Merging simulation and projection approaches to solve high-dimensional problems with an application to a new Keynesian model

Lilia Maliar
Stanford University

Serguei Maliar
Santa Clara University

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 ε-distinguishable set method, and a locally-adaptive variant of the ε-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.

Lilia Maliar: maliarl@stanford.edu
Serguei Maliar: smaliar@scu.edu

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 ε-distinguishable set method, and a locally-adaptive variant of the ε-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, for example, Marcet (1988), Smith (1993), Maliar and Maliar (2005), and Judd, Maliar, and Maliar (2011a). Second, it includes learning-based analysis; see, for example, 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, for example, Krusell and Smith (1998) and Weintraub, Benkard, and Van Roy (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, thus avoiding the cost of finding solutions in areas of the state space that are effectively never visited in equilibrium; see Judd, Maliar, and Maliar (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, for example, Wright and Williams (1984), Judd (1992), Christiano and Fisher (2000), Krueger and Kubler (2004), Aruoba, Fernández-Villaverde, and Rubio-Ramírez (2006), Anderson, Kim, and Yun (2010), Malin, Krueger, and Kubler (2011), Pichler (2011), and Judd, Maliar, Maliar, and Valero (2014). 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. So as 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.
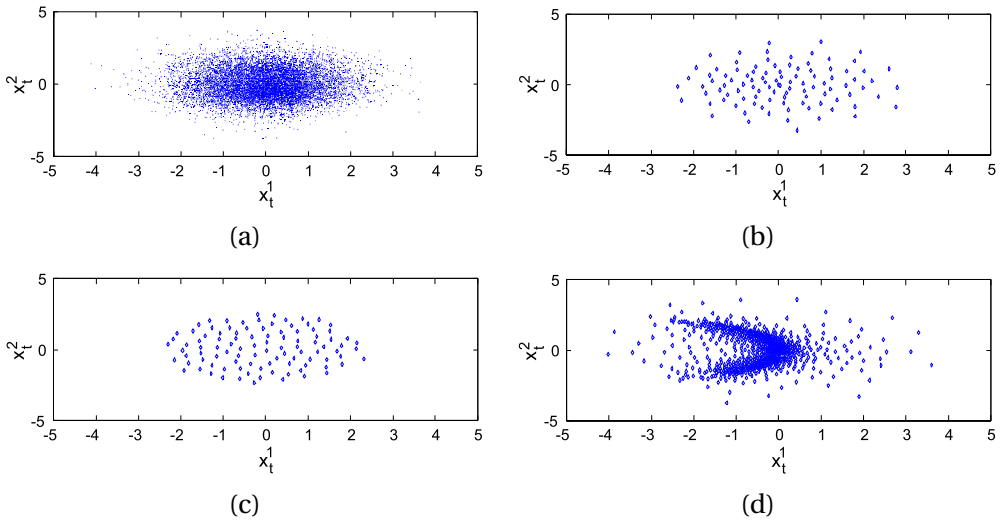
FIGURE 1.   (a) A set of random draws. (b) Cluster grid. (c) EDS grid. (d) Adaptive EDS grid.

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 the best features of stochastic simulation and projection methods, namely, it combines the adaptive geometry of stochastic simulation methods with the efficient discretization techniques of projection methods. As an example, in Figure 1(a), 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 Figure 1(b), (c), and (d), we distinguish three different subsets of the simulated points: we call them grids.

The grid shown in Figure 1(b) 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 1(c) 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 1(d), we show an example of a locally-adaptive EDS grid: instead of using a constant ε, we allow it to vary across the domain, that is, for each point (x_t^1, x_t^2), we have a different ε, that is, 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 1(a) is not an efficient choice: having many closely located grid points does not increase accuracy, but does increase the 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, that is, 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 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 that 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 a 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 less than 0.01%.

---

[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, Maliar, and Maliar (2012)), we also show applications of the EDS technique to dynamic programming.

Our second and more novel application is a new Keynesian model that includes a Taylor rule with a zero lower bound (ZLB) on nominal interest rates; see Woodford (2003) and Galí (2008). 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, 2007) and Del Negro, Schorfheide, Smets, and Wouters (2007). The EDS method is tractable for global polynomial approximations of degrees 2 and 3 (at least): the running time for solving the new Keynesian model is less than 25 minutes in all the cases considered; and for some versions of our method, the running time can be as low as 5 seconds at only moderate reductions in accuracy. 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 nonlinear solutions to medium- and large-scale new Keynesian models; see Judd, Maliar, and Maliar (2011b), Braun, Körber, and Waki (2012), Coibion, Gorodnichenko, and Wieland (2012), Fernández-Villaverde, Gordon, Guerrón-Quintana, and Rubio-Ramírez (2012), Gust, Lopez-Salido, and Smith (2012), Schmitt-Grohé and Uribe (2012), Aruoba and Schorfheide (2014), Gavion, Keen, Richter, and Throckmorton (2013), Mertens and Ravn (2013), and Richter and Throckmorton (2013). As is argued in Judd, Maliar, and Maliar (2011b), Fernández-Villaverde et al. (2012), and Braun, Körber, and Waki (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 (2014) 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, Maliar, and Maliar (2010, 2011b). Namely, to increase the accuracy

---

[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, Kim, and Yun (2010), and Adjemian and Juillard (2011).

of solutions in the ZLB area, first, they add grid points near the ZLB area using the actual data on the U.S. economy; second, they apply two piecewise bases to separately approximate the solution in the areas with active and nonactive ZLB.

Our locally-adaptive EDS methods are related to several other methods in the literature. First, the EDS technique with local bases has a 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 EDS technique with space-dependent ε resembles locally-adaptive sparse-grid techniques that refine an approximation by introducing new grid points and basis 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 (2014), who show the benefits of adaptive grid points and basis 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. An appendix and replication files are available in supplementary files on the journal website, http://qeconomics.org/supp/364/supplement.pdf and http://qeconomics.org/supp/364/code_and_data.zip.

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.

_____

[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, Fernández-Villaverde, and Rubio-Ramírez (2006), Den Haan (2010), Kollmann, Maliar, Malin, and Pichler (2011), and Maliar and Maliar (2014).

2.1 *A class of stochastic processes*

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

$$x_{t+1} = \varphi(x_t, \epsilon_{t+1}), \quad t = 0, 1, \ldots, \tag{1}$$

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

Many dynamic economic models have equilibrium laws of motion for state variables that can be represented by a stochastic system in the form (1). For example, the standard neoclassical growth model, described in Section 4, has the laws of motion for capital and productivity that are given by $k_{t+1} = K(k_t, a_t)$ 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.

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

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

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

DEFINITION 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 $\mathcal{A} \subset \mathcal{A}^*$.

DEFINITION 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). The function $\mathcal{P}(x, \mathcal{A})$ is the probability that stochastic system (1) whose state today is $x_t = x$ and 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 exist 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.

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

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

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 semicontinuous but may be not lower semicontinuous (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 $x_1, \ldots, x_n \subseteq \mathbb{R}^d$ 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 $\prod_{i=1}^d [-\infty, v_i)$, 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 logarithms 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 $x_1, \ldots, x_n \in X \subseteq \mathbb{R}^d$.

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 such method, namely, a multivariate kernel algorithm with a normal kernel that estimates the density function in a point x as

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

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 us to compute pairwise distances between all the sample points. Finally, in Step 3, we do not choose the density cutoff η, but choose a fraction of the sample to be removed, δ, that 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 that has the lowest density.

2.2.2 *An ε-distinguishable set*   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 A.3.

DEFINITION 7. *Let (X, D) be a bounded metric space. A set P^ε consisting of points $x_1^\varepsilon, \ldots, x_M^\varepsilon \in X \subseteq \mathbb{R}^d$ is called ε-distinguishable if $D(x_i^\varepsilon, x_j^\varepsilon) > \varepsilon$ 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 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 $x_1, \ldots, x_n \in X \subseteq \mathbb{R}^d$. 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 $O(nM)$, 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 interpoint 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 $(n-1) + (n-2) + \cdots + (n-M) = \sum_{i=1}^{M}(n-i) = nM - \frac{M(M+1)}{2} \leq nM$. When no points are eliminated from P, that is, $M = n$, 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 A.2. 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 require 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 $x_i \equiv (x_i^1, \ldots, x_i^d)$ be an observation $i = 1$ (there are n observations) and let $x^\ell \equiv (x_1^\ell, \ldots, x_n^\ell)^\top$ be a variable ℓ (there are d variables), that is, $X = (x^1, \ldots, x^d) = (x_1, \ldots, x_n)^\top$. We first compute the singular value decomposition of X, that is, $X = UQV^\top$, 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 $\mathrm{PC} \equiv XV$. The variables $\mathrm{PC} = (\mathrm{PC}^1, \ldots, \mathrm{PC}^d) \in \mathbb{R}^{n \times d}$ are called *principal components* (PCs) of X and are orthogonal (uncorrelated), that is, $(\mathrm{PC}^{\ell'})^\top \mathrm{PC}^\ell = 0$ 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, $D(x_i, x_j) = [\sum_{\ell=1}^{d}(\mathrm{PC}_i^\ell - \mathrm{PC}_j^\ell)^2]^{1/2}$, where all principal components $\mathrm{PC}^1, \ldots, \mathrm{PC}^d$ 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 2(a). We orthogonalize the data using the principal component (PC) transformation, and we normalize the PCs to unit variance; see Figure 2(b). 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 2(c). We construct an EDS; see Figure 2(d). We plot such a set in the PC and original coordinates in Figure 2(e) and (f), 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]
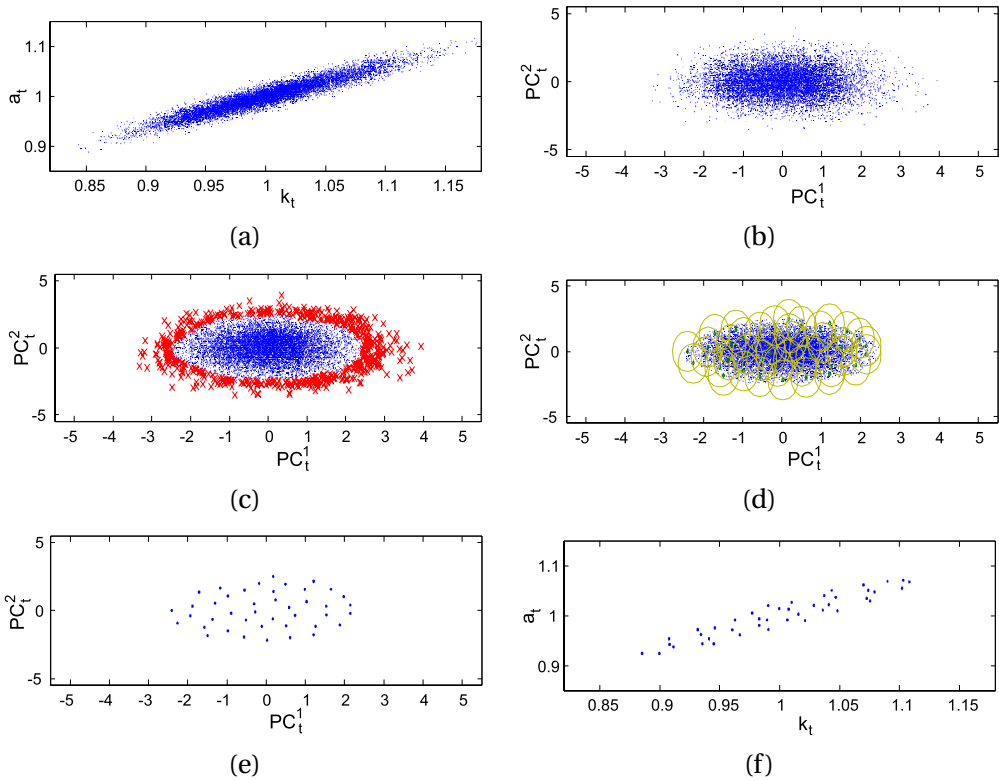
FIGURE 2. (a) Simulated points. (b) Principal components (PCs). (c) Density levels on PCs. (d) Constructing EDS. (e) EDS on PCs. (f) EDS on original data.

[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.

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 grid points in general. We address this question in Appendices A.1 and A.3; 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, Eichelbacker, Schreiber, and Yukich (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 nonoverlap 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 A.1). 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 A.3). 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 for 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 nonuniformity 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 A.3.

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 A.2; 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 $M(\varepsilon_{\max}^{(1)}) \leq \overline{M} \leq M(\varepsilon_{\min}^{(1)})$.

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 $\varepsilon_{\min}^{(i+1)} = \varepsilon$; 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 A.2 (see Proposition 2), namely, we set $\varepsilon_{\max}^{(1)} = 0.5 r_{\max} \overline{M}^{-1/d}$ and $\varepsilon_{\min}^{(1)} = r_{\min} (\overline{M}^{1/d} - 1)^{-1}$, 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 $r_{\min}$ and $r_{\max}$ 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 $O(n^2)$. This is because the estimation of the density function in Step 3 of Algorithm $\mathcal{A}^\eta$ has a complexity $O(n^2)$, and the construction of an EDS set in Step 1 of Algorithm P^ε has a complexity $O(nM)$. (The latter is significantly lower than the former if $M \ll n$.) The complexity $O(n^2)$ 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^ε-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 $x_1, \ldots, x_n \in X \subseteq \mathbb{R}^d$.

Step 3. Select an EDS P^ε of M points, $x_1^\varepsilon, \ldots, x_M^\varepsilon$ 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 that has the lowest density.

To control the fraction of the sample removed, we use the estimated density function $\widehat{g}$. Note that when eliminating a point $x_j^\varepsilon \in P^\varepsilon$, we remove $\frac{\widehat{g}(x_j^\varepsilon)}{\sum_{i=1}^{M} \widehat{g}(x_i^\varepsilon)}$ of the original sample. We therefore can proceed with eliminations of points from the EDS one by one until their cumulative mass reaches the target value of δ.
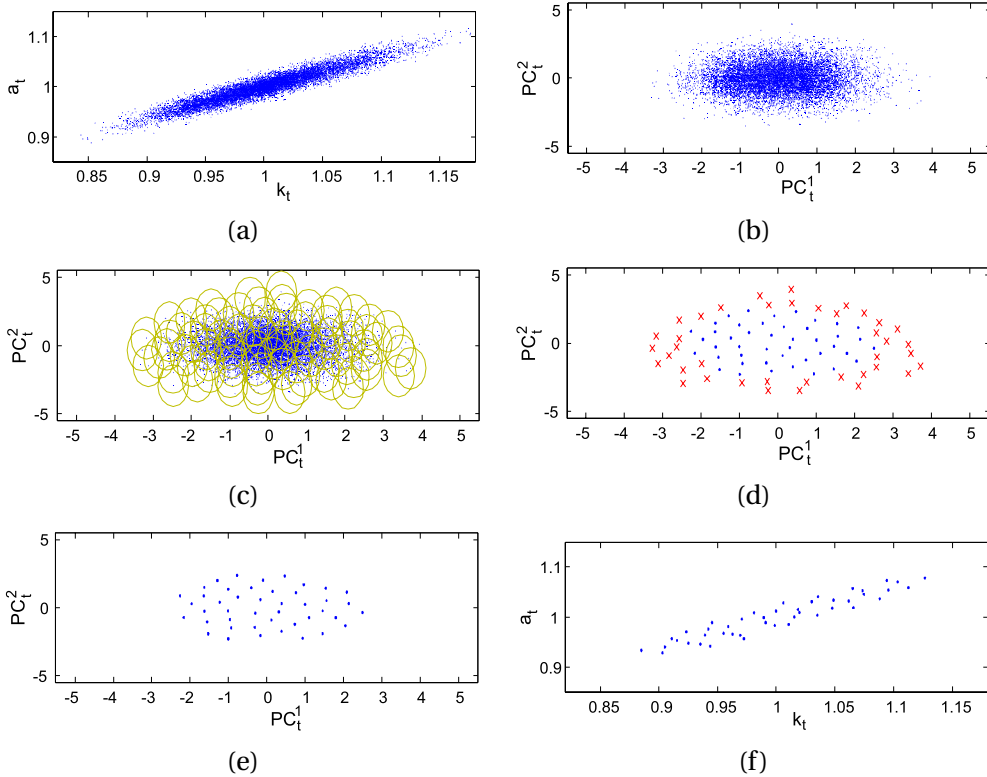
FIGURE 3. (a) Simulated points. (b) Principal components (PCs). (c) EDS on PCs. (d) Density levels on PCs. (e) EDS after removing low-probability points. (f) EDS on original data.

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 Figure 3(a)–(f).

We first compute the normalized PCs of the original sample; see Figure 3(b) (this step in the same as in Figure 2(b)). We compute an EDS P^ε on the normalized PCs; see Figure 3(c). 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 3(d). The resulting EDS is shown in Figure 3(e). Finally, we plot the EDS grid in the original coordinates in Figure 3(f).

2.4 Cluster-grid technique

We have described one specific EDS procedure for forming a discrete approximation to the essentially ergodic set of the 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, Landau, Leese, and Stahl (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: agglomerative hierarchical and K-means. The steps of an agglomerative hierarchical method are shown below, and a K-means method is described in Appendix B.2.[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 that is closest to the cluster's center.

As a measure of distance between two clusters, we use Ward's measure of distance; see Appendix B.1. In Figure 4(a)–(c), we show an example in which we partition a set of simulated points into four clusters and construct four 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 $O(n^3)$ and $O(n^{dM+1} \log n)$ for the agglomerative hierarchical and K-means algorithms, respectively. Also, the properties of grids produced by clustering methods are hard to characterize analytically.
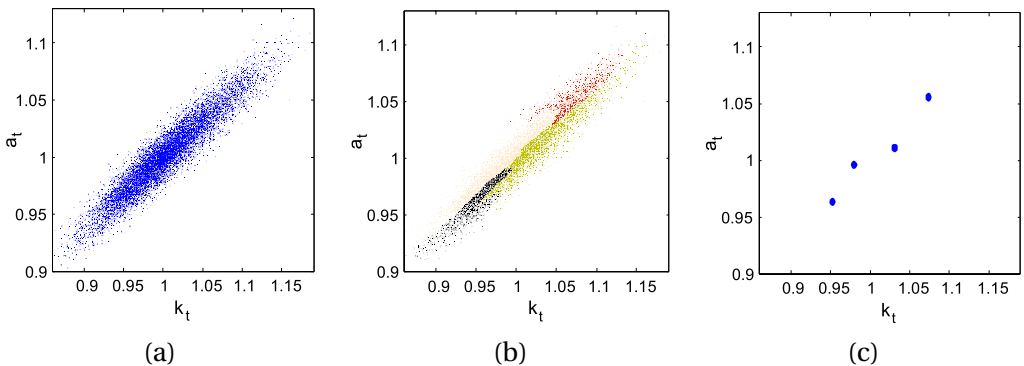

FIGURE 4. (a) Simulated points. (b) Four clusters. (c) Clusters grid.

---

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

As 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.

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 priori 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 denser grid points in X_1 than in the rest of the domain; an example of this construction is shown in Figure 1(d). However, in the typical case, it is not a priori 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 $\varepsilon_1, \ldots, \varepsilon_n$ for a given set $x_1, \ldots, x_n \in X \subseteq \mathbb{R}^d$ (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 $\mathcal{R}(x_i) = \|\widehat{f}(x_i) - f(x_i)\|$ 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 a 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 basis 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 nonlinearities. 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 func-

tional 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 $x_1^\varepsilon, \ldots, x_m^\varepsilon \in P^\varepsilon$ 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 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, Maliar, and Maliar (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, Maliar, and Maliar (2011a) for a discussion. However, global polynomial approximations may not be sufficiently flexible to accurately approximate decision functions with strong nonlinearities 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 a small neighborhood of a given EDS grid point. A global approximation is obtained by combining 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 5(b); see Niederreiter (1992) for a review of low-discrepancy methods. Thus, we recompute 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 recompute it in the existing grid points. This useful feature is specific to approximations with local basis functions; for global approximations, we need to recompute 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. Figure 5(a)–(d) plots four 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 (1997), Krueger and Kubler (2004), and Pichler (2011), respectively).[7]
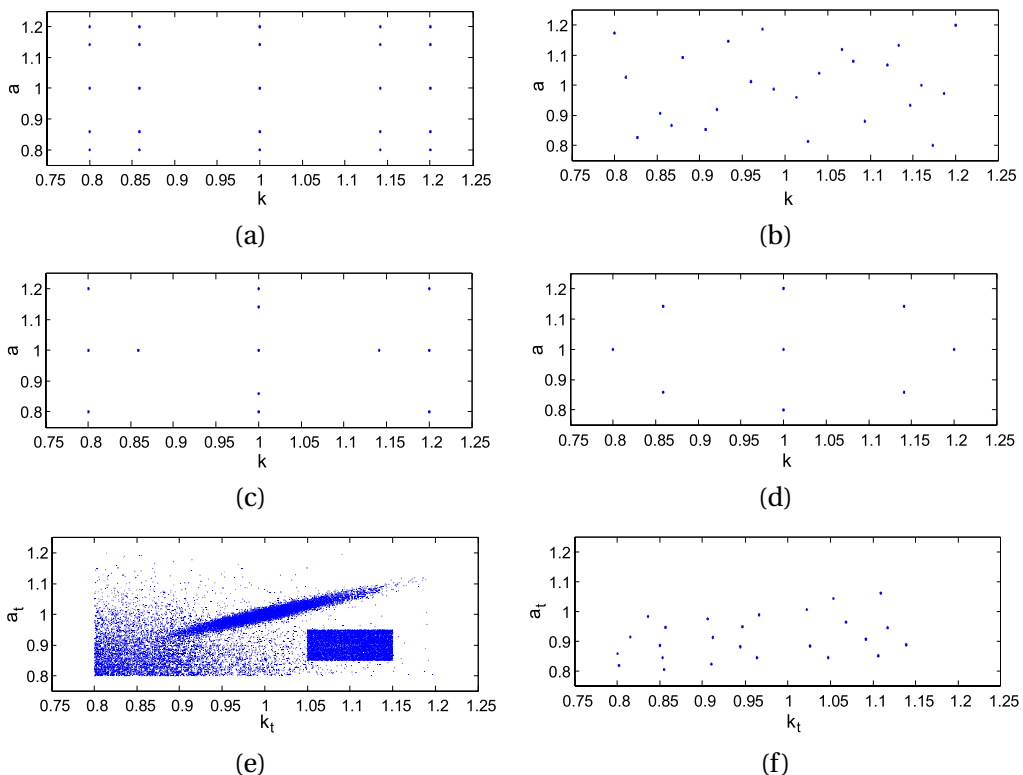

Figure 5. (a) Tensor-product Chebyshev grid. (b) Sobol grid. (c) Smolyak grid. (d) Monomial grid. (e) Simulated points. (f) EDS on simulated points.

[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

In turn, stochastic simulation methods use the adaptive geometry; see Marcet (1988), Smith (1993), Maliar and Maliar (2005), and Judd, Maliar, and Maliar (2011a) for examples of methods that 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 enormous cost savings 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 that used by stochastic simulation methods) with an efficient discretization (similar to that produced by low-discrepancy methods on a hypercube). In Figure 5(e), we show an example of a cloud of simulated points of irregular shape, and in Figure 5(f), 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 not be a good choice. First, focusing on a high-probability set may not have advantages relative 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 cost savings 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 (2014) 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 that do not naturally belong to a high-probability set of the studied

---

a Markov process with a finite-state Markov chain. This discretization technique requires the distribution function of the Markov process to be specified 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 the distribution function to be known. It can be applied to both exogenous and endogenous variables.

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

model. Finally, our worst case analysis in Appendix A.3 shows that there are scenarios in which EDSs constructed on simulated data are highly nonuniform. However, these scenarios require extreme assumptions about the density of simulated points, for example, 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.

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, but the EDS method can be also used to solve dynamic programming problems; see Judd, Maliar, and Maliar (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 $t = 0, 1, \ldots, \infty$,

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

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

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 nonstate variables—prices, consumption, labor supply, etcetera (also called non-predetermined variables), G is a continuously differentiable vector function, and $\epsilon_{t+1} \in \mathbb{R}^p$ is a vector of shocks. A solution is given by a set of equilibrium functions $s_{t+1} = S(s_t, z_t)$ and $y_t = Y(s_t, z_t)$ that satisfy (3) and (4) in the relevant area of the state space. In terms of the notation in 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 $\{\epsilon_{t+1}\}_{t=0,\ldots,T-1}$. Compute and fix $\{z_{t+1}\}_{t=0,\ldots,T-1}$ 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, $j = 1, \ldots, J$.

Step 1. Construction of an EDS grid.

   a. Use $\widehat{S}(\cdot; b^s)$ to simulate $\{s_{t+1}\}_{t=0,\ldots,T-1}$.

   b. Construct an EDS grid, $\Gamma \equiv \{s_m, z_m\}_{m=1,\ldots,M}$.

Step 2. Computation of a solution on EDS grid using a projection method.

   a. For $m = 1, \ldots, M$, construct residuals

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

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

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

Iterate on Steps 1 and 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 nonexplosive simulated series. For many problems in economics, linear solutions can be used as an initial guess; they are sufficiently accurate, numerically stable and readily available from automated perturbation software (we use Dynare solutions; see Adjemian, Bastani, Juillard, Mihoubi, Perendia, Ratto, and Villemot (2011)). Finding a sufficiently good initial guess can be a nontrivial 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 A.1, 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 the 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 Santos and Peralta-Alva (2005). Therefore, we are left to check that the EDS grid constructed on the simulated series also converges. Let $\Gamma' \equiv \{x_i'\}_{i=1,\ldots,M'}$ and $\Gamma'' \equiv \{x_j''\}_{j=1,\ldots,M''}$ be the EDS grids constructed on two different sets of simulated points. Our criteria of convergence is $\sup_{x_j'' \in \Gamma''} \inf_{x_i' \in \Gamma'} D(x_i', x_j'') < 2\varepsilon$. That is, each grid point of Γ'' has a grid point of Γ' at a distance smaller than 2ε (this is the maximum distance between the grid points on the essentially ergodic set; see Proposition 3 in Appendix A.3).

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 parameter 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 nonlinear solutions as accurate as the one constructed using highly accurate nonlinear solutions. Furthermore, we found that small changes in the model parameters do not require us to recompute the grid. In the presence of kinks, such as the ZLB in the 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 the Monte Carlo method in accuracy by orders of magnitude in the context of the studied class of models; see Judd, Maliar, and Maliar (2011a) for comparison results.[9] The cost of Gaussian product rules is prohibitive in high-dimensional problems, but monomial formulas are tractable even in

---

[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 $\overline{y} \sim \mathcal{N}(0, \frac{\sigma_y}{\sqrt{n}})$. If $\sigma_y = 1\%$ and $n = 10{,}000$, we have

models with hundreds of state variables; see Judd, Maliar, and Maliar (2011b) for a description of these formulas.

Solving systems of nonlinear equations: The convergence issue   In Step 2 of the EDS algorithm, we must find the parameter 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 $n = M \times H$ nonlinear 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 $n' = n$, that is, if 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 weighted sum of residuals in the model's equations (this case is similar to regression in econometrics).

A variety of numerical methods in the literature can be used to solve a system of non-linear equations in Step 2 of the EDS algorithm; see, for example, 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), and 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.

ALGORITHM 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 $\widehat{b} = \Psi(b^{(i)})$.

Step 2. If $\|\widehat{b} - b^{(i)}\| < \varpi$, 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 the cost of Newton-style methods does. As other nonlinear solvers, fixed-point iteration may fail to converge. The following example, borrowed from Judd (1998, p. 159), illustrates the possibility of nonconvergence.

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

---

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.

to a solution. However, we can also rewrite this equation as $x = x^3 - 1$ and construct a sequence $x^{(i+1)} = (x^{(i)})^3 - 1$, starting from $x^{(0)} = 1$, which diverges to $-\infty$.

This example shows that whether fixed-point iteration succeeds in finding a solution may depend on the 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 those 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; see Feng, Miao, Peralta-Alva, and Santos (2014).

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 ALGORITHM (Residuals in Equilibrium Conditions).

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

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

where $y_\tau = \widehat{Y}(s_\tau, z_\tau; b^y)$, $s'_\tau = \widehat{S}(s_\tau, z_\tau; b^s)$, $z'_{\tau,j} = Z(z_\tau, \epsilon_j^{\text{test}})$, $y'_{\tau,j} = \widehat{Y}(y_\tau, z'_{\tau,j}; b^y)$, 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 unacceptable, we modify the choices made in the EDS algorithm (i.e., simulation length, number of grid points, approximating functions, integration method) and recompute the solution. In the paper, we evaluate the accuracy on a set of simulated points. This new set of points is different from that used in the solution procedure: it is constructed under a different sequence of shocks (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 setup*

The representative agent solves

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

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

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

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; $\beta \in (0, 1)$; $\delta \in (0, 1]$; $A > 0$; $\rho \in (-1, 1)$; $\sigma \geq 0$; and 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, for example, Stokey, Lucas, and Prescott (1989, p. 392). For numerical experiments, we use $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$ with $\gamma \in \{\frac{1}{5}, 1, 5\}$ 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 by k', which yields $\widehat{k'} = \beta k' E[\frac{u'(c')}{u'(c)}(1 - \delta + a' A f'(k'))]$. 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.

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

TABLE 1. Accuracy and speed of the Euler equation EDS algorithm in the one-agent model.

| 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
First	−4.74	−3.81	25.5	−4.29	−3.31	24.7	−3.29	−2.35	23.6
Second	−6.35	−5.26	1.8	−5.94	−4.87	0.8	−4.77	−3.60	0.4
Third	−7.93	−6.50	1.9	−7.26	−6.04	0.9	−5.97	−4.47	0.4
Fourth	−9.37	−7.60	2.0	−8.65	−7.32	0.9	−7.05	−5.26	0.4
Fifth	−9.82	−8.60	14.25	−9.47	−8.24	5.5	−7.89	−6.46	2.8

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

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 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,
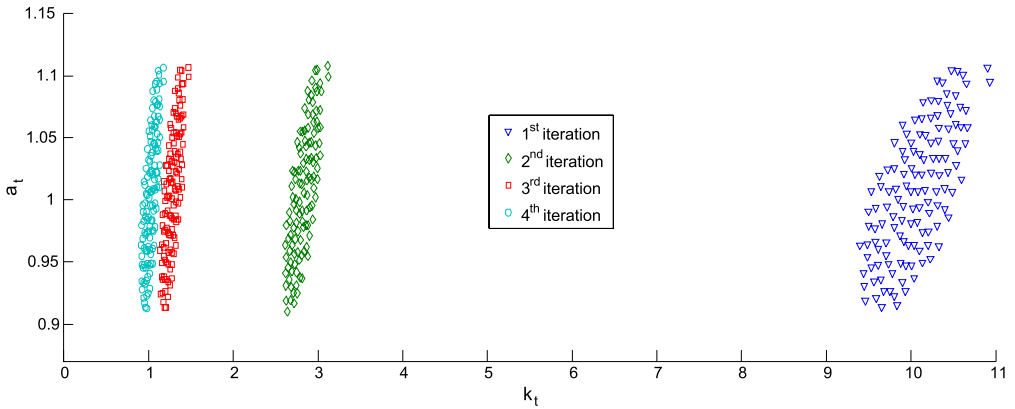
FIGURE 6. Convergence of the EDS grid starting from capital series normalized to 10 steady state levels.

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, Krueger, and Kubler (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 that studied in Judd et al. (2014), and differs from that in Krueger and Kubler (2004) and Malin, Krueger, and Kubler (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 four 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, Krueger, and Kubler (2011), we use intervals [0.8, 1.2] and $[\exp(-\frac{0.8}{1-\rho}),$ $\exp(\frac{0.8}{1-\rho})]$ for capital and productivity level, respectively. The Smolyak grid has 13 points (see Figure 5(c)), 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 evaluate the accuracy of solutions not only on a stochastic simulation, but also on a set of 100×100 points that are uniformly spaced on the same domain as those 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 poly-

TABLE 2. Accuracy and speed in the one-agent model: Smolyak grid versus EDS grid.

Polynomial Degree	Smolyak Grid					Accuracy on a Hypercube				
	Simulation		Rectangular			Simulation		Rectangular		
	L_1	L_∞	L_1	L_∞	CPU	L_1	L_∞	L_1	L_∞	CPU
First	−3.31	−2.94	−3.25	−2.54	0.39	−4.23	−3.31	−3.26	−2.38	9.89
Second	−4.74	−4.17	−4.32	−3.80	0.20	−5.89	−4.87	−4.41	−3.25	0.19
Third	−5.27	−5.13	−5.39	−4.78	0.22	−7.19	−6.16	−5.44	−4.11	0.17

Note: L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log 10 units) on a stochastic simulation of 10,000 observations and on a fixed grid of 100×100 points covering the rectangle with the intervals [0.8, 1.2] and [$\exp(-\frac{0.8}{1-\rho})$, $\exp(\frac{0.8}{1-\rho})$] for capital and productivity, respectively; CPU is the time necessary for computing a solution (in seconds).

nomial 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.

The setup There are N agents, interpreted as countries, that 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),

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

---

[10]Kollmann et al. (2011) compare the accuracy of solutions produced by several solution methods, including the cluster grid algorithm (CGA) introduced in the earlier version of the present paper and the Smolyak algorithm of Krueger and Kubler (2004) (see Maliar, Maliar, and Judd (2011) and Malin, Krueger, and Kubler (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.

subject to the aggregate resource constraint

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

where $\{k_0^h, a_0^h\}^{h=1,\dots,N}$ 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 $h \in \{1, \dots, N\}$; $\beta \in (0,1)$ is the discount factor; $\delta \in (0,1]$ is the depreciation rate; and 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 $\varsigma_{t+1}^h \sim \mathcal{N}(0, \sigma^2)$ 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,

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

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 parameter values 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), that is, $k_{t+1}^h = K^h(\{k_t^h, a_t^h\}^{h=1,\dots,N})$, $h = 1, \dots, N$. 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 used in the representative-agent case; see Appendix D for details of the computational procedure.

The choice of the integration method plays an important role in the accuracy and speed of our solution algorithm. The Monte Carlo method produces large sampling errors that dominate the overall accuracy of solutions. Quadrature product rules are accurate but their cost is prohibitive if the number of shocks is large. However, nonproduct

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, Maliar, and Maliar (2011a) for a detailed description of this and other integration methods, and see Judd, Maliar, and Maliar (2011a, 2012) and Maliar, Maliar, and Judd (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 memory congestion. If a solution method relies on product rules in constructing a grid, integration nodes, and 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.

The accuracy of solutions here is similar to what 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, Maliar, and Maliar (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, Maliar, and Maliar (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 a one-node quadrature inte-

TABLE 3. Accuracy and speed of the EDS algorithm in the multi-agent model.

Polynomial Degree	$N = 2$			$N = 20$			$N = 40$			$N = 200$		
	$\overline{M} = 300$, $2N$ Nodes			$\overline{M} = 1000$, $2N$ Nodes			$\overline{M} = 4000$, 1-Node			$\overline{M} = 1000$, 1-Node		
	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
First	-4.70	-3.17	0.7	-4.76	-3.05	21	-4.79	-3.09	89	-4.66	-2.90	105
Second	-6.01	-4.06	1.9	-5.88	-4.14	282	-5.48	-4.13	1463			

Note: L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log 10 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; $2N$ and 1-node denote the monomial rule with two nodes and one-node Gauss–Hermite integration rules, respectively.

gration 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 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, Eichenbaum, and Evans (2005), Smets and Wouters (2003, 2007), and 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), Schmitt-Grohé and Uribe (2007), and Ravenna and Walsh (2011)) and cubic approximations (e.g., Rudebusch and Swanson (2008)). The earlier literature that used global solution methods includes Adam and Billi (2006), Anderson, Kim, and Yun (2010), and Adjemian and Juillard (2011). The above studies either have 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, Maliar, and Maliar (2011b), Fernández-Villaverde et al. (2012), and Braun, Körber, and Waki (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 (2014) find that new Keynesian models may have multiple equilibria in the presence of ZLB; in particular, the last paper accurately computes such multiple equilibria with a full set of stochastic shocks. In

---

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

[13]In particular, Adam and Billi (2006) linearize all the first-order conditions except for the nonnegativity constraint for nominal interest rates, and Adjemian and Juillard (2011) assume perfect foresight to implement an extended path method of Fair and Taylor (1983).

the present paper, we focus on the conventional equilibrium in which target inflation coincides with actual inflation in the steady state.

5.1 *The setup*

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

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

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

where $(B_0, \eta_{u,0}, \eta_{L,0}, \eta_{B,0})$ 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; and $\gamma > 0$ and $\vartheta > 0$ are the utility-function parameters. The processes for shocks are

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

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

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

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

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

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

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

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),

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

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

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

where $L_t(i)$ is the labor input, $\exp(\eta_{a,t})$ 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 re-optimizing firm $i \in [0, 1]$ maximizes the current value of profit over the time when $\widetilde{P}_t$ remains effective,

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

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

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), and mc_{t+j} is the real marginal cost of output at time $t + j$ (which is identical across the firms).

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

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

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

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

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

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

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,

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

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 planner's problem

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

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

where $G_t \equiv \frac{\overline{G} Y_t}{\exp(\eta_{G,t})}$ 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 first-order conditions (FOCs) of the problem (29)–(30) imply that $Y_{N,t}$ depends only on exogenous shocks,

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

5.2 *Summary of equilibrium conditions*

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

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

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

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

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

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

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

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

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

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 $t+1$; and Δ_t is a measure of price dispersion across firms (also referred to as efficiency distortion). The conditions (32)–(38) correspond, respectively, to (E.17), (E.18), (E.23), (E.33), and (E.3) in Appendix E.

An interior equilibrium is described by eight equilibrium conditions (32)–(39) and six processes for exogenous shocks, (15)–(17), (22), (28), and (26). The system of equations must be solved with respect to eight unknowns $\{C_t, Y_t, L_t, \pi_t, \Delta_t, R_t, S_t, F_t\}$. There are two endogenous and six exogenous state variables, $\{\Delta_{t-1}, R_{t-1}\}$ and $\{\eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{a,t}, \eta_{R,t}, \eta_{G,t}\}$, 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 $S_t = S(x_t)$, $F_t = F(x_t)$, and $C_t^{-\gamma} = \mathrm{MU}(x_t)$ with $x_t = \{\Delta_{t-1}, R_{t-1}, \eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{a,t}, \eta_{R,t}, \eta_{G,t}\}$ using the Euler equations (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 $(R_t, \exp(\eta_{a,t}))$. We see that many points happen to be on the bound $R_t = 1$ and that the essentially ergodic set in the two figures is shaped roughly as a circle.

Accuracy and cost of solutions   Two parameters that play a special role in our analysis are the volatility of labor shocks σ_L and the target inflation rate π_*. Concerning σ_L, Del Negro et al. (2007) finds that shocks to labor must be as large as 40% to match the data, namely, they estimate the interval $\sigma_L \in [0.1821, 0.6408]$ with an average of $\sigma_L = 0.4054$. Concerning π_*, Del Negro et al. (2007) estimate the interval $\pi_* \in [1.0461, 1.0738]$ with an 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
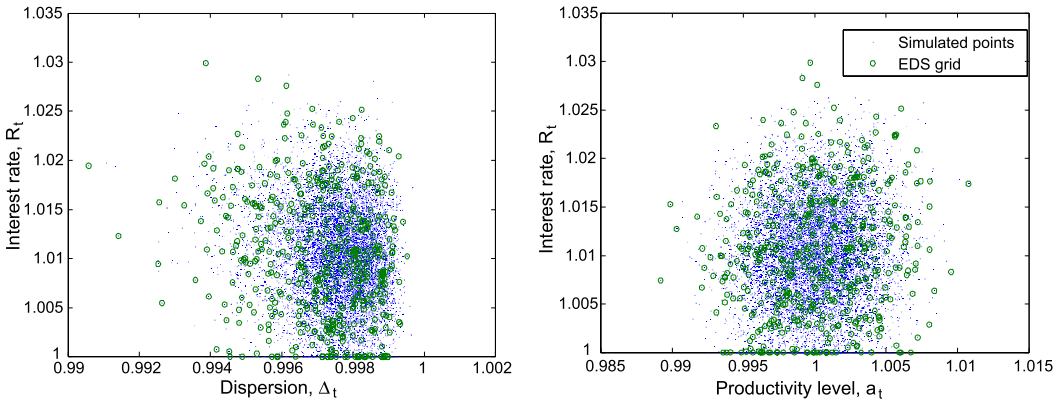
FIGURE 7. Simulated points and the grid for a new Keynesian model: ZLB is imposed.

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.

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 la-
bor 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 approxi-
mation. The accuracy of the EDS solutions also decreases, but does so 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 $\sigma_L = 0.1821$

---

[14]Chung, Laforte, Reifschneider, and Williams (2011) provide estimates of the incidence of the ZLB in the
U.S. economy. Christiano, Eichenbaum, and Rebelo (2011) 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.

	$\pi_* = 1$ 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
$M(\varepsilon)$	–	–	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($R \leq 1$), %	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

Note: 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); $M(\varepsilon)$ is the realized number of points in the EDS grid (the target number of grid points is $\overline{M} = 500$); PER1 and PER2 are the 1st- and 2nd-order perturbation solutions, respectively; EDS2 and EDS3 are 2nd- and 3rd-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($R \leq 1$) is a percentage number of periods in which $R \leq 1$; dif(X), % is the maximum absolute percentage difference between time series for variable X produced by the method in the given column and EDS3.

of labor shocks assumed in the first experiment.[15] 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   Figure 8(a) and (b) 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
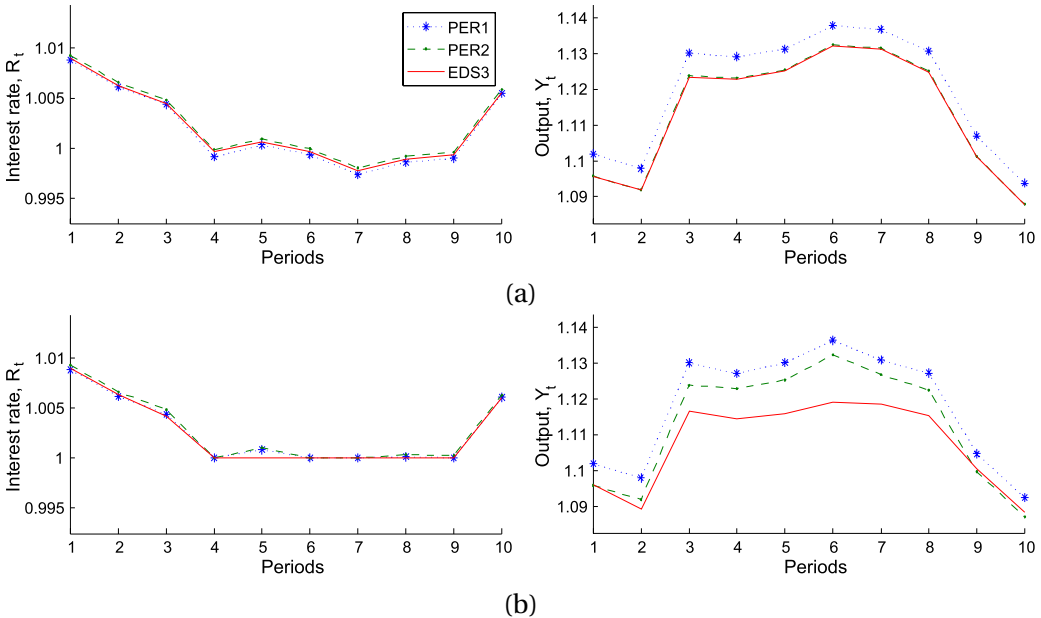

FIGURE 8. (a) A time-series solution to a new Keynesian model: ZLB is not imposed. (b) A time-series solution to a new Keynesian model: ZLB is imposed.

[15]In a model with an active ZLB, we find the convergence to be slow and fragile. This fact is possibly related to the finding of Benhabib, Schmitt-Grohé, and Uribe (2001a, 2001b) that a deterministic version of the model has multiple trajectories converging to a liquidity trap, in addition to a locally unique equilibrium that converges to a target inflation level.

and the EDS methods predict five periods of negative (net) interest rates (see periods 4 and 6–9 in Figure 8(a)). When the ZLB is imposed, the EDS methods, EDS2 and EDS3, predict a zero interest rate in those five periods, while the perturbation methods, PER1 and PER2, predict a zero interest rate in just three periods (see periods 4, 6, and 7 in Figure 8(b)).

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 five periods of a zero net interest rate (which correspond to five 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 the accuracy much. 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; 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, $x_1^\varepsilon, \ldots, x_M^\varepsilon$, using a constant value of ε. We then construct an eight-dimensional hypercube with a 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, 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

TABLE 5. Accuracy and speed of the EDS algorithm with local bases and locally-adaptive grid in the new Keynesian model.

	$\overline{M} = 2$			$\overline{M} = 10$			$\overline{M} = 100$			$\overline{M} = 500$		
	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

Note: L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log 10 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.

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, that is, $\varepsilon' = 2\overline{\varepsilon}$ 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 larger 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 at similar accuracy levels. In fact, for large values of $\overline{M}$, the EDS method with an evenly spaced grid, LB, slightly overperforms the locally-adaptive EDS method, LB-LA. Indeed, when we have to select just two areas, the adaptive way of selecting these areas 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 (2014). To increase the accuracy of solutions in the ZLB area, Aruoba and Schorfheide (2014) implement two modifications to the baseline cluster-grid algorithm studied in Judd, Maliar, and Maliar (2010, 2011b): first, they add grid points near the ZLB area using the actual data on the U.S. economy; second, they use two piecewise local bases to separately approximate the solution in the areas with active and nonactive 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; finally, there is a kink in the equilibrium rules due to the ZLB. We chose this application so as 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. There are versions of our method that are much cheaper. For example, if we construct a grid from a randomly chosen subset of simulated points and do not update the grid along iterations, the running time can be as low as 5 seconds for producing a second-degree nonlinear solution to the new Keynesian model with eight state variables. However, the accuracy of such a method will also be somewhat lower (these experiments are not reported). The EDS method would be tractable in much larger applications, as our results for the multicountry 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 basis 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 nonlinearities. 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 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 further increased by using more powerful hardware and software as well as parallelization techniques. Relatively low computational expense makes the EDS algorithm an ideal candidate for applications in which the equilibrium decision functions must be constructed a large number of times such as nested fixed point methods for econometric estimation, see, e.g., Fernández-Villaverde and Rubio-Ramírez (2007) and extended function path (EFP) framework proposed in Maliar, Maliar, Taylor, and Tsener (2014) for analyzing nonstationary and unbalanced growth models.

<center>REFERENCES</center>

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. [5, 31]

Adda, J. and R. Cooper (2003), *Dynamic Economics: Quantitative Methods and Applications*. MIT Press, Cambridge, Massachusetts and London, England. [6]

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. [21]

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, 31]

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. [2, 5, 31]

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. [2, 6]

Aruoba, S. and F. Schorfheide (2014), "Macroeconomic dynamics near ZLB: A tale of two countries." PIER Working Paper 14-035. [5, 6, 19, 31, 40]

Baryshnikov, Y., P. Eichelbacker, T. Schreiber, 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. [12]

Benhabib, J., S. Schmitt-Grohé, and M. Uribe (2001a), "Monetary policy and multiple equilibria." *American Economic Review*, 91 (1), 167–186. [38]

Benhabib, J., S. Schmitt-Grohé, and M. Uribe (2001b), "The perils of Taylor rules." *Journal of Economic Theory*, 96, 40–69. [38]

Bertsekas, D. and J. Tsitsiklis (1996), *Neuro-Dynamic Programming*. Optimization and Neural Computation Series. Athena Scientific, Belmont, Massachusetts. [2, 21]

Braun, R. A., L. M. Körber, and Y. Waki (2012), "Some unpleasant properties of log-linearized solutions when the nominal rate is zero." Working paper, Federal Reserve Bank of Atlanta. [5, 31]

Brumm, J. and S. Scheidegger (2013), "Using adaptive sparse grids to solve high-dimensional dynamic models." Manuscript, University of Zurich. [6, 16]

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. [31]

Christiano, L., M. Eichenbaum, and S. Rebelo (2011), "When is the government spending multiplier large?" *Journal of Political Economy*, 119 (1), 78–121. [36]

Christiano, L. and D. Fisher (2000), "Algorithms for solving dynamic models with occasionally binding constraints." *Journal of Economic Dynamics and Control*, 24, 1179–1232. [2, 6, 31]

Chung, H., J.-P. Laforte, D. Reifschneider, and J. Williams (2011), "Have we underestimated the probability of hitting the zero lower bound?" Working Paper 2011-01, Federal Reserve Bank of San Francisco. [36]

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. [5]

Del Negro, M., F. Schorfheide, F. Smets, and R. Wouters (2007), "On the fit of new Keynesian models." *Journal of Business & Economic Statistics*, 25 (2), 123–143. [5, 31, 35]

Den Haan, W. (1990), "The optimal inflation path in a Sidrauski-type model with uncertainty." *Journal of Monetary Economics*, 25, 389–409. [23]

Den Haan, W. (2010), "Comparison of solutions to the incomplete markets model with aggregate uncertainty." *Journal of Economic Dynamics and Control*, 34, 4–27. [6]

Den Haan, W. and A. Marcet (1990), "Solving the stochastic growth model by parameterized expectations." *Journal of Business & Economic Statistics*, 8, 31–34. [19]

Den Haan, W. and A. Marcet (1994), "Accuracy in simulations." *Review of Economic Studies*, 61, 3–18. [24]

Everitt, B., S. Landau, M. Leese, and D. Stahl (2011), *Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley, Chichester. [14]

Fair, R. and J. Taylor (1983), "Solution and maximum likelihood estimation of dynamic non-linear rational expectation models." *Econometrica*, 51, 1169–1185. [31]

Feng, Z., J. Miao, A. Peralta-Alva, and M. Santos (2014), "Numerical simulation of nonoptimal dynamic equilibrium models." *International Economic Review*, 55 (1), 83–110. [24]

Fernández-Villaverde, J., G. Gordon, P. Guerrón-Quintana, and J. Rubio-Ramírez (2012), "Nonlinear adventures at the zero lower bound." Working Paper 18058, NBER. [5, 31]

Fernández-Villaverde, J. and J. Rubio-Ramírez (2007), "Estimating macroeconomic models: A likelihood approach." *Review of Economic Studies*, 74, 1059–1087. [6, 41]

Galí, J. (2008), *Monetary Policy, Inflation and the Business Cycles: An Introduction to the New Keynesian Framework*. Princeton University Press, Princeton, New Jersey. [5, 32]

Gaspar, J. and K. Judd (1997), "Solving large-scale rational-expectations models." *Macroeconomic Dynamics*, 1, 45–75. [6, 23]

Gavion, W. T., B. D. Keen, A. W. Richter, and N. A. Throckmorton (2013), "Global dynamics at the zero lower bound." Working Paper 2013-007B, Federal Reserve Board St. Louis. [5]

Gust, C., D. Lopez-Salido, and M. E. Smith (2012), "The empirical implications of the interest-rate lower bound." Manuscript, Federal Reserve Board. [5]

Hughes, T. (1987), *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Prentice Hall, Englewood Cliff, New Jersey. [6, 17]

Judd, K. (1992), "Projection methods for solving aggregate growth models." *Journal of Economic Theory*, 58, 410–452. [2, 18, 24, 26]

Judd, K. (1998), *Numerical Methods in Economics.* MIT Press, Cambridge, Massachusetts. [6, 23, 24]

Judd, K., L. Maliar, and S. Maliar (2010), "A cluster grid projection method: Solving problems with high dimensionality." Working Paper 15965, NBER. [5, 15, 30, 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. [2, 17, 19, 22, 30]

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. [5, 15, 23, 31, 40]

Judd, K., L. Maliar, and S. Maliar (2012), "Merging simulation and projection approaches to solve high-dimensional problems." Working Paper 18501, NBER. [4, 20, 30]

Judd, K., L. Maliar, S. Maliar, and R. Valero (2014), "Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain." *Journal of Economic Dynamics and Control*, 44, 92–123. [2, 27]

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. [9]

Kollmann, R. (2002), "Monetary policy rules in the open economy: Effects on welfare and business cycles." *Journal of Monetary Economics*, 49, 989–1015. [5, 31]

Kollmann, R., S. Maliar, B. Malin, and P. Pichler (2011), "Comparison of solutions to the multi-country real business cycle model." *Journal of Economic Dynamics and Control*, 35, 186–202. [6, 28]

Krueger, D. and F. Kubler (2004), "Computing equilibrium in OLG models with production." *Journal of Economic Dynamics and Control*, 28, 1411–1436. [2, 18, 27, 28]

Krusell, P. and A. Smith (1998), "Income and wealth heterogeneity in the macroeconomy." *Journal of Political Economy*, 106, 868–896. [2]

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. [6, 16]

Maliar, L. and S. Maliar (2005), "Solving non-linear stochastic growth models: Iterating on value function by simulations." *Economics Letters*, 87, 135–140. [2, 19]

Maliar, L. and S. Maliar (2014), "Numerical methods for large scale dynamic economic models." In *Handbook of Computational Economics*, Vol. 3 (K. Schmedders and K. Judd, eds.), 325–477, Elsevier Science, Amsterdam. [6]

Maliar, S., L. Maliar, and K. Judd (2011), "Solving the multi-country real business cycle model using ergodic set methods." *Journal of Economic Dynamics and Control*, 35, 207–228. [28, 30]

Maliar, S., L. Maliar, J. Taylor, and I. Tsener (2014), "A tractable framework for analyzing nonstationary and unbalanced growth models." Manuscript. [41]

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. [2, 27, 28]

Marcet, A. (1988), "Solving non-linear models by parameterizing expectations." Unpublished manuscript, Carnegie Mellon University, Graduate School of Industrial Administration. [2, 19, 23]

Marcet, A. and G. Lorenzoni (1999), "The parameterized expectation approach: Some practical issues." In *Computational Methods for Study of Dynamic Economies* (R. Marimon and A. Scott, eds.), 143–171, Oxford University Press, New York. [19]

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. [2]

Marimon, R. and A. Scott (1999), *Computational Methods for Study of Dynamic Economies*. Oxford University Press, New York. [6]

McGrattan, E. (1996), "Solving the stochastic growth model with a finite element method." *Journal of Economic Dynamics and Control*, 20, 19–42. [6, 17]

Mertens, K. and M. Ravn (2011), "Credit channels in a liquidity trap." Discussion Paper 8322, CEPR. [36]

Mertens, K. and M. Ravn (2013), "Fiscal policy in an expectations driven liquidity trap." Manuscript, Cornell University. [5, 31]

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania. [4, 17]

Pakes, A. and P. McGuire (2001), "Stochastic algorithms, symmetric Markov perfect equilibria, and the 'curse' of dimensionality." *Econometrica*, 69, 1261–1281. [2]

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. [2, 18]

Powell, W. (2011), *Approximate Dynamic Programming*. Wiley, Hoboken, New Jersey. [2]

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. [31]

Rényi, A. (1958), "On a one-dimensional space-filling problem." *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 3, 109–127. [12]

Richter, A. W. and N. A. Throckmorton (2013), "The zero lower bound: Frequency, duration, and determinacy." Manuscript. [5]

Rudebusch, G. and E. Swanson (2008), "Examining the bond premium puzzle with a DSGE model." *Journal of Monetary Economics*, 55, S111–S126. [5, 31]

Rust, J. (1997), "Using randomization to break the curse of dimensionality." *Econometrica*, 65, 487–516. [18]

Santos, M. (1999), "Numerical solution of dynamic economic models." In *Handbook of Macroeconomics* (J. Taylor and M. Woodford, eds.), 312–382, Elsevier Science, Amsterdam. [6]

Santos, M. (2000), "Accuracy of numerical solutions using the Euler equation residuals." *Econometrica*, 68, 1377–1402. [24]

Santos, M. S. and A. Peralta-Alva (2005), "Accuracy of simulations for stochastic dynamic models." *Econometrica*, 73, 1939–1976. [22]

Schmitt-Grohé, S. and M. Uribe (2007), "Optimal simple and implementable monetary fiscal rules." *Journal of Monetary Economics*, 54, 1702–1725. [5, 31]

Schmitt-Grohé, S. and M. Uribe (2012), "The making of a great contraction with a liquidity trap and a jobless recovery." Manuscript, Columbia University. [5, 31]

Scott, D. and S. Sain (2005), "Multidimensional density estimation." In *Handbook of Statistics*, Vol. 24 (C. Rao, E. Wegman, and J. Solka, eds.), 229–261, Elsevier B. V., Amsterdam. [9]

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. [5, 31, 35, 36]

Smets, F. and R. Wouters (2007), "Shocks and frictions in US business cycles: A Bayesian DSGE approach." *American Economic Review*, 97 (3), 586–606. [5, 31, 35]

Smith, A. (1993), "Estimating non-linear time-series models using simulated vector autoregressions." *Journal of Applied Econometrics*, 8, S63–S84. [2, 19]

Stachursky, J. (2009), *Economic Dynamics: Theory and Computations*. MIT Press, Cambridge, Massachusetts. [7]

Stokey, N. L., R. E. Lucas Jr., and E. C. Prescott (1989), *Recursive Methods in Economic Dynamics*. Harvard University Press, Cambridge, Massachusetts. [7, 25]

Tauchen, G. and R. Hussey (1991), "Quadrature-based methods for obtaining approximate solutions to non-linear asset pricing models." *Econometrica*, 59, 371–396. [6, 18]

Taylor, J. (1993), "Discretion versus policy rules in practice." *Carnegie-Rochester Conference Series on Public Policy*, 39, 195–214. [31]

Taylor, J. and H. Uhlig (1990), "Solving non-linear stochastic growth models: A comparison of alternative solution methods." *Journal of Business & Economic Statistics*, 8, 1–17. [6]

Temlyakov, V. (2011), *Greedy Approximation*. Cambridge University Press, Cambridge. [10, 12]

Weintraub, G. Y., C. L. Benkard, and R. Van Roy (2008), "Markov perfect industry dynamics with many firms." *Econometrica*, 76 (6), 1375–1411. [2]

Winschel, V. and M. Krätzig (2010), "Solving, estimating and selecting non-linear dynamic models without the curse of dimensionality." *Econometrica*, 78 (2), 803–821. [6]

Wright, B. and J. Williams (1984), "The welfare effects of the introduction of storage." *Quarterly Journal of Economics*, 99, 169–192. [2, 23]

Woodford, M. (2003), *Interest and Prices*. Princeton University Press, Princeton, New Jersey. [5]

Zhao, O. and M. Woodroofe (2008), "Law of the iterated logarithm for stationary processes." *The Annals of Probability*, 36 (1), 127–142. [9]