilibria with a full set of stochastic shocks using a modified variant of a *cluster-grid algorithm (CGA)* introduced in Judd et al. (2010, 2011b). Namely, to increase the accuracy of solutions in the ZLB area, first, they add grid points near the ZLB area using the actual data on the U.S. economy; and second, they apply two piecewise bases to separately approximate the solution in the areas with active and non-active ZLB.

Our locally-adaptive EDS methers are related to several other methods in the literature. First, the EDS technique with local bases has similarity to finite element approximation methods that construct a global approximation using a combination of disjoint local approximations; see Hughes (1987) for a mathematic review of finite element methods, and see McGrattan (1996) for their applications to economics. Second, a locally-adaptive

[2]Most papers use linear approximations, however, there are also papers that compute quadratic approximations (e.g., Kollmann, 2002, and Schmitt-Grohé and Uribe, 2007) and cubic approximations (e.g., Rudebusch and Swanson, 2008). Earlier applications of nonlinear solution methods either focus on low-dimensional problems or employ simplifying assumptions; see Adam and Billi (2006), Anderson et al. (2010), and Adjemian and Juillard (2011).

EDS technique with space-dependent ε resembles locally-adaptive sparse-grid techniques which refines an approximation by introducing new grid points and bases functions in those areas in which the quality of approximation is low; see Ma and Zabaras (2009) for a review of such methods, and see Brumm and Scheidegger (2013) for their applications to economic problems. Finally, a locally-adaptive EDS technique is also related to the analysis of Aruoba and Schorfheide (2013) who show the benefits of adaptive grid points and bases functions in the context of a new Keynesian model with the ZLB.

The CGA and EDS projection methods can be used to accurately solve small-scale models that were previously studied using other global solution methods.[3] However, a comparative advantage of these algorithms is their ability to solve large-scale problems that other methods find intractable or expensive. The speed of the CGA and EDS algorithms also makes them potentially useful in estimation methods that solve economic models at many parameters vectors; see Fernández-Villaverde and Rubio-Ramírez (2007), and Winschel and Krätzig (2010). Finally, cluster grids and EDS grids can be used in other applications that require us to produce a discrete approximation to the ergodic distribution of a stochastic process with a continuous density function, in line with Tauchen and Hussey (1991).

The rest of the paper is as follows: In Section 2, we describe the construction of the simulation-based grids using EDS, locally-adaptive EDS and clustering techniques. 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.

2 Discrete approximations to the ergodic set

In this section, we introduce techniques that produce a discrete approximation to the ergodic set of a stochastic process with a continuous density function. Later, we will use the resulting discrete approximation as a grid for finding a solution in the context of a projection-style numerical method for solving dynamic economic models.

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

[3] 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), Adda and Cooper (2003), Aruoba et al. (2006), Den Haan (2010), Kollmann et al. (2011), and Maliar and Maliar (2013).

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 $k_{t+1} = K(k_t, a_t)$ and $a_{t+1} = a_t^{\rho} \exp(\epsilon_{t+1})$, respectively, where $\epsilon_{t+1} \sim \mathcal{N}(0, \sigma^2)$, $\sigma > 0$ and $\rho \in (-1, 1)$; by setting $x_t \equiv (k_t, a_t)$, 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 $\mathcal{A} \in \mathbb{X}$, $\mathcal{P}(\cdot, \mathcal{A})$ is $\mathbb{X}$-measurable function; and (ii) for each point $x \in X$, $\mathcal{P}(x, \cdot)$ is a probability measure on $(X, \mathbb{X})$.*

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

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 $\mathcal{P}(x, \mathcal{A}) = 1$ 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 $\mu(\mathcal{A}) = 0$ or $\mu(\mathcal{A}) = 1$ for every invariant set $\mathcal{A}$.*

These definitions are standard to the literature on dynamic economic models; see Stokey, Lucas and Prescott (1989), and Stachursky (2009). $\mathcal{P}(x, \mathcal{A})$ is the probability that 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 describe 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 $\mathcal{A} \subseteq \mathbb{X}$.*

2.2 An 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. 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 $v_i > 0$.

Assumption 3. *The empirical distribution function $\widehat{\mu}(J) \equiv \frac{C(P;J)}{n}$ converges to the true distribution function $\mu(J)$ for every $J \in \mathcal{J}$ when $n \to \infty$.*

If random draws are independent, the asymptotic rate of convergence of $\widehat{\mu}$ to μ is given by the so-called law of iterated logarithm of Kiefer's (1961), namely, it is $(\log\log n)^{1/2}(2n)^{-1/2}$. For serially correlated processes like (1), the convergence rate depends on specific assumptions; see Zhao and Woodroofe (2008) for the results on general stationary processes.

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 set P only each κth observation to make random draws (approximately) independent. As far as Step 3 is concerned, 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

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

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 Appendix A3.

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 points $x_1,...,x_n \in X \subseteq \mathbb{R}^d$.
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 complexity of Algorithm P^ε is of order $O(nM)$, where M is the number of points into the set P^ε. Indeed, consider the worst-case scenario such that ε is smaller than all inter-point distances for the first M points. Then, the algorithm will go through $n-M$ iterations without eliminating any point, and it will eliminate $n-M$ points at the end. 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 2 in Appendix A2. This means that asymptotically, when $n \to \infty$, the complexity of Algorithm P^ε is linear, $O(n)$.

2.2.3 Distance between points

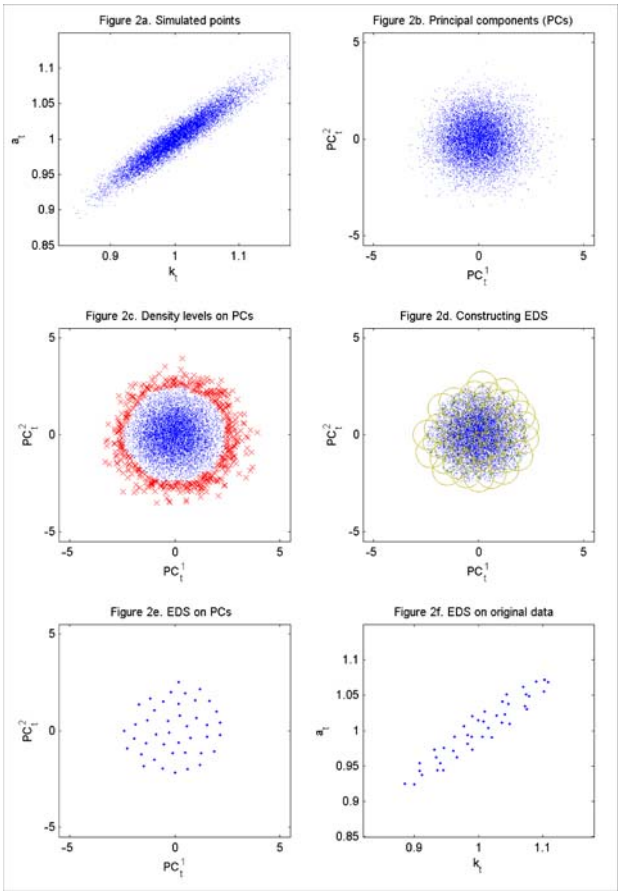
Both estimating the density function and constructing an EDS requires us to measure the distance between simulated points. Generally, variables in economic models have different measurement units and are correlated. This affects the distances between the simulated points and hence, affects the resulting 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 $= \left(\mathrm{PC}^1, ..., \mathrm{PC}^d\right) \in \mathbb{R}^{n \times d}$ are called *principal components* (PCs) of X, and are orthogonal (uncorrelated), i.e., $\left(\mathrm{PC}^{\ell'}\right)^\top \mathrm{PC}^\ell = 0$ 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, $D(x_i, x_j) = \left[\sum_{\ell=1}^d \left(\mathrm{PC}_i^\ell - \mathrm{PC}_j^\ell\right)^2\right]^{1/2}$, where all principal components $\mathrm{PC}^1, ...,\mathrm{PC}^d$ 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 stochastic 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 the 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 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.[4]

2.2.5 Dispersion and discrepancy of EDS grids

In our examples, the EDS grids constructed on simulated series appear to be uniform. However, an important question is whether our construction guarantees the uniformity of

[4]Our two-step procedure produces an approximation not only to the ergodic set but to the whole 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. If our purpose is to construct a set of evenly-spaced points, we do not need to use the density weights and should treat all points equally.

grid points in general. We address this question in Appendices A1 and A3, specifically, we provide formal results about the degrees of dispersion and discrepancy of EDS grids from a uniform distribution.

Our analysis is related to recent mathematical literature on covering-number problems (see, Temlyakov, 2011) and random sequential packing problems (see, Baryshnikov et al., 2008). A well-known example from this literature is a car-parking model of Rényi (1958). Cars of a length ε park at random locations along the roadside of a length one subject to a non-overlap with the previously parked cars. It is known that when cars arrive at uniform random positions, they are also distributed uniformly in the limit $\varepsilon \to 0$.[5].

Our analysis differs from Rényi's (1958) analysis in that cars can arrive at random positions with an arbitrary density function (normally, we do not know density functions of stochastic processes arising in an economic model that we try to solve). In terms of Rényi's (1958) problem, our results are as follows: We show that EDS grids are low-dispersion sequences for any density function, namely, any two points (cars) in the EDS grid are situated on the distance between ε and 2ε from each other, and this distance converges to 0 as $\varepsilon \to 0$ (see Proposition 1 in Appendix A1). However, we find that this fact alone is not sufficient to guarantee the asymptotic uniformity (low discrepancy) of the EDS grids (see Proposition 3 in Appendix A3). To see the intuition, consider Rényi's (1958) setup such that on the interval $[0, \lambda^*]$, evil drivers park their cars on the distance 2ε to leave as little parking space to other drivers as possible, and on the interval $[\lambda^*, 1]$, a police officer directs the cars to park on the distance ε in a socially optimal way. Under this construction, there are twice as many points in the second subinterval as in the first one for any ε (and this non-uniformity is not reduced when $\varepsilon \to 0$). Finally, we establish that even though EDS grids do not possess the property of low-discrepancy sequences in general, their discrepancy from the uniform distribution is bounded from above for any density function; see Proposition 3 in Appendix A3.

2.2.6 Number of points in EDS grids

Under our baseline Algorithm P^{ε}, the cardinality of an EDS grid (i.e., the number of points in it)depends on the value of $\varepsilon > 0$: the smaller is ε, the more points are included in the EDS grid. We derive bounds on the number of points in the EDS grids in Proposition 2 of Appendix A2, however, the exact relation is hard to characterize analytically, in particular, because the cardinality of the EDS grid depends on the order in which points are processed.

In applications, it may be necessary to control the number of grid points, for example, in a projection method, we need to construct a grid with a given number of grid points $\overline{M}$. To construct the relation between ε and $\overline{M} = \overline{M}(\varepsilon)$, we can use a simple numerical bisection method.

[5]Rényi (1958) shows that they occupy about 75% of the roadside at jamming, namely, $\lim_{\varepsilon \to 0} E[M]\varepsilon \approx 0.748$.

(**Algorithm $\overline{M}$): Construction of an EDS with a target number of points $\overline{M}$.**
For iteration $i=1$, fix $\varepsilon_{\min}^{(1)}$ and $\varepsilon_{\max}^{(1)}$ such that $M\left(\varepsilon_{\max}^{(1)}\right) \leq \overline{M} \leq M\left(\varepsilon_{\min}^{(1)}\right)$.
Step 1. On iteration i, take $\varepsilon = \frac{\varepsilon_{\min}^{(i)}+\varepsilon_{\max}^{(i)}}{2}$, construct an EDS and compute $M(\varepsilon)$.
Step 2. If $M(\varepsilon) > \overline{M}$, then set $\varepsilon_{\min}^{(i+1)} = \varepsilon$; and otherwise, set $\varepsilon_{\max}^{(i+1)} = \varepsilon$.
Iterate on Steps 1 and 2 until $M(\varepsilon)$ converges.

To find initial values of $\varepsilon_{\max}$ and $\varepsilon_{\min}$, we use the bounds established in Appendix A2 (see Proposition 2), namely, we set $\varepsilon_{\max}^{(1)} = 0.5 r_{\max} \overline{M}^{-1/d}$ and $\varepsilon_{\min}^{(1)} = r_{\min}\left(\overline{M}^{1/d}-1\right)^{-1}$, where $r_{\max}$ and $r_{\min}$ 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 2), we take $r_{\max}$ and $r_{\min}$ to be the radii of the limiting hyperspheres that contain none and all PCs of the simulated points, respectively.

2.3 Reducing the cost of constructing an EDS on the essentially ergodic set

The two-step procedure described in Section 2.2 has a complexity of $O(n^2)$. This is because the estimation of the density function in Step 3 of Algorithm $\mathcal{A}^\eta$ has a complexity of order $O(n^2)$, and the construction of an EDS set in Step 1 of Algorithm P^ε has a complexity of order $O(nM)$. (The latter is significantly lower than the former if $M \ll n$). The complexity of order $O(n^2)$ does not imply a substantial cost for the size of applications we study in the present paper, however, it might be expensive for larger applications.

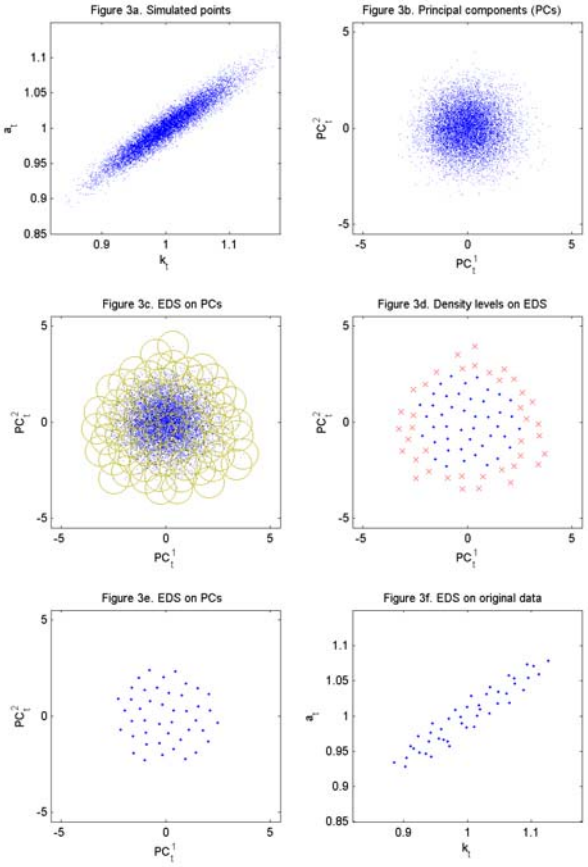
We now describe an alternative implementation of the two-step procedure that has a lower complexity, namely, $O(nM)$. The idea is to invert the steps in the two-step procedure described in Section 2.2, namely, we first construct an EDS with M points using all simulated points, and we then remove from the EDS a subset of points with the lowest density. Since we need to estimate the density function only in M simulated points, the complexity is reduced to $O(nM)$.

(**Algorithm $P^\varepsilon - cheap$): Construction of an EDS.**
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. Select an EDS P^ε of M points, $x_1^\varepsilon, ..., x_M^\varepsilon$ using Algorithm P^ε.
Step 4. Estimate the density function $\widehat{g}(x_i^\varepsilon) \approx g(x_i^\varepsilon)$ for all $x_i^\varepsilon \in P^\varepsilon$ using (2).
Step 5. Remove a fraction of points δ of P^ε 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 $\frac{\widehat{g}\left(x_j^\varepsilon\right)}{\sum_{i=1}^M \widehat{g}\left(x_i^\varepsilon\right)}$ 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 of δ.

To illustrate the application of the above procedure, we again use the example of the neoclassical stochastic growth model with a closed-form solution studied in Section 2.2.4;

see Figures 3a-3f.



We first compute the normalized PCs of the original sample; see Figure 3b (this step in the same as in Figure 2b). We compute an EDS P^{ε} on the normalized PCs; see Figure 3c. We then estimate the density function in all points of P^{ε} using the kernel density algorithm. We next remove 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 3d. The resulting EDS is shown in Figure 3e. Finally, we plot the EDS grid in the original coordinates in Figure 3f.

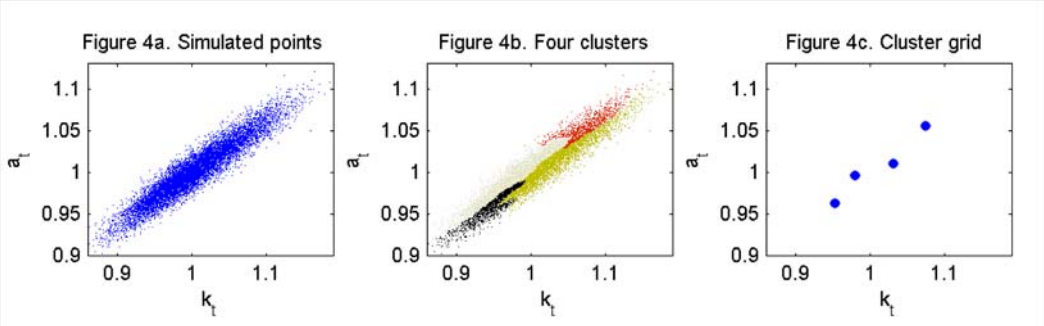
2.4 Cluster-grid technique

We have described one specific EDS procedure for forming a discrete approximation to the essentially ergodic set of tthe stochastic process (1). There are other procedures that can be used for this purpose. In particular, we can use methods from cluster analysis to select a set of representative points from a given set of simulated points; see Everitt et al. (2011) for a review of clustering techniques. Namely, we partition the simulated data into clusters (groups of closely-located points), and we replace each cluster with one representative point. In this paper, we study two clustering methods that can be used in the context of our analysis, an agglomerative hierarchical and K-means ones. The

steps of an agglomerative hierarchical method are shown below, and a K-means method is described in Appendix B2.[6]

(**Algorithm P^c**): **Agglomerative hierarchical clustering algorithm.**
Initialization. Choose M, the number of clusters to be created.
In a zero-order partition $\mathcal{P}^{(0)}$, each simulated point represents a cluster.
Step 1. Compute all pairwise distances between the clusters in a partition $\mathcal{P}^{(i)}$.
Step 2. Merge a pair of clusters with the smallest distance to obtain a partition $\mathcal{P}^{(i+1)}$.
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.

As a measure of distance between two clusters, we use Ward's measure of distance; see Appendix B1. In Figures 4a-4c, 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).



The advantage of the clustering methods is that we can control the number of grid points directly (while the number of points in an EDS is controlled via ε). The drawbacks are that their complexity is higher, namely, 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. Also, 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 cluster grid 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 Section 2.3). Prior to applying the clustering methods, we preprocess the data by constructing the normalized PCs, as we do when constructing an EDS grid in Section 2.2.3.

[6]The clustering methods were used to produce all the numerical results in the earlier versions of the paper, Judd et al. (2010, 2011b). In our examples, projection methods operating on cluster grids and those operating on EDSs deliver comparable accuracy of solutions.

2.5 Locally-adaptive EDS grids

The locally-adaptive EDS grid technique makes it possible to control the quality of approximation over the state space. Namely, we place more grid points in those areas in which the accuracy is low.

In simple cases, we may know a priory that an approximation is less accurate in some area $X_1 \subseteq \mathbb{R}^d$ than in other areas. Consequently, we can use a small ε_1 in the area X_1, and we can use a large ε_2 everywhere else. This produces more dense grid points in X_1 than in the rest of the domain; an example of this construction is shown in Figure 1d. However, in the typical case, it is not a priory known where the solution is accurate, and we proceed as follows:

(**Algorithm** P^ε – *locally adaptive*)**: Construction of a locally adaptive EDS.**
Step 1. Define $\varepsilon_1, ..., \varepsilon_n$ for a given set $x_1, ..., x_n \in X \subseteq \mathbb{R}^d$ (initially, $\varepsilon_i = \varepsilon$ for all i).
Step 2. Construct an EDS P^ε by using ε_i for each $x_i \in X$ and approximate $\widehat{f} \approx f$.
Step 3. Evaluate approximation errors $\mathcal{R}(x_i) = \left\| \widehat{f}(x_i) - f(x_i) \right\|$ for all $x_i \in X$.
Step 4. Define $\mathcal{E}(\mathcal{R}(x_i))$ to be a decreasing function of approximation errors.
Step 5. Compute $\varepsilon_i = \mathcal{E}(\mathcal{R}(x_i))$ for all $x_i \in X$ and go to Step 2.

Under the above algorithm, the larger is the approximation error in a given data point x_i, the smaller is the corresponding value of $\varepsilon_i = \mathcal{E}(\mathcal{R}(x_i))$ and hence, the higher is the density of grid points. In certain sense, this construction is similar to locally-adaptive techniques in the sparse-grid literature in that it refines an approximation by introducing new grid points and bases functions in those areas in which the quality of approximation is low; see Ma and Zabaras (2009) for a review of this literature; and see Brumm and Scheidegger (2013) for examples of economic applications. The locally-adaptive EDS grid technique is especially useful in applications with kinks and strong non-linearities. In Section 5, we will study an example of such an application – a new Keynesian model with a ZLB on the nominal interest rate.

2.6 Approximating a function off the EDS grid

There is a variety of numerical techniques in mathematical literature that can be used to approximate functions off the grid. They typically require us to assume a flexible functional form $\widehat{f}(x; b)$ characterized by a parameters vector b, and to find a parameters vector b that minimizes the approximation errors, $\epsilon(x_i^\varepsilon; b) \equiv \widehat{f}(x_i^\varepsilon; b) - f(x_i^\varepsilon)$, on the constructed EDS grid $x_1^\varepsilon, ..., x_m^\varepsilon \in P^\varepsilon$ according to some norm $\|\cdot\|$. If the constructed $\widehat{f}(\cdot; b)$ coincides with f in all grid points, then we say that $\widehat{f}(\cdot; b)$ interpolates f off the EDS grid (this requires that the number of grid points in the EDS grid is the same as the number of the parameters in b). Otherwise, we say that $\widehat{f}(\cdot; b)$ approximates f on the EDS grid (this is similar to a regression analysis in econometrics when the number of data points is larger than the number of the regression coefficients).

Global polynomial basis functions. A convenient choice for an approximating function is a high-degree ordinary polynomial function. Such a function is easy to construct, and it can be fitted to the data using simple and reliable linear approximation methods; see Judd et al. (2011a). Orthogonal polynomial families are another useful choice even though the property of orthogonality is not satisfied for the simulation-based grid points; see Judd et al. (2011a) for a discussion. However, global polynomial approximations may be not sufficiently flexible to accurately approximate decision functions with strong non-linearities and kinks.

Piecewise local basis functions. Piecewise local bases are more flexible than global ones because each local polynomial basis function approximates a decision function just in an a small neighborhood of a given EDS grid point. A global approximation is obtained by combing local approximations together. There are many ways to construct local approximations and to tie them up into a global approximation. Our baseline technique is as follows: For each grid point $x_i^{\varepsilon_i}$ in the EDS grid, we construct a hypercube centered at that specific point, cover the hypercube with a uniformly distributed set of points, and solve the model on this set of points. As a set of points that covers the hypercube uniformly, we use low-discrepancy sequences, namely, a Sobol sequence; an example of such a sequence is shown in Figure 5b; see Niederreiter (1992) for a review of low-discrepancy methods. Thus, we re-compute a solution to the model as many times as the number of points in the EDS grid. Under piecewise local polynomial approximations, we use low-degree polynomial bases, which helps us to keep the cost reasonably low. Finally, to simulate the solution, we rely on a nearest neighbor approach. Our construction of local bases has similarity to finite-element methods; see Hughes (1987) for a mathematic review of such methods; and see McGrattan (1996) for their applications to economics.

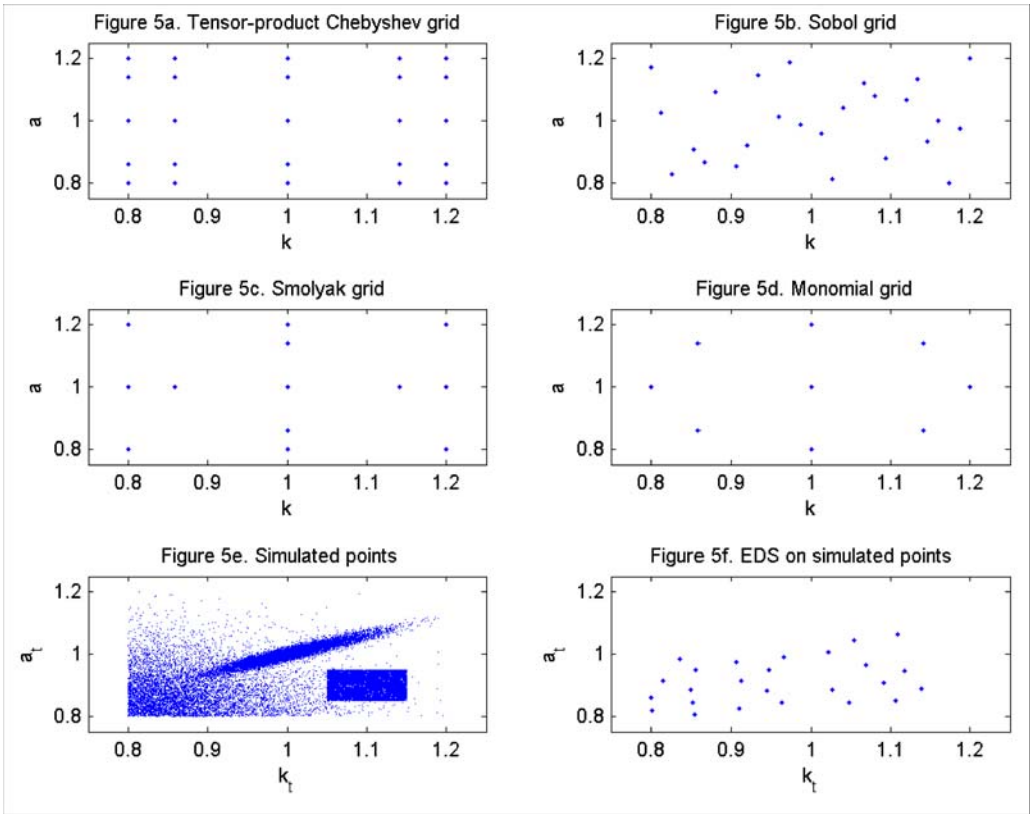
Piecewise local basis functions with locally-adaptive EDS grids. Piecewise local basis functions can be naturally combined with the locally-adaptive EDS grid technique. This combination enables us to refine the solution only in those areas in which the accuracy is not sufficient and to hold fixed the solution in the remaining points. That is, when we add new points to the EDS grid, we need to compute the solutions in the neighborhood of these new grid points but we need not re-compute it in the existing grid points. This useful feature is specific to approximations with local bases functions; for global approximations, we need to re-compute the solution entirely when changing either grid points or an approximating function.

3 Incorporating the EDS grid into projection methods

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 used in the context of numerical solution methods

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 context of projection solution methods is a fixed multidimensional hypercube. Figures 5a-5d plot 4 different discretizations of the hypercube: a tensor-product Chebyshev grid, a low-discrepancy Sobol grid, a sparse Smolyak grid and a monomial grid, respectively (in particular, these type of grids were used in Judd (1992), Rust (1998), Krueger and Kubler (2004), and Pichler (2011), respectively).[7]



In turn, stochastic simulation methods use the adaptive geometry; see Marcet (1988), Smith (1993), Maliar and Maliar (2005), Judd et al. (2011a) for examples of methods that

[7] Also, 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 know the distribution function. It can be applied to both exogenous and endogenous variables.

compute solutions on simulated series.[8] 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 a 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 features of the existing grids. It combines the 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 5e, we show an example of a cloud of simulated points of irregular shape, and in Figure 5f, 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.

There are cases in which the EDS grid may be not 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 have a sufficiently accurate solution outside the high-probability set, for example, when analyzing a transition path of a developing economy with low initial endowment. In those cases, one may augment the grid on a high-probability set to include some "important" points situated outside this set. An example of this approach is shown by Aruoba and Schorfheide (2013) in the context of a new Keynesian model. They construct a grid by combining selected draws from the ergodic distribution of the model with a set of values for state variables filtered from the actual data. In this way, they augment the grid to include points from the 2008-2009 Great recession which do not naturally belong to a high-probability set of the studied model. Finally, our worst case analysis in Appendix A3 shows that there are scenarios in which EDSs constructed on simulated data are highly non-uniform. However, these scenarios require extreme assumptions about the density of simulated points, e.g., a set of highly uneven Dirac point masses. We did not observe the worst-case outcomes in our experiments. If we know that we are in those cases, we may opt for grids on a multidimensional hypercube.

[8] For a detailed description of Marcet's (1988) method, see Den Haan and Marcet (1990), and Marcet and Lorenzoni (1999).

3.2 General description of the EDS algorithm

In this section, we develop a projection method that uses the EDS grid. We focus on equilibrium problems, however, the EDS method can be also used to solve dynamic programming problems; see Judd et al. (2012) for examples.

3.2.1 An equilibrium problem

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

$$E_t \left[G \left(s_t, z_t, y_t, s_{t+1}, z_{t+1}, y_{t+1} \right) \right] = 0, \tag{3}$$

$$z_{t+1} = Z \left(z_t, \epsilon_{t+1} \right), \tag{4}$$

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 $s_{t+1} = S \left(s_t, z_t \right)$, and $y_t = Y \left(s_t, z_t \right)$ that satisfy (3), (4) 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 (3), (4).

(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 (4).

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. For $m = 1, ..., M$, construct residuals

– $\mathcal{R}(s_m, z_m) = \sum_{j=1}^{J} \omega_j \cdot G\left(s_m, z_m, y_m, s'_m, z'_{m,j}, y'_{m,j}\right)$,

where $y_m \equiv \widehat{Y}(s_m, z_m; b^y)$, $s'_m \equiv \widehat{S}(s_m, z_m; b^s)$, $z'_{m,j} \equiv Z(z_m, \epsilon_j)$;

– $y'_{m,j} \equiv \widehat{Y}\left(s'_m, z'_{m,j}; b^y\right)$.

b. Find b^s and b^y that minimize residuals according to some norm.

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

3.2.3 Discussion of the computational choices

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 non-explosive 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). Finding a sufficiently good initial guess can be a non-trivial issue in some applications, and techniques from learning literature can be useful in this context; see Bertsekas and Tsitsiklis (1996) for a discussion.

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. Technically, in Appendix A1, we require 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.

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 parameters to identify. Collocation is a useful technique in the context of orthogonal polynomial constructions but is not convenient in 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 a slightly larger number of points than parameters, which also helps us to increase both accuracy and numerical stability.

Reconstructing the EDS grid iteratively. 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 $\Gamma'' \equiv \left\{x''_j\right\}_{j=1,...,M''}$ 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 is the maximum distance between the grid points on the essentially ergodic set; see Proposition 3 in Appendix A3).

How often do we need to reconstruct the EDS grid? Constructing EDS grids may be costly, especially, in problems with high dimensionality because we need to produce a long simulation, to estimate the density function and to construct EDS grids several times until a bisection procedure locates a grid with the target number of grid points. The cost of constructing EDS grids can be especially high in those applications in which researchers must solve their models repeatedly using different parameters vectors, for example, in estimation or calibration studies.

Hence, an important question is: "How often do we need to reconstruct the EDS grid in a given application?" We found that typically, the properties of solutions are not sensitive to small changes in the EDS grid. For example, the EDS grid constructed on a log-linear solution would normally lead to as accurate non-linear solutions as the one constructed using highly accurate non-linear solutions. Furthermore, we found that small changes in the model's parameters do not require us to re-compute the grid. In the presence of kinks, such as the ZLB in new Keynesian model, the solution is more sensitive to a specific construction of the EDS grid, however, using more accurate solutions for constructing the grid does not necessarily lead to smaller approximation errors. Thus, our experiments suggest that in many applications, we can construct an EDS grid just once using a relatively rough initial guess, and we can keep this grid when iterating on decision functions until convergence (without a visible accuracy loss).

Integration. Unlike simulation- and learning-based methods, 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 class of models; see Judd et al. (2011a) for comparison results.[9] The cost of Gaussian product rules is prohibitive in high-dimensional problems but monomial formulas are tractable even in models with hundreds of state variables; see Judd et al. (2011b) for the description of these formulas.

Solving systems of non-linear equations: the convergence issue. In Step 2 of the EDS algorithm, we must find the parameters vector $b \equiv (b^s, b^y)$ in the decision functions $\widehat{S}(\cdot; b^s)$ and $\widehat{Y}(\cdot; b^y)$ that satisfy the model's equations. Here, we have a system of $n = M \times H$ non-linear equations, where H is the number of equations in the vector function G with n' unknown parameters in b. By construction, $n' \leq n$. If $n' = n$, i.e., we have the same number of unknowns (grid points) as equations, we may have a unique solution that satisfies all equations exactly (this case is referred to as collocation). However, if $n' < n$, we construct a solution that satisfies the model's equations by minimizing a weighed sum of residuals in the model's equations (this case is similar to regression in econometrics).

There is a variety of numerical methods in the literature that can be used to solve a system of non-linear equations in Step 2 of the EDS algorithm, see, e.g., Judd (1998, pp 93-128)) for a review of such methods. In the paper, we restrict attention to a simple derivative-free fixed-point iteration method; see Wright and Willams (1984), Marcet (1988), Den Haan (1990), Gaspar and Judd (1997) for early applications of fixed-point iteration to economic problems. In terms of our problem, fixed-point iteration can be written as follows:

(FPI): Fixed-point iteration with damping.

Initialization. Write a system of equations in the form $\widehat{b} = \Psi(b)$.
Fix initial guess $b^{(0)}$, a norm $\|\cdot\|$ and a convergence criterion ϖ.
Step 1. On iteration i, compute $\widehat{b} = \Psi\left(b^{(i)}\right)$.
Step 2. If $\left\|\widehat{b} - b^{(i)}\right\| < \varpi$, then stop.
Otherwise, set $b^{(i+1)} = \xi\widehat{b} + (1-\xi)\, b^{(i)}$, where $\xi \in (0, 1]$ and go to *Step 1.*

That is, for iteration i, we guess some $b^{(i)}$, compute new $\widehat{b}$ and use it to update our guess for iteration $i+1$, where ξ is the damping parameter that controls the speed of updating. The advantage of fixed-point iteration is that it can iterate in this simple manner on objects of any dimensionality, for example, on a vector of the polynomial coefficients. The cost of this procedure does not grow rapidly with dimensionality of the problem, unlike does the cost of Newton-style methods. As other non-linear solvers, fixed-point

[9] 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 $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ 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, such a slow, $\sqrt{n}$-rate of convergence makes it very expensive to obtain highly accurate solutions using stochastic simulation.

iteration may fail to converge. The following example, borrowed from Judd (1998, p 159), illustrates the possibility of non-convergence.

Example. Let us find a solution to $x^3 - x - 1 = 0$ using a fixed point iteration. We can rewrite it as $x = (x+1)^{1/3}$ and construct a sequence $x^{(i+1)} = (x^{(i)}+1)^{1/3}$ starting from $x^{(0)} = 1$. This yields a sequence $x^{(1)} = 1.26$, $x^{(2)} = 1.31$, $x^{(3)} = 1.32$,... which converges to a solution. However, we can also rewrite this equation as $x = x^3 - 1$ and construct a sequence $x^{(i+1)} = \left(x^{(i)}\right)^3 - 1$ starting from $x^{(0)} = 1$ which diverges to $-\infty$.

This example shows that whether fixed-point iteration succeeds or not in finding a solution may depend on a specific way in which it is implemented. Judd (1998, pp 557-558) also shows that fixed-point iteration may fail to converge in growth models like the ones studied in the present paper under some parameterizations. Damping helps us to increase the likelihood of convergence; see Judd (1998, pp 78-84). Newton-style methods may have better convergence properties but they may also fail if an initial guess is not sufficiently accurate. In sum, CGA and EDS solution methods are effective numerical methods for solving high-dimensional applications, but they share limitations that are common for all projection methods, namely, they may fail to converge. In our examples, the studied EDS and CGA solution methods were 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.

3.2.4 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 which is different from that used in the solution procedure: it is constructed under a different sequence of shocks (i.e., we test accuracy out of sample). Other possible accuracy checks include evaluating

the 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.

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

$$\max_{\{k_{t+1},c_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \tag{5}$$

$$\text{s.t. } c_t + k_{t+1} = (1-\delta) k_t + a_t A f(k_t), \tag{6}$$

$$\ln a_{t+1} = \rho \ln a_t + \epsilon_{t+1}, \qquad \epsilon_{t+1} \sim \mathcal{N}\left(0, \sigma^2\right), \tag{7}$$

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). For numerical experiments, we use $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$ with $\gamma \in \left\{\frac{1}{5}, 1, 5\right\}$ and $f(k) = k^{\alpha}$ with $\alpha = 0.36$, and we set $\beta = 0.99$, $\delta = 0.025$, $\rho = 0.95$ and $\sigma = 0.01$. A version of the model under $u(c) = \ln(c)$, $\delta = 1$ and $f(k) = k^{\alpha}$ admits a closed-form solution $k_{t+1} = \alpha\beta a_t A k_t^{\alpha}$.

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 (5)–(7), the Euler equation is

$$1 = \beta E\left[\frac{u'(c')}{u'(c)}\left(1 - \delta + a' A f'(k')\right)\right], \tag{8}$$

where primes on the variables denote next-period values, and u' and f' denote the derivatives of u and f, respectively. We must solve for equilibrium rules $c = C(k,a)$ and $k' = K(k,a)$ that satisfy (6)–(8). To implement fixed-point iteration, we represent (8) in the form $k' = \Psi(k')$ by multiplying both sides with k' which yields $\widehat{k'} = \beta k' E\left[\frac{u'(c')}{u'(c)}\left(1 - \delta + a' A f'(k')\right)\right]$. In our iterative procedure, we substitute $(k')^{(i)}$ obtained

in iteration i in the right side of this equation, compute $\widehat{k}'$ in the left side and use the solution to improve our guess $(k')^{(i+1)}$ for iteration $i+1$; see Appendix C for a detailed description of the EDS solution method.

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

Table 1: Accuracy and speed of the Euler equation EDS 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

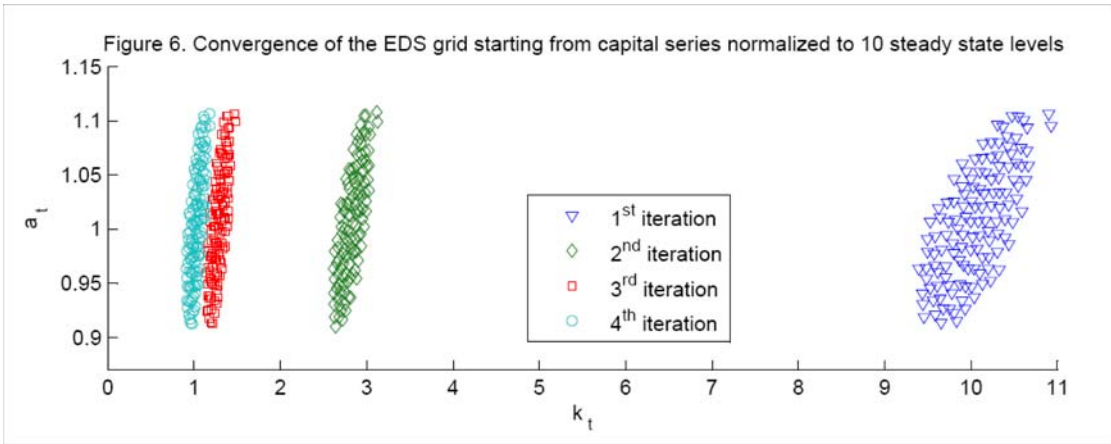
[a] Notes: L_1 and L_∞ are, repectively, the average and maximum of absolute residuals across optimality condition and test points (in log10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in seconds); γ is the coefficient of risk aversion; $M(\varepsilon)$ is the realized number of points in the EDS grid (the target number of grid points is $\overline{M}$=25).

The 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 $\gamma = 5$). Computing high-degree polynomial solutions is relatively fast (a few seconds). Most of the cost of the EDS algorithm comes from the construction of the EDS grid (here, the EDS grid is constructed just once under the polynomial solution of degree 1, and the time for the grid construction is included in the total time for computing that solution). The cost increases gradually with the degree of approximating polynomial because we have a larger number of terms in the approximating functions. This results in a "U"-shaped pattern for the CPU time in the table.

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 optimal growth model, even few quadrature nodes lead to very accurate solutions.

Autocorrection of the EDS grid. If an initial guess about the solution is poor, the simulated series will not cover the ergodic set. Will the EDS grid be able to autocorrect

itself in the context of the EDS algorithm? A general answer to this question is unknown. However, we observe autocorrection of the EDS grid in numerical experiments. In one of such experiments, 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 6 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 had an autocorrection property 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.

EDS grid versus Smolyak grid. Krueger and Kubler (2004) and Malin et al. (2011) develop a projection method that relies on a Smolyak sparse grid. To isolate the role of the grid construction in the algorithm's performance, we implement the Smolyak method in the same way as the EDS method, namely, we use an ordinary polynomial family for approximating decision functions, and we use fixed-point iteration for finding the polynomial coefficients. This implementation of the Smolyak method is in line with the one studied in Judd et al. (2013) and differs from the one in Krueger and Kubler (2004) and Malin et al. (2011) that builds on Smolyak polynomial function and time iteration (in particular, time iteration is more expensive than fixed-point iteration, and Smolyak polynomials have 4 times more basis functions and thus, are more flexible than ordinary polynomials). Thus, under our implementation, the EDS and Smolyak methods differ only in the choice of grid points.

As in Malin et al. (2011), we use intervals $[0.8, 1.2]$ and $\left[\exp\left(-\frac{0.8}{1-\rho}\right), \exp\left(\frac{0.8}{1-\rho}\right)\right]$ for capital and productivity level, respectively. The Smolyak grid has 13 points (see Figure 5c), so we use an EDS grid with the same number of points. With 13 grid points, we can identify the coefficients in ordinary polynomials up to degree 3. In this experiment, we

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

Polynomial degree	Smolyak grid					Accuracy on a hypercube				
	Simulation		Rectangular		CPU	Simulation		Rectangular		CPU
	L_1	L_∞	L_1	L_∞		L_1	L_∞	L_1	L_∞	
1st	-3.31	-2.94	-3.25	-2.54	0,39	-4.23	-3.31	-3.26	-2.38	9.89
2nd	-4.74	-4.17	-4.32	-3.80	0.20	-5.89	-4.87	-4.41	-3.25	0.19
3rd	-5.27	-5.13	-5.39	-4.78	0.22	-7.19	-6.16	-5.44	-4.11	0.17

[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 and on a fixed grid of 100×100 points covering the rectangular with the intervals $[0.8, 1.2]$ and $\left[\exp\left(-\frac{0.8}{1-\rho}\right), \exp\left(\frac{0.8}{1-\rho}\right)\right]$ for capital and productivity, respectively; CPU is the time necessary for computing a solution (in seconds).

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 accuracy results are shown in Table 2. The running time is similar for the Smolyak and EDS methods except that the EDS method needs additional time for constructing the grid.

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, 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.[10]

4.3 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 (5)–(7) to include multiple agents. This is a simple way to expand and to control the size of the problem.

[10] Kollmann et al. (2011b) 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 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.

The set up. 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. A social planner maximizes a weighted sum of expected lifetime utilities of N agents (countries),

$$\max_{\left\{c_t^h, k_{t+1}^h\right\}_{t=0,\ldots,\infty}^{h=1,\ldots,N}} E_0 \sum_{h=1}^{N} \lambda^h \left[\sum_{t=0}^{\infty} \beta^t u^h\left(c_t^h\right)\right] \tag{9}$$

subject to the aggregate resource constraint,

$$\sum_{h=1}^{N} c_t^h + \sum_{h=1}^{N} k_{t+1}^h = \sum_{h=1}^{N} k_t^h (1-\delta) + \sum_{h=1}^{N} a_t^h A f^h\left(k_t^h\right), \tag{10}$$

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

$$\ln a_{t+1}^h = \rho \ln a_t^h + \epsilon_{t+1}^h, \tag{11}$$

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,

$$u'(c_t) = \beta E_t\left\{u'(c_{t+1})\left[1 - \delta + a_{t+1}^h A f'\left(k_{t+1}^h\right)\right]\right\}, \tag{12}$$

where u' and f' denote the derivatives of u and f, respectively. Thus, the planner's solution is determined by the process for productivity (11), the resource constraint (10), and the set of Euler equations (12). We use the same values of the parameters for the multicountry model as in the one-agent model; in particular, we assume $\gamma = 1$.

Solution procedure. Our objective is to 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), i.e., $k_{t+1}^h = K^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,\ldots,N}\right)$, $h = 1, ..., N$. Since the countries are identical in their fundamentals (preferences and technology), the planner chooses the same level of consumption for all countries. We could have used this symmetry to simplify the solution procedure, however, we do not do so. Instead, we compute a decision rule of each country separately, treating them as completely heterogeneous.

In this manner, we can assess the cost of finding solutions in general multidimensional setups in which countries have heterogeneous preferences and technology. For each country, we essentially implement the same computational procedure as the one used in the representative-agent case; see Appendix D for details of the computational procedure.

The choice of an integration method used plays an important role in the accuracy and speed of our solution algorithm. The Monte Carlo method produces large sampling errors which dominate the overall accuracy of solutions. Quadrature product rules are accurate but their cost is prohibitive if the number of shocks is large. However, non-product monomial integration methods both produce very accurate solutions and are tractable in problems with high dimensionality. Moreover, in the studied class of models, an extremely simple and cheap integration method – a one-node quadrature rule – happens to produce accurate solutions; see Judd et al. (2011a) for a detailed description of this and other integration methods; and see Judd et al. (2011a, 2012) and Maliar et al. (2011) for accuracy comparisons of different integration methods in large-scale applications.

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 decision functions to approximate. The number of terms in an approximating polynomial function goes up, and we need to increase the number of grid points to identify the polynomial coefficients. The number of nodes in integration formulas also increases. Finally, operating with large data sets can lead to a memory congestion. If a solution method relies on product rules in constructing a grid, integration nodes, optimization, its cost increases exponentially (curse of dimensionality). However, our design of the EDS method does not rely on product rules and its cost grows with dimensionality of the problem 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 3.

Table 3: Accuracy and speed of the EDS algorithm in the multi-agent model.[a]

	$N=2$			$N=20$			$N=40$			$N=200$		
	Monom. $2N$ nodes			Monom. $2N$ nodes			GH 1−node			GH 1−node		
Polyn.	$\overline{M}$ =300			$\overline{M}$ =1000			$\overline{M}$ =4000			$\overline{M}$ =1000		
degree	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
1st	-4.70	-3.17	0.7	-4.76	-3.05	21	-4.79	-3.09	89	-4.66	-2.90	105
2nd	-6.01	-4.06	1.9	-5.88	-4.14	282	-5.48	-4.13	1463			

[a] Notes: L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in minutes); $\overline{M}$ is the target umber of points in the EDS grid, respectively; "Monom. $2N$" and "GH 1–node" are the monomial rule with two nodes and one-node integration rule, respectively.

The accuracy of solutions here is similar to that we had 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. 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; for sensitivity results, see Table 8 in an earlier version of the present paper, Judd et al. (2010).

The running time ranges from 36 seconds to 24 hours depending on the number of countries, polynomial degree and the integration technique used; see Judd et al. (2012) for sensitivity experiments. 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 one-node quadrature integration rule. Thus, the EDS algorithm is tractable in much larger problems than those studied in 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 (in the given application, this cheap combination produces a very accurate approximation).

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.[11] Our setup builds on the models considered in Christiano et al. (2005), Smets and Wouters (2003, 2007), Del Negro et al. (2007). This literature estimates new Keynesian models using the data on actual economies, while we use their parameters 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.[12] The studied 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 methods, most papers compute linear approximations, however, there are papers that compute quadratic approximations (e.g., Kollmann, 2002, and Schmitt-Grohé and Uribe, 2007) and cubic approximations (e.g., Rudebusch and Swanson, 2008). The earlier literature that used global solution methods includes Adam and Billi (2006), Anderson et al. (2010), and Adjemian and

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

[12] For the neoclassical growth model studied in Section 4, it would be also interesting to explore the case with occasionally binding borrowing constraints. Christiano and Fisher (2000) show how projection methods could be used to solve such a model.

Juillard (2011). The above studies have either few state variables or employ simplifying assumptions.[13] However, recent literature, equipped with novel solution methods, started an exploration of medium- and large-scale new Keynesian models. In particular, Judd et al. (2011b), Fernández-Villaverde et al. (2012) and Braun et al. (2012) show that conventional perturbation methods do not provide accurate approximations in the context of new Keynesian models with the ZLB. Moreover, recent papers by Schmitt-Grohé and Uribe (2012), Mertens and Ravn (2013) and Aruoba and Schorfheide (2013) find that new Keynesian models may have multiple equilibria in the presence of ZLB, and in particular, the last paper accurately computes such multiple equilibria with a full set of stochastic shocks. In the present paper, we focus on the conventional equilibrium in which target inflation coincides with actual inflation in the steady state.

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

$$\max_{\{C_t,L_t,B_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma}-1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta}-1}{1+\vartheta}\right] \tag{13}$$

$$\text{s.t. } P_t C_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} + T_t = B_{t-1} + W_t L_t + \Pi_t, \tag{14}$$

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 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

$$\eta_{u,t+1} = \rho_u \eta_{u,t} + \epsilon_{u,t+1}, \qquad \epsilon_{u,t+1} \sim \mathcal{N}\left(0, \sigma_u^2\right), \tag{15}$$

$$\eta_{L,t+1} = \rho_L \eta_{L,t} + \epsilon_{L,t+1}, \qquad \epsilon_{L,t+1} \sim \mathcal{N}\left(0, \sigma_L^2\right), \tag{16}$$

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

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

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

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

$$\max_{Y_t(i)} \quad P_t Y_t - \int_0^1 P_t(i) Y_t(i) \, di \tag{18}$$

$$\text{s.t. } Y_t = \left(\int_0^1 Y_t(i)^{\frac{\varepsilon-1}{\varepsilon}} di \right)^{\frac{\varepsilon}{\varepsilon-1}}, \tag{19}$$

where (19) 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),

$$\min_{L_t(i)} \quad \text{TC}(Y_t(i)) = (1-v) W_t L_t(i) \tag{20}$$

$$\text{s.t. } Y_t(i) = \exp\left(\eta_{a,t}\right) L_t(i), \tag{21}$$

$$\eta_{a,t+1} = \rho_a \eta_{a,t} + \epsilon_{a,t+1}, \qquad \epsilon_{a,t+1} \sim \mathcal{N}\left(0, \sigma_a^2\right), \tag{22}$$

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,

$$\max_{\widetilde{P}_t} \quad \sum_{j=0}^{\infty} \beta^j \theta^j E_t \left\{ \Lambda_{t+j} \left[\widetilde{P}_t Y_{t+j}(i) - P_{t+j} \text{mc}_{t+j} Y_{t+j}(i) \right] \right\} \tag{23}$$

$$\text{s.t. } Y_t(i) = Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon}, \tag{24}$$

where (24) is the demand for an intermediate good i (follows from the first-order condition of (18), (19)); Λ_{t+j} is the Lagrange multiplier on the household's budget constraint (14); 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

$$T_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} = P_t \frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)} + B_{t-1} + v W_t L_t, \tag{25}$$

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,

$$\eta_{G,t+1} = \rho_G \eta_{G,t} + \epsilon_{G,t+1}, \qquad \epsilon_{G,t+1} \sim \mathcal{N}\left(0, \sigma_G^2\right), \tag{26}$$

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

$$\Phi_t \equiv R_* \left(\frac{R_{t-1}}{R_*}\right)^{\mu} \left[\left(\frac{\pi_t}{\pi_*}\right)^{\phi_\pi} \left(\frac{Y_t}{Y_{N,t}}\right)^{\phi_y}\right]^{1-\mu} \exp\left(\eta_{R,t}\right), \tag{27}$$

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,

$$\eta_{R,t+1} = \rho_R \eta_{R,t} + \epsilon_{R,t+1}, \qquad \epsilon_{R,t+1} \sim \mathcal{N}\left(0, \sigma_R^2\right), \tag{28}$$

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

$$\max_{\{C_t, L_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma} - 1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta} - 1}{1+\vartheta}\right] \tag{29}$$

$$\text{s.t. } C_t = \exp\left(\eta_{a,t}\right) L_t - G_t, \tag{30}$$

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 (15), (16), (22), and (26), respectively. The FOCs of the problem (29), (30) imply that $Y_{N,t}$ depends only on exogenous shocks,

$$Y_{N,t} = \left[\frac{\exp\left(\eta_{a,t}\right)^{1+\vartheta}}{\left[\exp\left(\eta_{G,t}\right)\right]^{-\gamma} \exp\left(\eta_{L,t}\right)}\right]^{\frac{1}{\vartheta+\gamma}}. \tag{31}$$

5.2 Summary of equilibrium conditions

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

$$S_t = \frac{\exp\left(\eta_{u,t}+\eta_{L,t}\right)}{\exp\left(\eta_{a,t}\right)} L_t^{\vartheta} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}, \tag{32}$$

$$F_t = \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon-1} F_{t+1}\right\}, \tag{33}$$

$$C_t^{-\gamma} = \frac{\beta \exp\left(\eta_{B,t}\right) R_t}{\exp\left(\eta_{u,t}\right)} E_t\left[\frac{C_{t+1}^{-\gamma} \exp\left(\eta_{u,t+1}\right)}{\pi_{t+1}}\right], \tag{34}$$

$$\frac{S_t}{F_t} = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}, \tag{35}$$

$$\Delta_t = \left[(1-\theta)\left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{\varepsilon}{\varepsilon-1}} + \theta\frac{\pi_t^{\varepsilon}}{\Delta_{t-1}}\right]^{-1}, \tag{36}$$

$$Y_t = \exp\left(\eta_{a,t}\right) L_t \Delta_t, \tag{37}$$

$$C_t = \left(1 - \frac{\overline{G}}{\exp\left(\eta_{G,t}\right)}\right) Y_t, \tag{38}$$

$$R_t = \max\left\{1,\ \Phi_t\right\}, \tag{39}$$

where Φ_t is given by (27); 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 (32)–(38) correspond, respectively, to (E17), (E18), (E23), (E33) and (E3) in Appendix E.

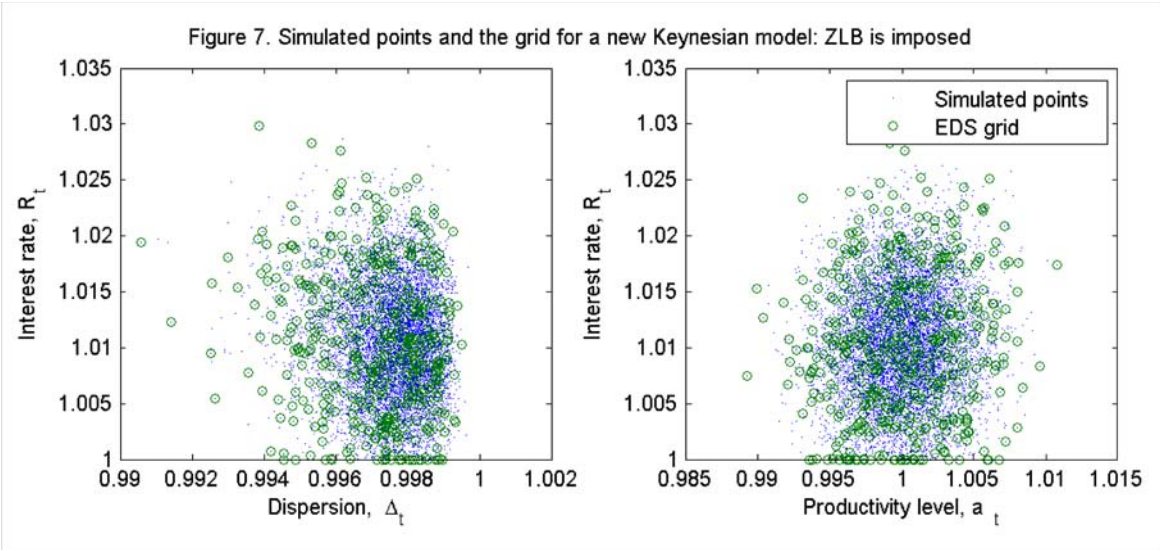
An interior equilibrium is described by 8 equilibrium conditions (32)–(39), and 6 processes for exogenous shocks, (15)–(17), (22), (28) and (26). The system of equations must be solved with respect to 8 unknowns $\{C_t, Y_t, L_t, \pi_t, \Delta_t, R_t, S_t, F_t\}$. There are 2 endogenous and 6 exogenous state variables, , $\{\Delta_{t-1}, R_{t-1}\}$, and $\left\{\eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{a,t}, \eta_{R,t}, \eta_{G,t}\right\}$, respectively.

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 the equilibrium rules $S_t = S\left(x_t\right)$, $F_t = F\left(x_t\right)$ and $C_t^{-\gamma}$ =MU(x_t) with $x_t = \left\{\Delta_{t-1}, R_{t-1}, \eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{a,t}, \eta_{R,t}, \eta_{G,t}\right\}$ using the Euler equations (32), (33) and (34), 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 7 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.

Figure 7. Simulated points and the grid for a new Keynesian model: ZLB is imposed



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 in a high-inflation economy.[14]

In Table 4, we show how the parameters σ_L and π_* affect the quality of numerical solutions.

[14] 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 4. 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
dif(*C), %*	1.00	0.19	0.12	0	6.57	1.49	0.72	0	4.31	3.65	2.26	0
dif(*Y), %*	1.00	0.19	0.12	0	6.57	1.48	0.72	0	4.33	3.65	2.26	0
dif(*L), %*	0.65	0.33	0.16	0	3.16	1.30	0.54	0	3.37	3.17	2.45	0
dif(π), %	0.30	0.16	0.11	0	1.05	0.79	0.60	0	1.17	1.39	0.79	0

[a] Notes: 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.

In the first experiment, we neglected the ZLB: the goal of this experiment is to show that net interest rates will be occasionally negative if ZLB is not imposed. 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 to 0.9916, and the frequency of $R_t < 1$ is about 2%.

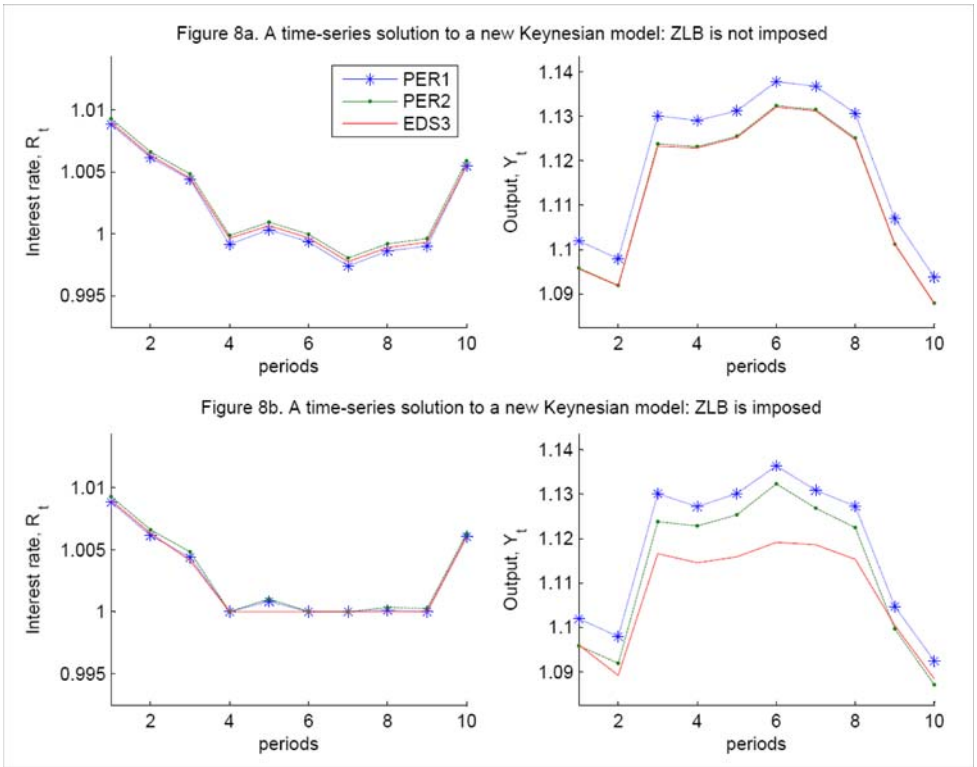
We design the next two experiments to separate the effect of the volatility of labor shocks σ_L and the inflation rate $\pi_* = 1$ on the quality of numerical solutions. In the second experiment, we consider a higher volatility of labor $\sigma_L = 0.4054$, and we set $\pi_* = 1.0598$, which is sufficient to preclude net nominal interest rates from being negative. The performance of the perturbation methods becomes significantly worse. 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 us 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 us increase the quality of approximation.

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 the variables produced by EDS3 and the other methods. The regularities are similar to those we observed for the residuals. First, the difference between the series produced by PER1 and EDS3 can be as large as 17%. Second, 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. Finally, the difference between the series produced by EDS2 and EDS3 is smaller in all cases (5.04% at most for the model with the imposed ZLB). Generally, the supplementary variables S_t and F_t differ more across methods than economically relevant variables such as Y_t, L_t and C_t.

Economic importance of the ZLB. Figures 8a and 8b 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 8a). When the ZLB is imposed, the EDS methods, 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 8b).



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 off 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%.

Piecewise local basis functions and locally adaptive EDS grids Experiments 1 and 3 in Table 4 showed that imposing the ZLB on the nominal interest rate reduces the accuracy of solutions produced by the EDS method. Increasing the degree of an approximating polynomial function does not increase much the accuracy. In particular, the maximum residuals of the second- and third-order solutions are $10^{-2.02}$ and $10^{-2.73}$ in the model with an inactive ZLB, while these residuals are respectively $10^{-1.58}$ and $10^{-1.81}$ in the model with an active ZLB. The residuals are large in the model with an active ZLB for two reasons: first, a global polynomial function is not sufficiently flexible to accurately

approximate the kink in the ZLB area and second, an evenly-spaced EDS grid does not produce sufficiently many grid points in the ZLB area. Below, we perform two additional experiments in which we analyze how the degree of flexibility of an approximating function and the specific placement of grid points affect the accuracy of solutions under the EDS method.

In the first experiment, we replace a global polynomial approximating function with piecewise local bases as described in Section 2.6. Namely, we first construct an EDS grid with M grid points, $x_1^{\varepsilon}, ..., x_M^{\varepsilon}$, using a constant value of ε; we then construct an 8-dimensional hypercube with the side of $10^{-3}\varepsilon$ that surrounds each grid point x_i^{ε}, and we populate it with a 100 low-discrepancy, Sobol, points; we then solve the model on each of the M hypercubes constructed. To simulate the model, we compute a distance from the current state x_i to all grid points, and we adopt a solution from the closest grid point - the nearest neighbor approach. In all experiments, we use second-degree ordinary polynomials as local bases. In the first row of Table 5 (see the experiment "LB"), we show the results depending on the number of local approximations used. As is seen from the

Table 5: Accuracy and speed of the EDS algorithm with local bases and locally adaptive grid in the new Keynesian model.[a]

	$\overline{M}$ =2			$\overline{M}$ =10			$\overline{M}$ =100			$\overline{M}$ =500		
	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
LB	-3.80	-1.66	4.2	-4.15	-1.83	6.4	-4.54	-2.29	29.3	-4.85	-2.35	131
LB-LA	-4.04	-1.75	4.7	-4.14	-1.93	6.6	-4.53	-2.21	34.7	-4.90	-2.25	139

[a] Notes: L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in minutes); $\overline{M}$ is the target number of points in the EDS grid, respectively; "LB" and "LB-LA" denote the EDS method with local bases and the EDS method with both local bases and locally-adaptive grid, respectively.

table, the accuracy of solutions increases with the number of local approximations used although the improvements become smaller as the number of approximations increases. In particular, we are able to reduce the maximum residuals to the order of $10^{-2.35}$ when constructing 500 local approximations. The running time for this experiment is about 2 hours.

In our second experiment, we study a variant of the EDS method in which we combine locally-adaptive grid points with local bases. There are many ways to construct an inverse relation between the size of the residuals and the value of ε; the specific construction of this relation may considerably affect the accuracy results. In our example, we use the following procedure: In each simulated point x_i, we compute $\varepsilon_i = |\ln \mathcal{R}(x_i)|\,\overline{\varepsilon}$, where $\mathcal{R}(x_i)$ is the maximum residual across all model's equations, and $\overline{\varepsilon}$ is a normalizing parameter. For example, if for some points x' and x'', the residuals are $\ln \mathcal{R}(x') = -2$ and $\ln \mathcal{R}(x'') = -6$, then the corresponding value of ε differs by a factor of 3, i.e., $\varepsilon' = 2\overline{\varepsilon}$ and $\varepsilon' = 6\overline{\varepsilon}$. We compute the normalizing parameter $\overline{\varepsilon}$ using our bisection procedure to obtain the target

number of grid points, $\overline{M}$.

The results about the EDS method with both local bases and locally adaptive grid are shown in the second row of Table 5 (see the experiment "LB-LA"). We observe that the EDS method with locally adaptive grid points, LB-LA, has large accuracy improvements than the EDS method with uniformly spaced grid points, LB, in the first row of Table 5. In particular, when $\overline{M} = 2$, the average residuals for the two methods, LB-LA and LB, are $10^{-4.04}$ and $10^{-3.80}$, respectively, and the maximum residuals are $10^{-1.75}$ and $10^{-1.66}$, respectively. However, as the number of grid points increases, both methods arrive to similar accuracy levels. In fact, for large values of $\overline{M}$, the EDS method with an evenly-spaced grid, LB, slightly overpeforms the locally-adaptive EDS method, LB-LA. Indeed, when we have to select just two areas, the way in which these areas are selected is relatively more important than if we have to select 500 areas.

We shall finally discuss the relation between our locally-adaptive EDS solution method and those studied in Aruoba and Schorfheide (2013). To increase the accuracy of solutions in the ZLB area, Aruoba and Schorfheide (2013) implemented two modifications to the baseline cluster-grid algorithm studied in Judd et al. (2010, 2011b): first, they add grid points near the ZLB area using the actual data on the US economy; and second, they use two piecewise local bases to separately approximate the solution in the areas with active and non-active ZLB. For this special case, the locally-adaptive EDS technique with $\overline{M} = 2$ delivers a similar type of analysis if one of the two EDS grid points is selected in a neighborhood of the ZLB area, namely, we also approximate the solution on two disjoint sets of 100 Sobol points that represent the areas with an active and inactive ZLB using two piecewise local bases. The key novelty of the locally-adaptive EDS method is that it automates the construction of adaptive grid points and local basis functions in a general case of $\overline{M}$ piecewise local approximations.

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 or local bases functions, 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 the EDS projection algorithm can be increased by adapting the density of grid points to a given application and by using more flexible functional families that can accommodate kinks and strong non-linearities. This increases the computational cost but the studied EDS methods are naturally parallelizable, and the cost can be reduced.

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 and accuracy of the EDS algorithm can be increased by far using more powerful hardware and software, as well as parallelization techniques.

Aknowledgements

First of all, we thank Editor of Quantitative Economics Frank Schorfheide for his guidance and many useful comments that led to a substantial improvement of the paper. We are also indebted to Editor of Econometrica Jean-Marc Robin and five anonymous referees for a large number of comments and suggestions in three rounds of the revision. Also, we benefited from the comments of seminar participants at University of London, Canadian Central Bank, Cornell University, Federal Reserve Bank of San Francisco, Penn State University, Santa Clara University, Stanford University, University Autonoma de Barcelona, University of Alicante, University of Bilbao, University of California at Berkeley, University of Chicago, University of Edinburgh, and University of Oxford, as well as from participants of the 2011 Society of Economic Dynamics meeting, the 2011, 2012 and 2013 meetings of Society of Computational Economics, the 2013 meeting of Macro Financial Modeling group at the University of Chicago and the Summer 2013 Computation in CA Workshop at Stanford University. The idea of locally-adaptive EDS grids was suggested to us by Debora Lucas. Errors are ours. Lilia Maliar and Serguei Maliar acknowledge support from the Hoover Institution and Department of Economics at Stanford University, University of Alicante, Ivie, MECD and FEDER funds under the projects SEJ-2007-62656 and ECO2012-36719.

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] Adda, J. and R. Cooper, (2003). Dynamic Economics: Quantitative Methods and Applications. London, England: The MIT Press, Cambridge Massachusetts.

[3] 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.

[4] 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.

[5] 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.

[6] Aruoba, S., J. Fernández-Villaverde and J. Rubio-Ramírez, (2006). Comparing solution methods for dynamic equilibrium economies. Journal of Economic Dynamics and Control 30, 2477-2508.

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

[8] 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.

[9] Benhabib, J., S. Schmitt-Grohé and M. Uribe, (2001a). Monetary policy and multiple equilibria. American Economic Review 91(1), 167-186.

[10] Benkard, Weintraub and van Roy (2008). Markov perfect industry dynamics with many firms. Econometrica 76(6), 1375-1411.

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

[12] Braun, R.A., L.M.Körber , and Y. Waki (2012). Some unpleasent properties of log-linearized solutions when the nominal rate is zero. Federal Reserve Bank of Atlanta Working Paper Series.

[13] Brumm, J. and S. Scheidegger, (2013). Using adaptive sparse grids to solve high-dimensional dynamic models. Manuscript. University of Zurich.

[14] 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.

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

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

[17] 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.

[18] 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.

[19] Coibion, O., Y. Gorodnichenko and J. Wieland, (2012), The optimal inflation rate in new Keynesian models: should central banks raise their inflation targets in light of the zero lower bound. Manuscript.

[20] 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.

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

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

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

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

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

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

[27] 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.

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

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

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

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

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

[33] Gavion, W.T., B.D.Keen, A.W.Richter, and N.A. Throckmorton, (2013). Global dynamics at the zero lower bound, Federal Reserve Board St. Louis working paper 2013-007B.

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

[35] Hughes, T., (1987). The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Prentice Hall, Englewood Cliff, NJ.

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

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

[38] 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.

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

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

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

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

[43] Judd, K., L. Maliar, S. Maliar and R. Valero, (2013). Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. NBER working paper 19326.

[44] 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.

[45] 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.

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

[47] 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.

[48] Kollmann, R., S. Maliar, B. Malin and P. Pichler, (2011b). Comparison of solutions to the multi-country real business cycle model. Journal of Economic Dynamics and Control 35, 186-202.

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

[50] Ma, X. and N. Zabaras, (2009). An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, Journal of Computational Physics 228, 3084–3113.

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

[52] Maliar, L. and S. Maliar, (2013). Numerical methods for large scale dynamic economic models. in: K. Schmedders and K. Judd (Eds.), Handbook of Computational Economics, Amsterdam: Elsevier Science, forthcoming.

[53] 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.

[54] 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.

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

[56] 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.

[57] 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.

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

[59] McGrattan, E., (1996). Solving the stochastic growth model with a finite element method. Journal of Economic Dynamics and Control 20, 19-42.

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

[61] Mertens, K. and M. Ravn, (2013). Fiscal policy in an expectations driven liquidity trap. Manuscript, Cornell University.

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

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

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

[65] Pichler, P., (2011). Solving the multi-country real business cycle model using a monomial rule Galerkin method. Journal of Economic Dynamics and Control 35, 240-251.

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

[67] Ravenna, F. and C. Walsh, (2011). Welfare-based optimal monetary policy with unemployment and sticky prices: a linear-quadratic framework. American Economic Journal: Macroeconomics 3, 130-162.

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

[69] Richter, A.W. and N.A. Throckmorton (2013). The zero lower bound: frequency, duration, and determinacy. Manuscript.

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

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

[72] 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.

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

[74] 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.

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

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

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

[78] Schmitt-Grohé, S. and M. Uribe, (2012). The making of a great contraction with a liquidity trap and a jobless recovery. Manuscript, Columbia University.

[79] 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.

[80] 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.

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

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

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

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

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

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

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

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

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

[90] Winschel, V. and M. Krätzig, (2010). Solving, estimating and selecting non-linear dynamic models without the curse of dimensionality. Econometrica 78(2), 803-821.

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

[92] Woodford, M. (2003). Interest and prices. Princeton University Press.

[93] Zhao, O. and M. Woodroofe, (2008). Law of the iterated logarithm for stationary processes. The Annals of Probability 36 (1), 127–142.

Appendix A. Properties of EDS grids

In this appendix, 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.

Appendix A1. Dispersion of points in EDS grids

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 8. *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*

$$d_n(P; X) = \sup_{x \in X} \inf_{1 \leq i \leq n} D(x, x_i), \tag{A1}$$

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 9. *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 1 *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 1 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 $\varepsilon \to 0$.

Appendix A2. Number of points in EDS grids

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 2 *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 1, 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. ■

Appendix A3. Discrepancy measures of EDS grids

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 Kiefer's (1961), 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 1 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 3 *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 2,

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

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

Appendix A4. Relation to mathematical literature

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

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

$$\mathcal{D}_M^*\left(P^\varepsilon;\mathcal{J}\right)\leq c\cdot d^{3/2}\left[\max\left\{\ln d,\ln M\right\}\right]^{1/2}M^{-1/2},\tag{A2}$$

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 (A2) 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.

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.[15] 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) discussed in Section 2.2.5.

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).

Our best- and worst-case scenarios

One-dimensional versions of our Propositions 2 and 3 have implications for Rényi's (1958) car parking model. Namely, Proposition 2 implies that the cars occupy between 50% and 100% of the roadside ($\frac{1}{2} \leq \lim\limits_{\varepsilon \to 0} M\varepsilon \leq 1$). 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, cars are parked to leave as little parking space to other drivers as possible, and in the latter case, cars are parked in a socially optimal 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 3 yields the worst-case scenario for discrepancy in Rényi's (1958) model, $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B}) \leq \frac{\sqrt{2}-1}{\sqrt{2}+1} \approx 0.17$, 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 2ε, and on $[\lambda^*, 1]$, all points are situated on the distance ε. 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 $\lim\limits_{\varepsilon \to 0} \left[\lambda^* - \frac{\frac{\lambda^*}{2\varepsilon}}{\frac{\lambda^*}{2\varepsilon} + \frac{1-\lambda^*}{\varepsilon}} \right] = \frac{\sqrt{2}-1}{\sqrt{2}+1} \approx 0.17$, which is the same value as implied by Proposition 3. To support this scenario, we assume that g has the Dirac point masses on the distances 2ε and ε in the intervals $[0, \lambda^*]$ and $[\lambda^*, 1]$, respectively.

When the dimensionality increases, our bounds become loose. Proposition 2 implies $\left(\frac{1}{2}\right)^d \leq \lim\limits_{\varepsilon \to 0} M\varepsilon^d \leq 1$ 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.

[15] 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.

Furthermore, when $d = 10$, Proposition 3 implies that $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B}) \leq \frac{\sqrt{2^{10}}-1}{\sqrt{2^{10}}+1} \approx 0.94$, which is almost uninformative since $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B}) \leq 1$ by definition. However, we cannot improve upon the general results of Propositions 2 and 3: our examples with Dirac point masses show that there exist density functions g under which the established bounds are attained.

Appendix B. Clustering algorithms

In this appendix, we describe two variants of clustering algorithms that we used for constructing grids: a hierarchical clustering algorithm and a K-means clustering algorithm. Clustering algorithms were used to produce all the results in Judd et al. (2010), which is an earlier version of the present paper.

Appendix B1. Hierarchical clustering algorithm

A hierarchical agglomerative clustering algorithm is described in Section 2.4. It begins from individual objects (observations) and agglomerates them iteratively into larger objects (clusters).

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 described in Section 2.4. The sample data contains 5 observations for 2 variables, x^1 and x^2:

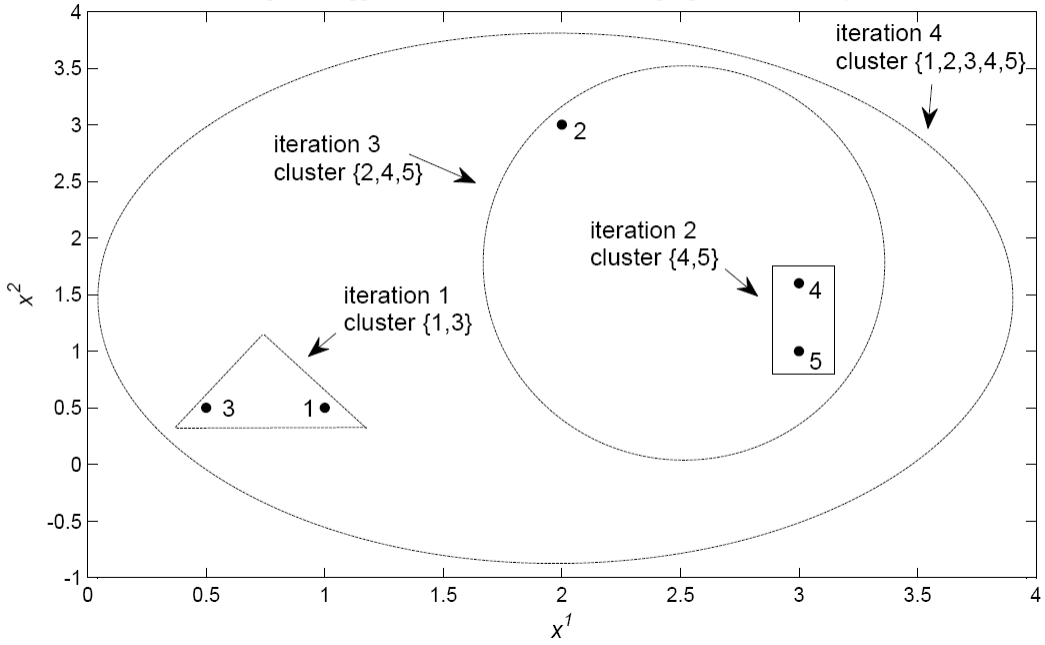
Observation	Variable x^1	Variable x^2
1	1	0.5
2	2	3
3	0.5	0.5
4	3	1.6
5	3	1

(B1)

We will consider two alternative measures of distance between clusters, the nearest-neighbor (or single) and Ward's ones. The nearest-neighbor distance measure is simpler to understand (because the distance between clusters can be inferred from the distance between observations without additional computations). However, the Ward's distance measure lead to somewhat more accurate results and thus, is our preferred choice for numerical analysis.

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 $\{1, 2, 3, 4, 5\}$. The computations performed under two distance measures are described in subsequent two sections.

Figure 9. Agglomerative hierarchical clustering algorithm: an example.



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 for (B1) 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,

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

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 (B2).[16] Such a measure shows how much the dispersion of observations changes when clusters $A \equiv \{x_1, ..., x_I\}$ and $B \equiv \{y_1, ..., y_J\}$ 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 $D(x_i, \overline{x})$ 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., $SSD(A) \equiv \sum_{i=1}^{I} [D(x_i, \overline{x})]^2$.

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

$$\widetilde{D}(A,B) = SSD(A \cup B) - [SSD(A) + SSD(B)]. \tag{B2}$$

[16] 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.

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

As an example, consider the distance between the singleton clusters 1 and 2 in (B1), 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,

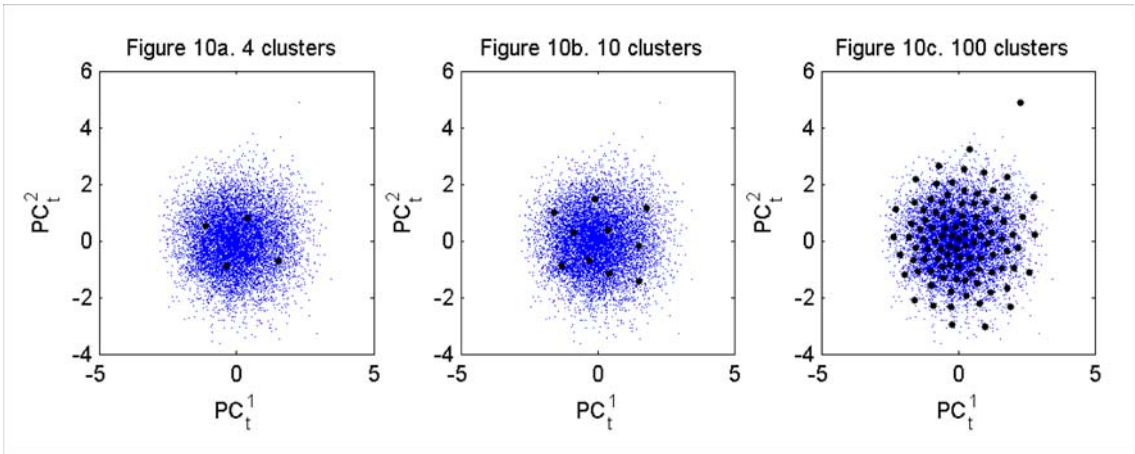
$$\widetilde{D}(A,B) = \frac{I \cdot J}{I+J} \sum_{\ell=1}^{\mathcal{L}} \left(\overline{x}^{\ell} - \overline{y}^{\ell}\right)^2,$$

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 tends to mimic the density of simulated points.



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:

(**Algorithm** K-means**): K-means clustering algorithm.**
Initialization. Choose M, the number of clusters to be created.
Generate randomly M clusters to obtain initial partition $\mathcal{P}^{(0)}$.
Step 1. On iteration i, for each cluster $A \in \mathcal{P}^{(i)}$, 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 $\mathcal{P}^{(i+1)}$.
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.

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. In the context of solution methods, the hierarchical and K-means clustering algorithms perform very similar.

Appendix C. One-agent model

In this section, we elaborate a description if the algorithms based on the EDS grid for solving the neoclassical growth model considered in Section 4.1.

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. To implement fixed-point iteration on $\widehat{K}$, we rewrite (8) in the form

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

In the true solution, k' on both sides of (C1) takes the same values and thus, cancels out. However, in the FPI iterative process, k' on two sides of (C1) takes different values, namely, we substitute $k' = \widehat{K}\left(\cdot; b\right)$ in the right side of (C1), and we compute the left side of (C1); fixed-point iteration 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 (7).

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 $\{k_{t+1}\}_{t=0,...,T}$.

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.

Computational choices. We parameterize the model (5)–(7) 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, we parameterize the capital equilibrium rule 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 a least-squares 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,

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

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 $\mathcal{R}\left(k_i, a_i\right)$.

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. Multi-agent model

In this section, we provide additional details about the solution procedure for the multi-agent model studied in Section 4.3. We parameterize the capital equilibrium rule of each country with a flexible functional form

$$K^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,\ldots,N}\right) \approx \widehat{K}^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,\ldots,N}; b^h\right), \tag{D1}$$

where b^h is a vector of coefficients. We rewrite the Euler equation (12) as

$$k_{t+1}^h = E_t\left\{\beta \frac{u'\left(c_{t+1}\right)}{u'\left(c_t\right)}\left[1-\delta+a_{t+1}^h A f'\left(k_{t+1}^h\right)\right] k_{t+1}^h\right\}. \tag{D2}$$

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 $\widehat{K}^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,\ldots,N}; b^h\right)$ is the best possible approximation of $K^h\left(\left\{k_t^h, a_t^h\right\}^{h=1,\ldots,N}\right)$ 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 $\left\{b^h\right\}^{h=1,\ldots,N}$, 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 with 2 nodes for each shock for $N < 12$, we use the monomial rule with $N^2 + 1$ points for N from $12 \leq N < 20$, and we use the monomial rule $2N$ points for $N \geq 20$; see Judd et al. (2011a) for a detailed description of these integration methods.

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 (13)–(17) 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 (14). 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 (18), (19) 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 (19), 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 (20)–(22) 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 (21). The derivative of the total cost in (20) 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

$$\begin{aligned}
P_t &= \left(\int_0^1 P_t(i)^{1-\varepsilon} \, di \right)^{\frac{1}{1-\varepsilon}} = \\
&\left[\int_{\text{reopt.}} P_t(i)^{1-\varepsilon} \, di + \int_{\text{non-reopt.}} P_t(i)^{1-\varepsilon} \, di \right]^{\frac{1}{1-\varepsilon}},
\end{aligned} \tag{E19}$$

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

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

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

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

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

We divide both sides of (E21) by P_t,

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

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 (14) with the government budget constraint (25), 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 (32) 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 (33)–(38) 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 (13); $\phi_y = 0.07$, $\phi_\pi = 2.21$, and $\mu = 0.82$ in the Taylor rule (39); $\varepsilon = 4.45$ in the production function of the final-good firm (19); $\theta = 0.83$ (the fraction of the intermediate-good firms affected by price stickiness); $\overline{G} = 0.23$ in the government budget constraint (25); 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 (15), (26) and (16). 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 (22), (17) and (28). We set the discount factor at $\beta = 0.99$. To parameterize the Taylor rule (39), 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 (32), (33) and (38), 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; see Judd et al. (2011a) for a detailed description of these integration formulas. 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.