

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

(*Quantitative Economics*, Vol. 6, No. 1, March 2015, 1–47)

LILIA MALIAR
Stanford University

SERGUEI MALIAR
Santa Clara University

APPENDIX A: PROPERTIES OF EDS GRIDS

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

A.1 Dispersion of points in EDS grids

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

DEFINITION 8. Let P be a set consisting of points $x_1, \dots, x_n \in X \subseteq \mathbb{R}^d$, and let (X, D) be a bounded metric space. The dispersion of P in X is given by

$$d_n(P; X) = \sup_{x \in X} \inf_{1 \leq i \leq n} D(x, x_i), \quad (\text{A.1})$$

where D is a (Euclidean) metric on X .

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

DEFINITION 9. Let S be a sequence of elements on X , and let $x_1, \dots, x_n \in X \subseteq \mathbb{R}^d$ be the first n terms of S . The sequence S is called low dispersion if $\lim_{n \rightarrow \infty} d_n(S; X) = 0$.

Lilia Maliar: maliar1@stanford.edu

Serguei Maliar: smaliar@scu.edu

Copyright © 2015 Lilia Maliar and Serguei Maliar. Licensed under the Creative Commons Attribution-NonCommercial License 3.0. Available at <http://www.qeconomics.org>.

DOI: 10.3982/QE364

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

PROPOSITION 1. *Let P be any set of n points $x_1, \dots, x_n \in X \subseteq \mathbb{R}^d$ with a dispersion $d_n(P; X) < \varepsilon$. Let (X, D) be a bounded metric space, and let P^ε be an EDS $x_1^\varepsilon, \dots, x_M^\varepsilon$ constructed by Algorithm P^ε . Then the dispersion of P^ε is bounded by $\varepsilon < d_M(P^\varepsilon; X) < 2\varepsilon$.*

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

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

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

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

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

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

Proposition 1 states that any $x \in X$ has a neighbor $x_j^\varepsilon \in P^\varepsilon$ that is situated at most at distance 2ε . The dispersion of points in an EDS goes to zero as $\varepsilon \rightarrow 0$.

A.2 Number of points in EDS grids

The number of points in an EDS is unknown a priori. It depends on the value of ε and the order in which the points from P are processed. Temlyakov (2011, Theorem 3.3 and Corollary 3.4) provides the bounds on the number of points in a specific class of EDSs, namely, those that cover a unit ball with balls of radius ε . We also confine our attention to a set X given by a ball; however, in our case, balls of radius ε do not provide a covering of X . The bounds for our case are established in the following proposition.

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

PROOF. To prove the first inequality, notice that, by Proposition 1, the balls $B(x_1^\varepsilon; 2\varepsilon), \dots, B(x_M^\varepsilon; 2\varepsilon)$ cover X . Hence, we have $M\lambda_d(2\varepsilon)^d \geq \lambda_d r^d$, where λ_d is the volume of a d -dimensional unit ball. This gives the first inequality.

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

A.3 Discrepancy measures of EDS grids

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

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

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

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

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

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

If a sequence is low discrepancy, then it is also low dispersion; see Niederreiter (1992, Theorem 6.6). Indeed, a sequence that covers X uniformly must become increasingly dense everywhere on X as $n \rightarrow \infty$. However, the converse is not true. A sequence that becomes increasingly dense on X as $n \rightarrow \infty$ does not need to become increasingly uniform since density may be distributed unevenly. Thus, the result of Proposition 1 that the

dispersion of an EDS converges to zero in the limit does not mean that its discrepancy does so.

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

DEFINITION 12. The spherical discrepancy $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B})$ is defined as the discrepancy of P over the family J generated by the intersection of d -dimensional open balls $B(0; r)$ centered at 0 with radius $r \leq 1$.

PROPOSITION 3. Let P be any set of n points $x_1, \dots, x_n \in B(0; 1) \subseteq \mathbb{R}^d$ with a dispersion $d_n(P; X) < \varepsilon$. Then the discrepancy of an EDS constructed by Algorithm P^ε under \mathcal{B} is bounded by $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B}) \leq \frac{\sqrt{2^d-1}}{\sqrt{2^d+1}}$.

PROOF. Let $\lambda \equiv \lambda(B(0; r)) = \lambda(B(0; 1))r^d = r^d$ be a Lebesgue measure of $B(0; r)$, and let $\frac{C(P^\varepsilon; B(0; r))}{M}$ be the fraction of points from P^ε in the ball $B(0; r)$. Consider the case when $\lambda(B(0; r)) \geq \frac{C(P^\varepsilon; B(0; r))}{M}$ and let us compute the maximum discrepancy $\mathcal{D}_M^{\max}(P^\varepsilon; \mathcal{B})$ across all possible EDSs using the results of Proposition 2,

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

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

A.4 Relation to mathematical literature

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

Existence results for a covering-number problem Temlyakov (2011) studies the problem of finding a covering number—a minimum number of balls of radius ε that cover a given compact set (such as a d -dimensional hypercube or hypersphere). In particular,

he shows the following result. There exists an EDS P^ε on a unit hypercube $[0, 1]^d$ whose star discrepancy is bounded by

$$\mathcal{D}_M^*(P^\varepsilon; \mathcal{J}) \leq c \cdot d^{3/2} [\max\{\ln d, \ln M\}]^{1/2} M^{-1/2}, \quad (\text{A.2})$$

where c is a constant; see Temlyakov (2011, Proposition 6.72). The discrepancy of such an EDS converges to 0 as $M \rightarrow \infty$ (i.e., $\varepsilon \rightarrow 0$). However, constructing an EDS with the property (A.2) is operationally difficult and costly. Also, Temlyakov (2011) selects points from a compact subset of \mathbb{R}^d , and his analysis cannot be directly applied to our problem of finding an ε -distinguishable subset of a given finite set of points.

Probabilistic results for random sequential packing problems Probabilistic analysis of an EDS is nontrivial because points in such a set are spatially dependent: once we place a point in an EDS, it affects the placement of all subsequent points.

Some related probabilistic results are obtained in the literature on a random sequential packing problem.¹ Consider a bounded set $X \subseteq \mathbb{R}^d$ and a sequence of d -dimensional balls whose centers are independent and identically distributed (i.i.d.) random vectors $x_1, \dots, x_n \in X$ with a given density function g . A ball is packed if and only if it does not overlap with any ball that has already been packed. If not packed, the ball is discarded. At saturation, the centers of accepted balls constitute an EDS. A well known unidimensional example of this general problem is the car-parking model of Rényi (1958) discussed in Section 2.2.5.

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

Our best- and worst-case scenarios One-dimensional versions of our Propositions 2 and 3 have implications for Rényi's (1958) car-parking model. Namely, Proposition 2 implies that the cars occupy between 50% and 100% of the roadside ($\frac{1}{2} \leq \lim_{\varepsilon \rightarrow 0} M\varepsilon \leq 1$). These are the best- and worst-case scenarios in which the cars are parked at distances ε and 0 from each other, respectively. (In the former case, cars are parked to leave as little parking space for other drivers as possible, and in the latter case, cars are parked in a socially optimal way.) The density functions that support the worst and best scenarios are those that contain the Dirac point masses at distances 2ε and ε , respectively.

Proposition 3 yields the worst-case scenario for discrepancy in Rényi's (1958) model, $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B}) \leq \frac{\sqrt{2}-1}{\sqrt{2}+1} \approx 0.17$, which is obtained under $\lambda^* = \frac{\sqrt{2}}{\sqrt{2}+1}$. This bound is attainable. Indeed, consider an EDS on $[0, 1]$ such that on the interval $[0, \lambda^*]$, all points are

¹This problem arises in spatial birth–growth models, germ–grain models, percolation models, spatial-graphs models; see, for example, Baryshnikov, Eichelbacher, Schreiber, and Yukich (2008) for a review.

situated at a distance 2ε , and on $[\lambda^*, 1]$, all points are situated on the distance ε . In the first interval, we have $\frac{\lambda^*}{2\varepsilon} \leq M \leq \frac{\lambda^*}{2\varepsilon} + 1$ points and in the second interval, we have $\frac{1-\lambda^*}{\varepsilon} \leq M \leq \frac{1-\lambda^*}{\varepsilon} + 1$ points. On the first interval, the limiting discrepancy is

$$\lim_{\varepsilon \rightarrow 0} \left[\lambda^* - \frac{\frac{\lambda^*}{2\varepsilon}}{\frac{\lambda^*}{2\varepsilon} + \frac{1-\lambda^*}{\varepsilon}} \right] = \frac{\sqrt{2}-1}{\sqrt{2}+1} \approx 0.17,$$

which is the same value as implied by Proposition 3. To support this scenario, we assume that g has Dirac point masses at distances 2ε and ε in the intervals $[0, \lambda^*]$ and $[\lambda^*, 1]$, respectively.

When the dimensionality increases, our bounds become loose. Proposition 2 implies $(\frac{1}{2})^d \leq \lim_{\varepsilon \rightarrow 0} M \varepsilon^d \leq 1$, which means that M can differ by a factor of 2^d under the worst- and best-case scenarios; for example, when $d = 10$, M can differ by a factor of 1024. Furthermore, when $d = 10$, Proposition 3 implies that $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B}) \leq \frac{\sqrt{2^{10}}-1}{\sqrt{2^{10}}+1} \approx 0.94$, which is almost uninformative since $\mathcal{D}_M^s(P^\varepsilon; \mathcal{B}) \leq 1$ by definition. However, we cannot improve on the general results of Propositions 2 and 3: our examples with Dirac point masses show that there exist density functions g under which the established bounds are attained.

APPENDIX B: CLUSTERING ALGORITHMS

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

B.1 Hierarchical clustering algorithm

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

A numerical example of implementing the agglomerative hierarchical clustering algorithm We provide a numerical example that illustrates the construction of clusters under the agglomerative hierarchical algorithm described in Section 2.4. The sample data contain five observations for two variables, x^1 and x^2 , see Table B.1. We will consider two alternative measures of distance between clusters: nearest neighbor (or single) and Ward's. The nearest-neighbor distance measure is simpler to understand (because the distance between clusters can be inferred from the distance between observations without additional computations). However, Ward's distance measure leads to somewhat more accurate results and, thus, is our preferred choice for numerical analysis.

Both measures lead to an identical set of clusters shown in Figure 9. On iteration 1, we merge observations 1 and 3 into a cluster $\{1, 3\}$; on iteration 2, we merge observations 4 and 5 into a cluster $\{4, 5\}$; on iteration 3, we merge observations 2 and $\{4, 5\}$ into a

TABLE B.1.

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

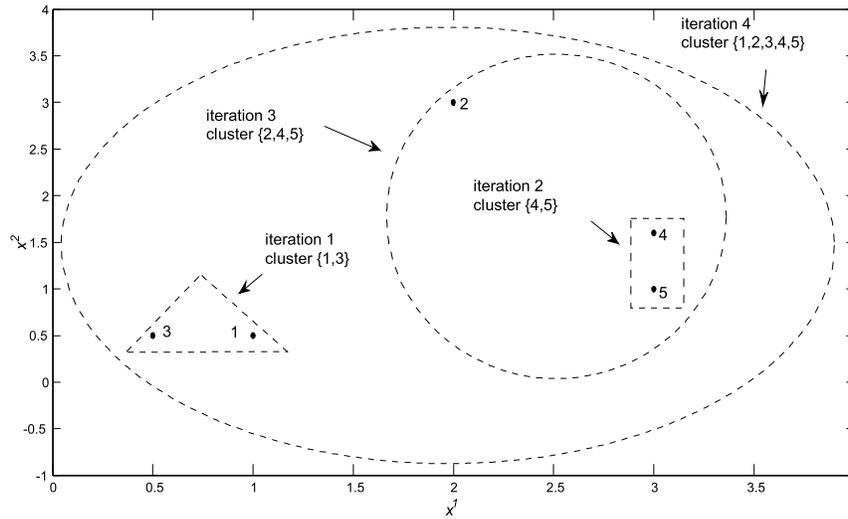


FIGURE 9. Agglomerative hierarchical clustering algorithm: an example.

cluster $\{2, 4, 5\}$; and finally, on iteration 4, we merge clusters $\{1, 3\}$ and $\{2, 4, 5\}$ into one cluster that contains all observations $\{1, 2, 3, 4, 5\}$. The computations performed under two distance measures are described in the subsequent two sections.

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

Let us compute a matrix of distances for Table B.1 between singleton clusters in which each entry ij corresponds to D_{ij} :

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

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

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

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

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

where $\tilde{D}(\{1, 3\}, 2) = \min\{D_{12}, D_{32}\} = 2.7$, $\tilde{D}(\{4, 5\}, 2) = \min\{D_{42}, D_{52}\} = 1.7$, and $\tilde{D}(\{1, 3\}, \{4, 5\}) = \min\{D_{14}, D_{15}, D_{34}, D_{35}\} = 2.1$. Hence, the smallest nonzero distance in S_3 is $\tilde{D}(\{4, 5\}, 2)$, so we merge clusters 2 and $\{4, 5\}$ into a cluster $\{2, 4, 5\}$. The only two unmerged clusters left are $\{1, 3\}$ and $\{2, 4, 5\}$, so that the last step is to merge those two to obtain the cluster $\{1, 2, 3, 4, 5\}$. The procedure of constructing clusters is summarized in Table B.2.

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

Ward's measure of distance We now construct clusters using Ward's measure of distance (B.1).² Such a measure shows how much the dispersion of observations changes when clusters $A \equiv \{x_1, \dots, x_I\}$ and $B \equiv \{y_1, \dots, y_J\}$ are merged together compared to the case when A and B are separate clusters.

TABLE B.2.

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

²If a measure of distance between groups of observations does not fulfill the triangular inequality, it is not a distance in the conventional sense and is referred to in the literature as dissimilarity.

Formally, we proceed as follows:

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

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

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

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

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

$$\tilde{D}(A, B) = \text{SSD}(A \cup B) - [\text{SSD}(A) + \text{SSD}(B)]. \quad (\text{B.1})$$

This measure is known to lead to spherical clusters of a similar size; see, for example, Everitt, Landau, Leese, and Stahl (2011, p. 79), which is in line with our goal of constructing a uniformly spaced grid that covers the essentially ergodic set. In our experiments, Ward's measure yielded somewhat more accurate solutions than the other measures of distance considered, such as the nearest neighbor, furthest neighbor, and group average; see, for example, Romesburg (1984) and Everitt et al. (2011) for reviews.

As an example, consider the distance between the singleton clusters 1 and 2 in Table B.1, that is, $\tilde{D}(1, 2)$. The center of the cluster $\{1, 2\}$ is $\bar{x}_{\{1,2\}} = (\bar{x}_{\{1,2\}}^1, \bar{x}_{\{1,2\}}^2) = (1.5, 1.75)$, and $\text{SSD}(1) = \text{SSD}(2) = 0$. Thus, we have

$$\begin{aligned} \tilde{D}(1, 2) &= \text{SSD}(\{1, 2\}) \\ &= (1 - 1.5)^2 + (2 - 1.5)^2 + (0.5 - 1.75)^2 + (3 - 1.75)^2 = 3.625. \end{aligned}$$

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

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

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

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

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

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

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

We have

$$\begin{aligned} \text{SSD}(\{1, 3\}) &= (1 - 0.75)^2 + (0.5 - 0.75)^2 + (0.5 - 0.5)^2 + (0.5 - 0.5)^2 = 0.125, \\ \text{SSD}(\{1, 2, 3\}) &= (1 - 7/6)^2 + (2 - 7/6)^2 + (0.5 - 7/6)^2 \\ &\quad + (0.5 - 4/3)^2 + (3 - 4/3)^2 + (0.5 - 4/3)^2 \\ &= 16/3, \end{aligned}$$

and $\text{SSD}(2) = 0$. Thus, we obtain

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

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

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

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

On iteration 3, the matrix of distances is

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

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

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

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

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

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

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

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

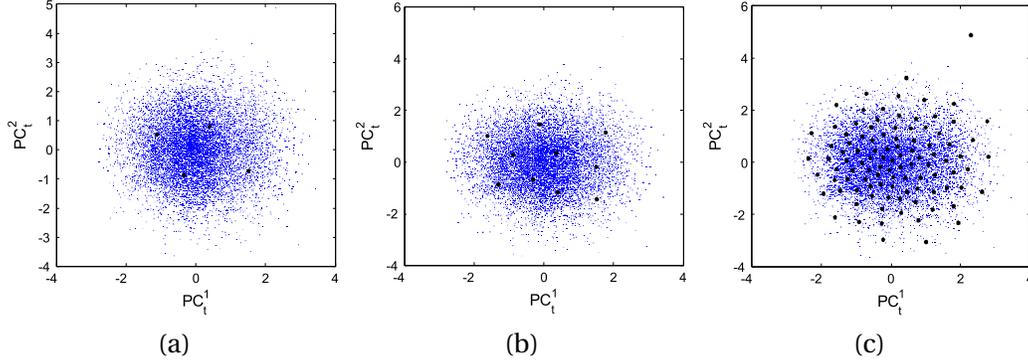


FIGURE 10. (a) 4 clusters. (b) 10 clusters. (c) 100 clusters.

B.2 K -means clustering algorithm

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

ALGORITHM K -MEANS (K -Means Clustering Algorithm).

Initialization. Choose M , the number of clusters to be created.

Randomly generate M clusters to obtain initial partition $\mathcal{P}^{(0)}$.

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

Step 2. For each x_i , compute the distance $D(x_i, \bar{x})$ to all clusters' centers.

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

Iterate on Steps 1 and 2. Stop when the number of clusters in the partition is M . Represent each cluster with a simulated point that is closest to the cluster's center.

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

APPENDIX C: ONE-AGENT MODEL

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

An EDS algorithm iterating on the Euler equation

We parameterize K with a flexible functional form $\widehat{K}(\cdot; b)$ that depends on a coefficients vector b . Our goal is to find b that finds $\widehat{K} \approx K$ on the EDS grid given the functional form $\widehat{K}(\cdot; b)$. We compute b using fixed-point iteration (FPI). To implement fixed-point iteration on \widehat{K} , we rewrite (8) in the form

$$k' = \beta k' E \left[\frac{u_1(c')}{u_1(c)} (1 - \delta + a' Af_1(k')) \right]. \quad (\text{C.1})$$

In the true solution, k' on both sides of (C.1) takes the same value and, thus, cancels out. However, in the FPI iterative process, k' on two sides of (C.1) takes different values, namely, we substitute $k' = \widehat{K}(\cdot; b)$ in the right side of (C.1) and we compute the left side of (C.1); fixed-point iterations on b are performed until the two sides coincide.

ALGORITHM EE (An Algorithm Iterating on the Euler Equation).

Step 0. Initialization.

- a. Choose (k_0, a_0) and T .
- b. Draw $\{\epsilon_{t+1}\}_{t=0, \dots, T-1}$. Compute and fix $\{a_{t+1}\}_{t=0, \dots, T-1}$ using (7).
- c. Choose an approximating function $K \approx \widehat{K}(\cdot, b)$.
- d. Make an initial guess on b .
- e. Choose integration nodes ϵ_j and weights ω_j , $j = 1, \dots, J$.

Step 1. Construction of an EDS grid.

- a. Use $\widehat{K}(\cdot, b)$ to simulate $\{k_{t+1}\}_{t=0, \dots, T}$.
- b. Construct an EDS grid $\Gamma = \{k_m, a_m\}_{m=1, \dots, M}$.

Step 2. Computation of a solution for K .

- a. At iteration i , for $m = 1, \dots, M$, compute
 - $k'_m = \widehat{K}(k_m, a_m; b^{(i)})$ and $a'_{m,j} = a_m^\rho \exp(\epsilon_j)$ for all j ;
 - $k''_{m,j} = \widehat{K}(k'_m, a'_{m,j}; b^{(i)})$ for all j ;
 - $c_m = (1 - \delta)k_m + a_m Af(k_m) - k'_m$;
 - $c'_{m,j} = (1 - \delta)k'_m + a_m^\rho \exp(\epsilon_j) Af(k'_m) - k''_{m,j}$ for all j ;
 - $\widehat{k}'_m \equiv \beta \sum_{j=1}^J \omega_j \cdot \left[\frac{u_1(c'_{m,j})}{u_1(c_m)} [1 - \delta + a_m^\rho \exp(\epsilon_j) Af_1(k'_m)] k'_m \right]$.
- b. Find b that solve the system in Step 2a.
 - Run a regression to get $\widehat{b} \equiv \arg \min_b \sum_{m=1}^M \| \widehat{k}'_m - \widehat{K}(k_m, a_m; b) \|$.
 - Use damping to compute $b^{(i+1)} = (1 - \xi)b^{(i)} + \xi \widehat{b}$.
 - Check for convergence: end Step 2 if $\frac{1}{M} \sum_{m=1}^M \left| \frac{(k'_m)^{(i+1)} - (k'_m)^{(i)}}{(k'_m)^{(i)}} \right| < \varpi$.

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

Computational choices We parameterize the model (5)–(7) by assuming $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$ with $\gamma \in \{\frac{1}{5}, 1, 5\}$ and $f(k) = k^\alpha$ with $\alpha = 0.36$. We set $\beta = 0.99$, $\delta = 0.025$, $\rho = 0.95$, and $\sigma = 0.01$. We normalize the steady state of capital to 1 by assuming $A = \frac{1/\beta-(1-\delta)}{\alpha}$. The simulation length is $T = 100,000$, and we pick each tenth point so that $n = 10,000$. The damping parameter is $\xi = 0.1$ and the convergence parameter is $\varpi = 10^{-11}$. In Algorithm EE, we parameterize the capital equilibrium rule using complete ordinary polynomials of degrees up to 5. For example, for degree 2, we have $\widehat{K}(k, a; b) = b_0 + b_1k + b_2a + b_3k^2 + b_4ka + b_5a^2$, where $b \equiv (b_0, \dots, b_5)$. We approximate conditional expectations with a 10-node Gauss–Hermite quadrature rule (QR). We compute the vector of coefficients b using a least-squares method based on QR factorization. To construct an initial EDS grid, we simulate the model under an (arbitrary) initial guess $k' = 0.95k + 0.05a$ (this guess matches the steady-state level of capital equal to 1).

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

$$\mathcal{R}(k_i, a_i) \equiv \sum_{j=1}^{j^{\text{test}}} \omega_j^{\text{test}} \cdot \left[\beta \frac{u_1(c'_{i,j})}{u_1(c_i)} [1 - \delta + a_i^\rho \exp(\epsilon_j^{\text{test}}) Af_1(k'_i)] \right] - 1,$$

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

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

APPENDIX D: MULTI-AGENT MODEL

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

$$K^h(\{k_t^h, a_t^h\}^{h=1, \dots, N}) \approx \widehat{K}^h(\{k_t^h, a_t^h\}^{h=1, \dots, N}; b^h), \quad (\text{D.1})$$

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

$$k_{t+1}^h = E_t \left\{ \beta \frac{u'(c_{t+1})}{u'(c_t)} [1 - \delta + a_{t+1}^h Af'(k_{t+1}^h)] k_{t+1}^h \right\}. \quad (\text{D.2})$$

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

The steps of the EDS algorithm here are similar to those of Algorithm EE described in Appendix C for the one-agent model. However, we now iterate on N equilibrium rules

of the heterogeneous countries instead of just one equilibrium rule of the representative agent. That is, we make an initial guess on N coefficient vectors $\{b^h\}^{h=1,\dots,N}$, approximate N conditional expectations in Step 2a, and run N regressions in Step 2b. The damping parameter in $(b^h)^{(i+1)} = (1 - \xi)(b^h)^{(i)} + \xi \widehat{b}^h$ is $\xi = 0.1$, and the convergence parameter is $\varpi = 10^{-8}$. In the accuracy check, we evaluate the size of Euler equation residuals on a stochastic simulation of length $T^{\text{test}} = 10,200$ (we discard the first 200 observations to eliminate the effect of initial conditions). To test the accuracy of solutions, we use the Gauss–Hermite quadrature product rule with two nodes for each shock for $N < 12$, we use the monomial rule with $N^2 + 1$ points for N from $12 \leq N < 20$, and we use the monomial rule $2N$ points for $N \geq 20$; see Judd, Maliar, and Maliar (2011) for a detailed description of these integration methods.

APPENDIX E: NEW KEYNESIAN MODEL WITH THE ZLB

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

Households

The FOCs of the household problem (13)–(17) with respect to C_t , L_t , and B_t are

$$\Lambda_t = \frac{\exp(\eta_{u,t})C_t^{-\gamma}}{P_t}, \quad (\text{E.1})$$

$$\exp(\eta_{u,t} + \eta_{L,t})L_t^\vartheta = \Lambda_t W_t, \quad (\text{E.2})$$

$$\exp(\eta_{u,t})C_t^{-\gamma} = \beta \exp(\eta_{B,t})R_t E_t \left[\frac{\exp(\eta_{u,t+1})C_{t+1}^{-\gamma}}{\pi_{t+1}} \right], \quad (\text{E.3})$$

where Λ_t is the Lagrange multiplier associated with the household's budget constraint (14). After combining (E.1) and (E.2), we get

$$\exp(\eta_{L,t})L_t^\vartheta C_t^\gamma = \frac{W_t}{P_t}. \quad (\text{E.4})$$

Final-good producers

The FOC of the final-good producer's problem (18)–(19) with respect to $Y_t(i)$ yields the demand for the i th intermediate good:

$$Y_t(i) = Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon}. \quad (\text{E.5})$$

Substituting the condition (E.5) into (19), we obtain

$$P_t = \left(\int_0^1 P_t(i)^{1-\varepsilon} di \right)^{1/(1-\varepsilon)}. \quad (\text{E.6})$$

Intermediate-good producers

The FOC of the cost-minimization problem (20)–(22) with respect to $L_t(i)$ is

$$\Theta_t = \frac{(1-v)W_t}{\exp(\eta_{a,t})}, \quad (\text{E.7})$$

where Θ_t is the Lagrange multiplier associated with (21). The derivative of the total cost in (20) is the nominal marginal cost, $\text{MC}_t(i)$,

$$\text{MC}_t(i) \equiv \frac{d\text{TC}(Y_t(i))}{dY_t(i)} = \Theta_t. \quad (\text{E.8})$$

Conditions (E.7) and (E.8) taken together imply that the real marginal cost is the same for all firms:

$$\text{mc}_t(i) = \frac{(1-v)}{\exp(\eta_t^a)} \cdot \frac{W_t}{P_t} = \text{mc}_t. \quad (\text{E.9})$$

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

$$E_t \sum_{j=0}^{\infty} (\beta\theta)^j \Lambda_{t+j} Y_{t+j} P_{t+j}^{\varepsilon+1} \left[\frac{\tilde{P}_t}{P_{t+j}} - \frac{\varepsilon}{\varepsilon-1} \text{mc}_{t+j} \right] = 0. \quad (\text{E.10})$$

From the household's FOC (E.1), we have

$$\Lambda_{t+j} = \frac{\exp(\eta_{u,t+j}) C_{t+j}^{-\gamma}}{P_{t+j}}. \quad (\text{E.11})$$

Substituting (E.11) into (E.10), we get

$$E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp(\eta_{u,t+j}) C_{t+j}^{-\gamma} Y_{t+j} P_{t+j}^{\varepsilon} \left[\frac{\tilde{P}_t}{P_{t+j}} - \frac{\varepsilon}{\varepsilon-1} \text{mc}_{t+j} \right] = 0. \quad (\text{E.12})$$

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

$$\chi_{t,j} \equiv \begin{cases} 1 & \text{if } j=0, \\ \frac{1}{\pi_{t+j} \cdot \pi_{t+j-1} \cdots \pi_{t+1}} & \text{if } j \geq 1. \end{cases} \quad (\text{E.13})$$

Then $\chi_{t,j} = \chi_{t+1,j-1} \cdot \frac{1}{\pi_{t+1}}$ for $j > 0$. Therefore, (E.12) becomes

$$E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp(\eta_{u,t+j}) C_{t+j}^{-\gamma} Y_{t+j} \chi_{t,j}^{-\varepsilon} \left[\tilde{P}_t \chi_{t,j} - \frac{\varepsilon}{\varepsilon-1} \text{mc}_{t+j} \right] = 0, \quad (\text{E.14})$$

where $\tilde{p}_t \equiv \frac{\tilde{P}_t}{P_t}$. We express \tilde{p}_t from (E.14) as

$$\tilde{p}_t = \frac{E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp(\eta_{u,t+j}) C_{t+j}^{-\gamma} Y_{t+j} \chi_{t,j}^{-\varepsilon} \frac{\varepsilon}{\varepsilon-1} \text{mc}_{t+j}}{E_t \sum_{j=0}^{\infty} (\beta\theta)^j \exp(\eta_{u,t+j}) C_{t+j}^{-\gamma} Y_{t+j} \chi_{t,j}^{1-\varepsilon}} \equiv \frac{S_t}{F_t}. \quad (\text{E.15})$$

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

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

Substituting mc_t from (E.9) into the above recursive formula for S_t , we have

$$S_t = \frac{\varepsilon}{\varepsilon-1} \exp(\eta_{u,t}) C_t^{-\gamma} Y_t \frac{(1-v)}{\exp(\eta_{a,t})} \cdot \frac{W_t}{P_t} + \beta\theta E_t \{ \pi_{t+1}^{\varepsilon} S_{t+1} \}. \quad (\text{E.16})$$

Substituting $\frac{W_t}{P_t}$ from (E.4) into (E.16), we get

$$S_t = \frac{\varepsilon}{\varepsilon-1} \exp(\eta_{u,t}) Y_t \frac{(1-v)}{\exp(\eta_{a,t})} \cdot \exp(\eta_{L,t}) L_t^{\vartheta} + \beta\theta E_t \{ \pi_{t+1}^{\varepsilon} S_{t+1} \}. \quad (\text{E.17})$$

For F_t , the corresponding recursive formula is

$$F_t = \exp(\eta_{u,t}) C_t^{-\gamma} Y_t + \beta\theta E_t \{ \pi_{t+1}^{\varepsilon-1} F_{t+1} \}. \quad (\text{E.18})$$

Aggregate price relationship

Condition (E.6) can be rewritten as

$$\begin{aligned} P_t &= \left(\int_0^1 P_t(i)^{1-\varepsilon} di \right)^{1/(1-\varepsilon)} \\ &= \left[\int_{\text{reopt.}} P_t(i)^{1-\varepsilon} di + \int_{\text{nonreopt.}} P_t(i)^{1-\varepsilon} di \right]^{1/(1-\varepsilon)}, \end{aligned} \quad (\text{E.19})$$

where “reopt.” and “nonreopt.” denote, respectively, the firms that reoptimize and do not reoptimize their prices at t .

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

$$\int_{\text{nonreopt.}} P_t(i)^{1-\varepsilon} di = \int_0^1 \theta P(j)^{1-\varepsilon} \omega_{t-1}(j) dj = \theta P_{t-1}^{1-\varepsilon}. \quad (\text{E.20})$$

Substituting (E.20) into (E.19) and using the fact that all reoptimizers set $\tilde{P}_t^{1-\varepsilon}$, we get

$$P_t = [(1-\theta)\tilde{P}_t^{1-\varepsilon} + \theta P_{t-1}^{1-\varepsilon}]^{1/(1-\varepsilon)}. \quad (\text{E.21})$$

We divide both sides of (E.21) by P_t ,

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

and express

$$\tilde{p}_t = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta} \right]^{1/(1-\varepsilon)}. \quad (\text{E.22})$$

Combining (E.22) and (E.15), we obtain

$$\frac{S_t}{F_t} = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta} \right]^{1/(1-\varepsilon)}. \quad (\text{E.23})$$

Aggregate output

Let us define aggregate output

$$\bar{Y}_t \equiv \int_0^1 Y_t(i) di = \int_0^1 \exp(\eta_{a,t}) L_t(i) di = \exp(\eta_{a,t}) L_t, \quad (\text{E.24})$$

where $L_t = \int_0^1 L_t(i) di$ follows by the labor-market clearing condition. We substitute demand for $Y_t(i)$ from (E.5) into (E.24) to get

$$\bar{Y}_t = \int_0^1 Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon} di = Y_t P_t^\varepsilon \int_0^1 P_t(i)^{-\varepsilon} di. \quad (\text{E.25})$$

Let us introduce a new variable \bar{P}_t :

$$(\bar{P}_t)^{-\varepsilon} \equiv \int_0^1 P_t(i)^{-\varepsilon} di. \quad (\text{E.26})$$

Substituting (E.24) and (E.26) into (E.25) gives us

$$Y_t \equiv \bar{Y}_t \left(\frac{\bar{P}_t}{P_t} \right)^\varepsilon = \exp(\eta_{a,t}) L_t \Delta_t, \quad (\text{E.27})$$

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

$$\Delta_t \equiv \left(\frac{\bar{P}_t}{P_t} \right)^\varepsilon. \quad (\text{E.28})$$

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

Law of motion for price dispersion Δ_t

By analogy with (E.21), the variable \bar{P}_t , defined in (E.26), satisfies

$$\bar{P}_t = [(1 - \theta)\tilde{P}_t^{-\varepsilon} + \theta(\bar{P}_{t-1})^{-\varepsilon}]^{-1/\varepsilon}. \quad (\text{E.29})$$

Using (E.29) in (E.28), we get

$$\Delta_t = \left(\frac{[(1 - \theta)\tilde{P}_t^{-\varepsilon} + \theta(\bar{P}_{t-1})^{-\varepsilon}]^{-1/\varepsilon}}{P_t} \right)^\varepsilon. \quad (\text{E.30})$$

This implies

$$\Delta_t^{1/\varepsilon} = \left[(1 - \theta) \left(\frac{\tilde{P}_t}{P_t} \right)^{-\varepsilon} + \theta \left(\frac{\bar{P}_{t-1}}{P_t} \right)^{-\varepsilon} \right]^{-1/\varepsilon}. \quad (\text{E.31})$$

In terms of $\tilde{\tilde{p}}_t \equiv \frac{\tilde{P}_t}{P_t}$, condition (E.31) can be written as

$$\Delta_t = \left[(1 - \theta) \tilde{\tilde{p}}_t^{-\varepsilon} + \theta \frac{\bar{P}_{t-1}^{-\varepsilon}}{P_t^{-\varepsilon}} \cdot \frac{P_{t-1}^{-\varepsilon}}{P_{t-1}^{-\varepsilon}} \right]^{-1}. \quad (\text{E.32})$$

By substituting $\tilde{\tilde{p}}_t$ from (E.22) into (E.32), we obtain the law of motion for Δ_t :

$$\Delta_t = \left[(1 - \theta) \left[\frac{1 - \theta \pi_t^{\varepsilon-1}}{1 - \theta} \right]^{-\varepsilon/(1-\varepsilon)} + \theta \frac{\pi_t^\varepsilon}{\Delta_{t-1}} \right]^{-1}. \quad (\text{E.33})$$

Aggregate resource constraint

Combining the household's budget constraint (14) with the government's budget constraint (25), we have the aggregate resource constraint

$$P_t C_t + P_t \frac{\bar{G} Y_t}{\exp(\eta_{G,t})} = (1-v)W_t L_t + \Pi_t. \quad (\text{E.34})$$

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

$$\begin{aligned} \Pi_t &= \int_0^1 \Pi_t(i) di = \int_0^1 P_t(i)Y_t(i) di - (1-v)W_t \int_0^1 L_t(i) di \\ &= P_t Y_t - (1-v)W_t L_t, \end{aligned}$$

where $P_t Y_t = \int_0^1 P_t(i)Y_t(i) di$ follows by a zero-profit condition of the final-good firms. Hence, (E.34) can be rewritten as

$$P_t C_t + P_t \frac{\bar{G}}{\exp(\eta_{G,t})} Y_t = P_t Y_t. \quad (\text{E.35})$$

In real terms, the aggregate resource constraint (E.35) becomes

$$C_t = \left(1 - \frac{\bar{G}}{\exp(\eta_{G,t})}\right) Y_t. \quad (\text{E.36})$$

Equilibrium conditions

Condition (32) in the main text follows from (E.17) under the additional assumption $\frac{\varepsilon}{\varepsilon-1}(1-v) = 1$, which ensures that the model admits a deterministic steady state (this assumption is commonly used in the related literature; see, for example, Christiano, Eichenbaum, and Rebelo (2011)). Conditions (33)–(38) in the main text correspond to conditions (E.18), (E.23), (E.33), (E.3), (E.27), and (E.36) in the present appendix.

Steady state

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

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

$$S_* = \frac{(1 - \bar{G})^{-1} C_*}{\Delta_*} + \beta \theta \pi_*^\varepsilon S_*,$$

$$R_* = \pi_* / \beta,$$

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

Calibration procedure

Most of the parameters are calibrated using the estimates of Del Negro, Schorfheide, Smets, and Wouters (2007, Table 1, column “DSGE posterior”); namely, we assume $\gamma = 1$ and $\vartheta = 2.09$ in the utility function (13); $\phi_y = 0.07$, $\phi_\pi = 2.21$, and $\mu = 0.82$ in the Taylor rule (39); $\varepsilon = 4.45$ in the production function of the final-good firm (19); $\theta = 0.83$ (the fraction of the intermediate-good firms affected by price stickiness); $\bar{G} = 0.23$ in the government budget constraint (25); and $\rho_u = 0.92$, $\rho_G = 0.95$, $\rho_L = 0.25$, $\sigma_u = 0.54\%$, $\sigma_G = 0.38\%$, $\sigma_L = 18.21\%$ (the latter is a lower estimate of Del Negro et al. (2007, Table 1, column “DSGE posterior”)), and $\sigma_L = 40.54\%$ (an average estimate of Del Negro et al. (2007)) in the processes for shocks (15), (26), and (16). From Smets and Wouters (2007), we take the values of $\rho_a = 0.95$, $\rho_B = 0.22$, $\rho_R = 0.15$, $\sigma_a = 0.45\%$, $\sigma_B = 0.23\%$, and $\sigma_R = 0.28\%$ in the processes for shocks (22), (17), and (28). We set the discount factor at $\beta = 0.99$. To parameterize the Taylor rule (39), we use the steady-state interest rate $R_* = \frac{\pi_*}{\beta}$, and we consider two alternative values of the target inflation, $\pi_* = 1$ (a zero net inflation target) and $\pi_* = 1.0598$ (this estimate comes from Del Negro et al. (2007)).

Solution procedure

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

To approximate the equilibrium rules, we use the family of ordinary polynomials. To compute the conditional expectations in the Euler equations (32), (33), and (38), we use monomial formula *M1* with $2N$ nodes.

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

The simulation length is $T = 100,000$, and we pick each tenth point so that $n = 10,000$. The target number of grid points is $\bar{M} = 1000$. In Step 2b, the damping parameter is set at $\xi = 0.1$, and the convergence parameter is set at $\varpi = 10^{-7}$. We compute residuals on a stochastic simulation of 10,200 observations (we eliminate the first 200 observations). In the test, we use monomial rule *M2* with $2 \cdot 6^2 + 1$ nodes, which is more accurate than monomial rule *M1* used in the solution procedure; see Judd et al. (2011) for a detailed description of these integration formulas. Dynare does not evaluate the accuracy of perturbation solutions itself. We wrote a MATLAB routine that simulates the perturbation solutions and evaluates their accuracy using the Dynare’s representation of the

state space that includes the current endogenous state variables $\{\Delta_{t-1}, R_{t-1}\}$, the past exogenous state variables $\{\eta_{u,t-1}, \eta_{L,t-1}, \eta_{B,t-1}, \eta_{a,t-1}, \eta_{R,t-1}, \eta_{G,t-1}\}$, and the current disturbances $\{\epsilon_{u,t}, \epsilon_{L,t}, \epsilon_{B,t}, \epsilon_{a,t}, \epsilon_{R,t}, \epsilon_{G,t}\}$.

ALGORITHM EE-NK (An Algorithm Iterating on the Euler Equation).

Step 0. Initialization.

- a. Choose $(\Delta_{-1}, R_{-1}, \eta_{u,0}, \eta_{L,0}, \eta_{B,0}, \eta_{a,0}, \eta_{R,0}, \eta_{G,0})$ and T .
- b. Draw $\{\epsilon_{u,t+1}, \epsilon_{L,t+1}, \epsilon_{B,t+1}, \epsilon_{a,t+1}, \epsilon_{R,t+1}, \epsilon_{G,t+1}\}_{t=0,\dots,T-1}$. Compute and fix $\{\eta_{u,t+1}, \eta_{L,t+1}, \eta_{B,t+1}, \eta_{a,t+1}, \eta_{R,t+1}, \eta_{G,t+1}\}_{t=0,\dots,T-1}$.
- c. Choose approximating functions $S \approx \widehat{S}(\cdot; b^S)$, $F \approx \widehat{F}(\cdot; b^F)$, $\text{MU} \approx \widehat{\text{MU}}(\cdot; b^{\text{MU}})$.
- d. Make initial guesses on b^S , b^F , and b^{MU} .
- e. Choose integration nodes, $\{\epsilon_{u,j}, \epsilon_{L,j}, \epsilon_{B,j}, \epsilon_{a,j}, \epsilon_{R,j}, \epsilon_{G,j}\}_{j=1,\dots,J}$ and weights, $\{\omega_j\}_{j=1,\dots,J}$.

Step 1. Construction of an EDS grid.

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

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

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

- a. At iteration i , for $m = 1, \dots, M$, compute
 - $S_m = \widehat{S}(x_m; b^S)$, $F_m = \widehat{F}(x_m; b^F)$, $C_m = [\widehat{\text{MU}}(x_m; b^{\text{MU}})]^{-1/\gamma}$;
 - π_m from $\frac{S_m}{F_m} = \left[\frac{1-\theta\pi_m^{\varepsilon-1}}{1-\theta} \right]^{1/(1-\varepsilon)}$ and $\Delta'_m = [(1-\theta) \left[\frac{1-\theta\pi_m^{\varepsilon-1}}{1-\theta} \right]^{\varepsilon/(\varepsilon-1)} + \theta \frac{\pi_m^{\varepsilon}}{\Delta_m}]^{-1}$;
 - $Y_m = (1 - \frac{\bar{G}}{\exp(\eta_{G,m})})^{-1} C_m$, and $L_m = Y_m [\exp(\eta_{a,m}) \Delta'_m]^{-1}$;
 - $Y_{N,m} = \left[\frac{\exp(\eta_{a,m})^{1+\vartheta}}{[\exp(\eta_{G,m})]^{-\gamma} \exp(\eta_{L,m})} \right]^{1/(\vartheta+\gamma)}$;
 - $R'_m = \max\{1, \Phi_m\}$, $\Phi_m = R_* \left(\frac{R_m}{R_*} \right)^\mu \left[\left(\frac{\pi_m}{\pi_*} \right)^\phi \left(\frac{Y_m}{Y_{N,m}} \right)^{\phi\gamma} \right]^{1-\mu} \exp(\eta_{R,m})$;
 - $x'_{m,j} = (\Delta'_m, R'_m, \eta'_{u,m,j}, \eta'_{L,m,j}, \eta'_{B,m,j}, \eta'_{a,m,j}, \eta'_{R,m,j}, \eta'_{G,m,j})$ for all j ;
 - $S'_{m,j} = \widehat{S}(x'_{m,j}; b^S)$, $F'_{m,j} = \widehat{F}(x'_{m,j}; b^F)$, $C'_{m,j} = [\widehat{\text{MU}}(x'_{m,j}; b^{\text{MU}})]^{-1/\gamma}$;
 - $\pi'_{m,j}$ from $\frac{S'_{m,j}}{F'_{m,j}} = \left[\frac{1-\theta(\pi'_{m,j})^{\varepsilon-1}}{1-\theta} \right]^{1/(1-\varepsilon)}$;
 - $\widehat{S}_m = \frac{\exp(\eta_{u,m} + \eta_{L,m})}{\exp(\eta_{a,m})} L_m^\vartheta Y_m + \beta \theta \sum_{j=1}^J \omega_j \cdot \{(\pi'_{m,j})^\varepsilon S'_{m,j}\}$;
 - $\widehat{F}_m = \exp(\eta_{u,m}) C_m^{-\gamma} Y_m + \beta \theta \sum_{j=1}^J \omega_j \cdot \{(\pi'_{m,j})^{\varepsilon-1} F'_{m,j}\}$;
 - $\widehat{C}_m^{-\gamma} = \frac{\beta \exp(\eta_{B,m}) R'_m}{\exp(\eta_{u,m})} \sum_{j=1}^J \omega_j \cdot \left[\frac{(C'_{m,j})^{-\gamma} \exp(\eta'_{u,m,j})}{\pi'_{m,j}} \right]$.

- b. Find b^S , b^F , and b^{MU} that solve the system in Step 2a.
- Get: $\hat{b}^S \equiv \arg \min_{b^S} \sum_{m=1}^M \|\hat{S}_m - \hat{S}(x_m; b^S)\|$. Similarly, get \hat{b}^F and \hat{b}^{MU} .
 - Use damping to compute $b^{(i+1)} = (1 - \xi)b^{(i)} + \xi \hat{b}$, where $b \equiv (\hat{b}^S, \hat{b}^F, \hat{b}^{\text{MU}})$.
 - Check for convergence: end Step 2 if $\frac{1}{M} \max\{\sum_{m=1}^M |\frac{(S_m)^{(i+1)} - (S_m)^{(i)}}{(S_m)^{(i)}}|, \sum_{m=1}^M |\frac{(F_m)^{(i+1)} - (F_m)^{(i)}}{(F_m)^{(i)}}|, \sum_{m=1}^M |\frac{(\text{MU}_m)^{(i+1)} - (\text{MU}_m)^{(i)}}{(\text{MU}_m)^{(i)}}|\} < \varpi$.
- Iterate on Steps 1 and 2 until convergence of the EDS grid.

REFERENCES

- 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. [5]
- Christiano, L., M. Eichenbaum, and S. Rebelo (2011), “When is the government spending multiplier large?” *Journal of Political Economy*, 119 (1), 78–121. [19]
- 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. [20]
- Everitt, B., S. Landau, M. Leese, and D. Stahl (2011), *Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley, Chichester. [9]
- Judd, K., L. Maliar, and S. Maliar (2010), “A cluster grid projection method: Solving problems with high dimensionality.” Working Paper 15965, NBER. [6]
- Judd, K., L. Maliar, and S. Maliar (2011), “Numerically stable and accurate stochastic simulation approaches for solving dynamic models.” *Quantitative Economics*, 2, 173–210. [14, 20]
- 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. [3]
- Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania. [1, 3]
- Rényi, A. (1958), “On a one-dimensional space-filling problem.” *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 3, 109–127. [5]
- Romesburg, C. (1984), *Cluster Analysis for Researchers*. Lifetime Learning Publications, Belmont, California. [9]
- Smets, F. and R. Wouters (2007), “Shocks and frictions in US business cycles: A Bayesian DSGE approach.” *American Economic Review*, 97 (3), 586–606. [20]
- Temlyakov, V. (2011), *Greedy Approximation*. Cambridge University Press, Cambridge. [2, 4, 5]