A Cluster-Grid Algorithm: Solving Problems With High Dimensionality*

Kenneth L. Judd, Lilia Maliar and Serguei Maliar

August 11, 2011

Abstract

We develop a cluster-grid algorithm (CGA) that solves dynamic economic models on their ergodic sets and is tractable in problems with high dimensionality (hundreds of state variables) on a desktop computer. The key new feature is the use of methods from cluster analysis to approximate an ergodic set. CGA guesses a solution, simulates the model, partitions the simulated data into clusters and uses the centers of the clusters as a grid for solving the model. Thus, CGA avoids costs of finding the solution in areas of the state space that are never visited in equilibrium. In one example, we use CGA to solve a large-scale new Keynesian model that includes a Taylor rule with a zero lower bound on nominal interest rates.

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

Key Words : ergodic set; clusters; large-scale economy; new Keynesian model; ZLB; projection method; numerical method; stochastic simulation

*This paper is a substantially revised version of an earlier version that circulated as NBER working paper 15965. We are indebted to the editor and three anonymous referees for many useful comments and suggestions, and in particular, for a suggestion to analyze a new Keynesian model. Errors are ours. Lilia Maliar and Serguei Maliar acknowledge support from the Hoover Institution at Stanford University, Ivie, the Ministerio de Ciencia e Innovación and FEDER funds under the project SEJ-2007-62656 and the Generalitat Valenciana under the grants BEST/2011/283 and BEST/2011/282, respectively.

1 Introduction

This paper introduces a projection method that solves dynamic economic models on ergodic sets realized in equilibrium. The key new feature is the use of methods from cluster analysis to approximate an ergodic set. We guess a solution, simulate the model, partition the simulated data into clusters and use the centers of the clusters as a grid for solving the model. We call this method *a cluster-grid algorithm* (CGA).

Making the solution domain endogenous to the model allows us to avoid the cost of finding the solution in areas of the state space that are never visited in equilibrium. The higher is the dimensionality of the problem, the larger is the saving from focusing on the ergodic set. We complement the cluster-grid construction with other computational techniques suitable for high-dimensional applications, namely, low-cost monomial integration rules and a fixed-point iteration method for finding parameters of the equilibrium policy functions. Taken together, these techniques make CGA tractable in problems with high dimensionality.

We first apply CGA to the usual test models, the standard neoclassical growth models with one or multiple agents (countries), and find it to be reliable and tractable. First, CGA delivers accuracy levels comparable to the highest accuracy attained in the related literature: unit-free approximation errors (evaluated on a stochastic simulation of $10,000$ observations) are smaller than 10^{-3} to 10^{-8} for the polynomial approximations of degrees from 1 to 5, respectively. Second, CGA is tractable in larger problems than those studied in the related literature; in particular, we compute global quadratic solutions for the models with up to 80 state variables on a desktop computer. Third, we identify combinations of the approximating functions and integration rules that complement each other. There is no value in using high-quality approximations without using high-quality integration, and vice versa. Finally, the cost of the hierarchical clustering algorithm we use is modest even if the dimensionality is high; for example, it takes about one minute to construct a grid of 300 points (clusters' centers) using a simulation of $10,000$ observations for an economy with 400 state variables.

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. Our model has eight state variables and is characterized by a kink in policy functions due to the ZLB. We parameterize the model using the estimates of Smets and Wouters (2003, 2007), and Del Negro, Schorfheide, Smets and Wouters (2007). We compute CGA polynomial solutions of degrees 2 and 3, referred to as CGA2 and CGA3, respectively. The running time of CGA is less than 25 minutes for all cases. For comparison, we also

compute perturbation solutions of orders 1 and 2, referred to as PER1 and PER2, respectively. When we simulate the perturbation solutions, we set the nominal interest rate to the maximum of zero and the interest rate implied by the perturbation solution, as is commonly done in the literature.

We find that the importance of the ZLB depends on the target inflation rate. The ZLB is quantitatively important in the economy with 0% (net) target inflation (the ZLB binds in 8% of cases) but not in the economy with 5.98% target inflation rate (the ZLB binds in just 0.13% of cases). We find that the perturbation method is unreliable. PER1 is highly inaccurate – the approximation errors can be as large as 17%. PER2 has the maximum error that ranges from 2% to 9% depending on the parameterization. The errors increase with the target inflation rate (even if the ZLB is not imposed) and are particularly large when the ZLB binds. In contrast, the accuracy of CGA solutions is not significantly affected by the target inflation rate and ZLB. In all experiments considered, CGA2 and CGA3 produce the maximum errors of less than 2% and 1%, respectively. The difference in accuracy between the CGA and perturbation methods is economically important. In particular, the perturbation method significantly understates the duration of ZLB episodes.

The CGA method is related to three classes of methods in the literature.[1] First, CGA is similar to stochastic simulation methods of Fair and Taylor (1984), Den Haan and Marcet (1990), Rust (1996), Pakes and McGuire (2001), Maliar and Maliar (2005), and Judd, Maliar and Maliar (2011) in that it computes a solution on the ergodic set. The key difference between CGA and those methods is that we use a cluster-grid representation of the ergodic set, which is more efficient than a set of simulated points containing many redundant closely located points.

Second, CGA is similar to projection methods of Judd (1992) and Krueger and Kubler (2004) in that it computes a solution on a grid of points, however, in CGA, this grid is iteratively adapted to approximate the ergodic set, whereas the previous projection methods use only one fixed grid of preselected points. At the solution, CGA will place its grid points on the ergodic-set domain, which is typically much smaller than the hypercube domains examined by the above methods (the size of a hypercube grows exponentially with the dimensionality of the state space).[2] We perform experiments that compare

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

[2] The key difference between the methods of Judd (1992) and Kubler and Krueger (2004) is that the former uses a tensor-product grid within a hypercube, while the latter relies on a (non-product) Smolyak sparse grid with a smaller set of points within the hypercube.

the solutions under the cluster and hypercube grids. We find that the cluster grid leads to more accurate solutions in the ergodic set than the Smolyak grid, however, the Smolyak grid does better on the hypercube domain, illustrating the trade-off between the fit inside and outside the ergodic set. We also find that the cluster grid is autocorrecting – even if our initial guess implies a poor approximation of the ergodic set, the cluster grid quickly converges to the ergodic set along iterations.

Finally, CGA is similar to perturbation methods in that it can solve problems with high dimensionality.[3] However, CGA solutions are global in the sense that they are accurate on the ergodic set, whereas perturbation solutions are accurate in some neighborhood of the steady state, which is typically much smaller than the ergodic set. The accuracy of perturbation solutions decreases rapidly away from the steady state; see, e.g., Judd and Guu (1993), Aruoba et al. (2006), and Kollmann et al. (2011) for assessments of accuracy of perturbation methods.

CGA can be used to accurately solve small-scale models that were studied using other solution methods. However, a comparative advantage of CGA is its ability to solve large-scale problems that other methods find intractable or expensive. Such problems commonly arise in macroeconomics (multiple agents), international trade (multiple countries and goods), industrial organization (multiple firms), finance (multiple assets), climate change (multiple countries and sectors), etc. The speed of CGA also makes it useful in estimation methods that solve economic models at many parameters vectors; see Fernández-Villaverde and Rubio-Ramírez (2007) for a discussion.

The rest of the paper is as follows: In Section 2, we describe the construction of our endogenous cluster grid. In Section 3, we provide a general description of CGA for the studied class of dynamic economic models. In Section 4, we apply the CGA algorithm to solving the standard neoclassical growth model. In Section 5, we use CGA to solve a new Keynesian model with the ZLB. In Section 6, we conclude.

2 Cluster grid

The objective of this paper is to develop a projection method that solves dynamic models on the ergodic set. In this section, we construct a grid that approximates the ergodic set and that will be used as a solution domain.

[3]Perturbation methods are studied in, e.g., Judd and Guu (1993), Gaspar and Judd (1997), Collard and Juillard (2001), and Kollmann, Kim and Kim (2011).

2.1 An advantage of focusing on the ergodic set

Consider an example of the standard representative-agent neoclassical growth model with a closed-form solution (see Section 3 for a description of this model). In Figure 1a, we plot simulated series for capital and productivity over $10,000$ periods, which we use as an approximation of the ergodic set realized in equilibrium. The ergodic set has the shape of an ellipse. We can therefore save on cost by solving the model just on this ellipse instead of the standard rectangular domain that encloses the ellipse.[4]

The savings increase rapidly with the dimensionality of the problem. Suppose that the ergodic set is a hypersphere. With p state variables, the ratio of the volume of a hypersphere to the volume of a hypercube that encloses it is equal to

$$\mathcal{V}^p = \begin{cases} \frac{(\pi/2)^{\frac{p-1}{2}}}{1\cdot 3\cdot \ldots \cdot p} \text{ for } p = 1,3,5... \\ \frac{(\pi/2)^{\frac{p}{2}}}{2\cdot 4\cdot \ldots \cdot p} \text{ for } p = 2,4,6... \end{cases}. \tag{1}$$

For dimensions 2, 3, 4, 5, 10, 30 and 100, the ratio $\mathcal{V}^p$ is 0.79, 0.52, 0.31, 0.16, $3\cdot 10^{-3}$, $2\cdot 10^{-14}$ and $2\cdot 10^{-70}$, respectively. The ratio between the volume of a hyperelliptic ergodic set and the enclosing hypercube is even smaller. Thus, in high-dimensional problems, enormous cost savings are possible when we focus on the ergodic set instead of the standard hypercube domain.

2.2 A grid approximating the ergodic set

The simplest possible finite set of points that approximates the ergodic set is a set of points from simulations; effectively, this is the grid used by stochastic simulation methods, see Judd, Maliar and Maliar (2011). In this paper, we propose a more efficient grid that approximates the ergodic set: we replace a large number of closely located simulated points with a relatively small number of "representative" points. We construct such a grid using techniques from cluster analysis. A clustering algorithm partitions a set of observations into disjointed subsets called *clusters* so that observations within each cluster are more similar to one another than observations belonging to different clusters. In Figure 1b, we show an example of a partition of the simulated points from the previous example into 4 clusters. We then replace observations in each cluster with just one point, the cluster's center, computed as the average of all observations in the given cluster. In Figure 1c, we show the corresponding centers of 4 clusters. We call the collection of the clusters'

[4]Having a control over the domain on which a problem is solved is particularly useful in applications where non-ergodic-set areas are relevant for the analysis.

centers a *cluster grid*, and we use such a grid as a solution domain for our projection method.[5]

2.3 Hierarchical clustering algorithm

We study a hierarchical algorithm which begins from individual objects (observations) and agglomerates them iteratively into larger objects – clusters.

Data preprocessing In the example shown in Figure 1a, the two state variables – capital and productivity level – have different ranges of values and are significantly correlated. Both, measurement units of variables and correlation between variables, affect the distances between observations and, hence, the resulting clusters. We preprocess the simulated data prior to constructing clusters. We first orthogonalize the variables (i.e., transform correlated variables into uncorrelated ones), and we then normalize them (i.e., transform into a unit-invariant form).

Let $X \in \mathbb{R}^{I \times \mathcal{L}}$ be a set of simulated data. Let x_i^ℓ be an element of X in the ith row, denoted by x_i, and ℓth column, denoted by x^ℓ. We refer to $x_i \equiv \left(x_i^1, ..., x_i^{\mathcal{L}}\right)$ as an observation i (there are I observations), and we refer to $x^\ell \equiv \left(x_1^\ell, ..., x_I^\ell\right)^\top$ as a variable ℓ (there are $\mathcal{L}$ variables). Thus, we have $X = \left(x^1, ..., x^{\mathcal{L}}\right) = (x_1, ..., x_I)^\top$.

To orthogonalize the data, we use a principal components (PCs) transformation. Let the variables $\left(x^1, ..., x^{\mathcal{L}}\right)$ be normalized to zero mean and unit variance. Consider the singular value decomposition of X, defined as $X = USV^\top$, where $U \in \mathbb{R}^{I \times \mathcal{L}}$ and $V \in \mathbb{R}^{\mathcal{L} \times \mathcal{L}}$ are orthogonal matrices, and $S \in \mathbb{R}^{\mathcal{L} \times \mathcal{L}}$ is a diagonal matrix with diagonal entries $s_1 \geq s_2 \geq ... \geq s_{\mathcal{L}} \geq 0$, called *singular values* of X. Perform a linear transformation of X using the matrix of singular vectors V as follows: $Z \equiv XV$, where $Z = \left(z^1, ..., z^{\mathcal{L}}\right) \in \mathbb{R}^{I \times \mathcal{L}}$. The variables $z^1, ..., z^{\mathcal{L}}$ are called *principal components* of X, and are orthogonal (uncorrelated), $\left(z^{\ell'}\right)^\top z^\ell = 0$ for any $\ell' \neq \ell$ and $\left(z^\ell\right)^\top z^\ell = s_\ell^2$. The sample variance of z^ℓ is s_ℓ^2/I, and, thus, z^1 and $z^{\mathcal{L}}$ have the largest and smallest sample variances, respectively. Figure 2a shows the directions of two principal components for our example. In Figure 2b, we switch to the PC directions by translating the origin and rotating the system of coordinates. Finally, in Figure 2c, we normalize PCs to unit variance. The resulting ergodic set will be used for constructing clusters (in our example, this set has

[5] Clustering techniques are used in the unsupervised classification literature to identify natural groups in the data. Our use of clustering tools has a more limited goal: we simply construct a set of evenly spaced points that approximates the cloud of simulated data.

the shape of a circle). After clusters are constructed, we return to the original system of coordinates by using an inverse PCs transformation.

Distance between individual observations As a measure of distance between two points (observations) x_i and x_j, we use the Euclidean (or L_2 norm) distance

$$d\left(x_i, x_j\right) = \left[\sum_{\ell=1}^{\mathcal{L}} \left(x_i^{\ell} - x_j^{\ell}\right)^2\right]^{1/2}, \tag{2}$$

where $x_i \equiv \left(x_i^1, ..., x_i^{\mathcal{L}}\right) \in \mathbb{R}^{\mathcal{L}}$ and $x^j \equiv \left(x_j^1, ..., x_j^{\mathcal{L}}\right) \in \mathbb{R}^{\mathcal{L}}$.

Distance between groups of observations As a measure of distance between two groups of observations (clusters), $A \equiv \{x_1, ..., x_I\}$ and $B \equiv \{y_1, ..., y_J\}$, we use Ward's measure of distance.[6] This measure shows how much the dispersion of observations changes when the clusters A and B are merged together compared to the case when A and B are separate clusters.

Formally, we proceed as follows:

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

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

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

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

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

$$D\left(A, B\right) = SSD\left(A \cup B\right) - \left[SSD\left(A\right) + SSD\left(B\right)\right]. \tag{3}$$

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

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

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

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

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

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

Iterate on Steps 1 and 2. Stop when the number of clusters in the partition is M. (In the online Appendix A, we illustrate the operation of this algorithm by way of example.)

In Figures 2d, 2e and 2f, we draw, respectively, 4, 10 and 100 clusters on the normalized PCs shown in Figure 2c (the clusters in Figure 1c are obtained from those in Figure 2d). We draw attention to two features of the cluster grid. First, the constructed clusters provide a relatively uniform coverage of the ergodic set. Second, clustering algorithms can identify disjointed areas of the state space and hence, can cover ergodic sets of irregular shapes including those composed of multiple recurrent classes (for example, in the upper part of Figure 2f, we observe a cluster which is considerably separated from the rest of the clusters).

3 General description of the CGA algorithm

In this section, we outline the studied class of problems and provide a general description of the CGA algorithm.

3.1 The studied class of problems

We study a class of dynamic economic models, whose solutions are characterized by the set of equilibrium conditions for $t = 0, 1, \ldots, \infty$,

$$E_t \left[G \left(x_t, z_t, y_t, x_{t+1}, z_{t+1}, y_{t+1} \right) \right] = 0, \tag{4}$$

$$z_{t+1} = Z \left(z_t, \epsilon_{t+1} \right), \tag{5}$$

where the initial condition (x_0, z_0) is given; E_t denotes the expectations operator conditional on information available at t; $x_t \in \mathbb{R}^{n_x}$ is a vector of endogenous state variables at t; $z_t \in \mathbb{R}^{n_z}$ is a vector of exogenous (random) state variables at t; $y_t \in \mathbb{R}^{n_y}$ is a vector of non-state variables – prices, consumption, labor supply, etc. – also called non-predetermined variables; G is

a continuously differentiable vector function; $\epsilon_{t+1} \in \mathbb{R}^{n_z}$ is a vector of disturbances whose probability distribution is given (ϵ_{t+1} is not known at t); z_{t+1} in (5) has a unique invariant measure with finite moments; x_{t+1} is known at t, while y_{t+1} is not known at t.

A solution is given by a set of policy functions $x_{t+1} = X(x_t, z_t)$, and $y_t = Y(x_t, z_t)$ that satisfy (4), (5) in the relevant area of the state space. We assume that the functions G and Z satisfy jointly a set of regularity conditions that ensure that the solution exists and is unique. We also assume that there is a unique steady state. Finally, we assume that the ergodic set consists of a unique recurrent class.

3.2 The CGA algorithm

The cluster-grid projection algorithm had two stages and proceeds as follows:

Stage 1. Compute a candidate solution.

- *Initialization.* Choose initial state (x_0, z_0) for simulations. Choose a simulation length, T. Draw a sequence for shocks $\{\epsilon_t\}_{t=1,...,T}$. Construct and fix $\{z_{t+1}\}_{t=0}^{T-1}$ using $z_{t+1} = Z(z_t, \epsilon_{t+1})$. Parameterize the policy functions for endogenous variables with flexible functional forms $x_{t+1} = X(x_t, z_t) \approx \widehat{X}(x_t, z_t; b^x)$ and $y_t = Y(x_t, z_t) \approx \widehat{Y}(x_t, z_t; b^y)$.[7] Make an initial guess on the coefficients vectors b^x and b^y.

- *Step 1. (Construct the cluster grid.)* Given b^x and b^y, simulate the model T periods forward. Construct M clusters on the simulated series of state variables $\{x_t, z_t\}_{t=1,...,T}$ and compute the clusters' centers $\mathcal{G} \equiv \{x_m, z_m\}_{m=1,...,M}$ to be used as a grid for finding a solution.

- *Step 2. (Solve for the policy functions on the grid $\mathcal{G}$.)* Substitute $\widehat{X}(x_t, z_t; b^x)$ and $\widehat{Y}(x_t, z_t; b^y)$ in (4). For $m = 1, ..., M$, approximate the conditional expectation by a weighted average of the integrand in a set of J nodes

$$\sum_{j=1}^{J} \omega_j \cdot G\left(x_m, z_m, y_m, x'_m, z'_{m,j}, y'_{m,j}\right) = 0, \tag{6}$$

where $y_m \equiv \widehat{Y}(x_m, z_m; b^y)$, $x'_m \equiv \widehat{X}(x_m, z_m; b^x)$, $z'_{m,j} \equiv Z(z_m, \varepsilon_j)$, $y'_{m,j} \equiv \widehat{Y}\left(\widehat{X}(x_m, z_m; b^y), Z(z_m, \varepsilon_j)\right)$; the primes on the variables mean

[7] Typically, we do not numerically approximate all policy functions but a minimal subset of such functions that is sufficient for inferring all variables using analytical relations derived from equilibrium conditions.

their next-period values; and ε_j and ω_j are the integration nodes and weights, respectively. Find b^x and b^y that solve the system (6).[8]

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

Stage 2. Accuracy check.

Subject the candidate solution obtained in Stage 1 to a tight accuracy check. Construct a set of points for the state variables $\{x_\tau, z_\tau\}_{\tau=1,\ldots,T^{\text{test}}}$ that represents the domain on which accuracy is tested. Evaluate the size of approximation errors in those points using

$$\mathcal{E}(x_\tau, z_\tau) \equiv \sum_{j=1}^{J^{\text{test}}} \omega_j^{\text{test}} \cdot \left[G\left(x_\tau, z_\tau, y_\tau, x'_\tau, z'_{\tau,j}, y'_{\tau,j}\right)\right], \tag{7}$$

where $y_\tau = \widehat{Y}(x_\tau, z_\tau; b^y)$, $x'_\tau = \widehat{X}(x_\tau, z_\tau; b^x)$, $z'_{\tau,j} = Z\left(z_\tau, \varepsilon_j^{\text{test}}\right)$ and $y'_{\tau,j} = \widehat{Y}\left(\widehat{X}(x_\tau, z_\tau; b^y), Z\left(z_\tau, \varepsilon_j^{\text{test}}\right)\right)$; and $\varepsilon_j^{\text{test}}$ and ω_j^{test} are the integration nodes and weights, respectively. Find a mean and/or maximum of (7) and judge whether the candidate solution has an economically acceptable error. If not, modify the choices made in Stage 1 (i.e., simulation length, number of clusters, approximating functions, integration method) and repeat Stage 1.

4 Neoclassical stochastic growth model

In this section, we use CGA 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 model

The representative agent solves

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

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

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

[8] Different combinations of computational techniques can be used to implement this step. In Section 4, we discuss some possible choices including those of a family of approximating functions, integration method and iterative procedure for finding b^x and b^y.

where initial condition (k_0, a_0) is given; c_t, k_t and a_t are, respectively, consumption, capital and productivity level; $\beta \in (0,1)$ is the discount factor; $\delta \in (0,1]$ is the depreciation rate of capital; $\rho \in (-1,1)$ and $\sigma \geq 0$ are the autocorrelation coefficient of the productivity level and standard deviation of the productivity shock, respectively; A is a normalizing constant; u and f are the utility and production functions, respectively; both are strictly increasing, continuously differentiable and concave. The Euler equation that corresponds to (8)–(10) is

$$u'(c_t) = E_t \left\{ \beta u'(c_{t+1}) \left[1 - \delta + a_{t+1} A f'(k_{t+1}) \right] \right\}, \tag{11}$$

where u' and f' are the first derivatives of the utility and production functions, respectively. We look for a solution to (8)–(10) in the form of capital policy function, $k_{t+1} = K(k_t, a_t)$, that satisfies (9)–(11). Under our assumptions, the solution exists and is unique; see, e.g., Stockey and Lucas with Prescott (1989, p. 392). In particular, under $u(c) = \ln(c)$, $d = 1$ and $f(k) = k^\alpha$, the model admits a closed-form solution $k_{t+1} = \alpha\beta\theta_t A k_t^\alpha$ (this solution was used to produce Figures 1 and 2).

4.2 Implementation of CGA

To solve the model described in Section 3, we parameterize the capital policy function with a flexible functional form, $K(k_t, a_t) \approx \widehat{K}(k_t, a_t; b)$, that depends on a coefficients vector b. We then rewrite the Euler equation (11) in the following equivalent form

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

We need to compute b which makes $\widehat{K}(k_t, a_t; b)$ be the best possible approximation of $K(k_t, a_t)$ in the relevant area of the state space given the functional form $\widehat{K}$. The optimal capital policy function is a fixed-point solution to (12): if we substitute $K(k_t, a_t)$ into the right side of (12) and compute conditional expectation, we must get the same function $k_{t+1} = K(k_t, a_t)$ for all (k_t, a_t) in the relevant area of the state space.

Step 1 of the CGA algorithm is as described in Section 3.2: we make a guess on b, use $k_{t+1} = \widehat{K}(k_t, a_t; b)$ to simulate the series $\{k_t, a_t\}_{t=1,\ldots,T}$, construct M clusters on these series, and compute the grid of clusters' centers $\mathcal{G} = \{k_m, a_m\}_{m=1,\ldots,M}$. *Step 2* is elaborated below.

Step 2 (i). At iteration p, given the current guess $b^{(p)}$, approximate the

conditional expectation in (12) in each point of $\mathcal{G}$,

$$\widehat{k}'_m \equiv \sum_{j=1}^{J} \omega_j \cdot \left[\beta \frac{u'\left(c'_{m,j}\right)}{u'\left(c_m\right)} \left[1 - \delta + a_m^{\rho} \exp\left(\varepsilon_j\right) A f'\left(k'_m\right)\right] k'_m \right], \quad (13)$$

$$c_m = (1-\delta)\, k_m + a_m A f\left(k_m\right) - k'_m, \quad (14)$$

$$c'_{m,j} = (1-\delta)\, k'_m + a_m^{\rho} \exp\left(\varepsilon_j\right) A f\left(k'_m\right) - k''_m, \quad (15)$$

with $k'_m = \widehat{K}\left(k_m, a_m; b^{(p)}\right)$ and $k''_{m,j} = \widehat{K}\left(k'_m, a_m^{\rho} \exp\left(\varepsilon_j\right); b^{(p)}\right)$; and ε_j and ω_j are the integration nodes and weights, respectively.

Step 2 (ii). Run a regression with some norm $\|\cdot\|$ to get

$$\widehat{b} \equiv \arg\min_{b} \sum_{m=1}^{M} \left\| \widehat{k}'_m - \widehat{K}\left(k_m, a_m; b\right) \right\|. \quad (16)$$

Step 2 (iii). Check for convergence and end Step 2 if

$$\frac{1}{M} \sum_{m=1}^{M} \left| \frac{k'_m - \widehat{k}'_m}{k'_m} \right| < \varpi. \quad (17)$$

Step 2 (iv). Compute $b^{(p+1)}$ for iteration $p+1$ using fixed-point iteration

$$b^{(p+1)} = (1-\xi)\, b^{(p)} + \xi \widehat{b}, \quad (18)$$

where $\xi \in (0, 1]$ is a damping parameter. Go to *Step 2 (i)*.

We now provide a discussion of the steps of the above solution method.

Approximating function The approximating function $\widehat{K}$ can be any (polynomial or non-polynomial) function that is flexible enough to accurately approximate the policy function. We restrict attention to polynomial functions that are linear in the coefficients b, i.e., $\widehat{K}\left(k_t, a_t; b\right) = \sum_{i=0}^{n} b_i \psi_i\left(k_t, a_t\right)$, where $b \equiv \left(b_0, b_1, ..., b_n\right)^{\top} \in \mathbb{R}^{n+1}$, and $\{\psi_i \mid i = 0, ..., n\}$ is a set of basis functions. Thus, the regression in *Step 2 (ii)* is linear.

Integration in *Step 2 (i)* The formula (13) for evaluating the conditional expectation is consistent with a variety of numerical integration methods including Monte Carlo and deterministic (Gaussian quadrature and monomial) methods. We restrict attention to deterministic integration methods as they dominate the Monte Carlo method in terms of accuracy and cost in the context of the studied models; see Judd et al. (2011) for a comparison of Monte Carlo and deterministic integration methods.

Approximation method in *Step 2 (ii)* To implement regression in *Step 2 (ii)*, we must choose a norm for regression errors. An obvious choice is the L_2 norm that leads to the ordinary least-squares (OLS) method. However, if regressors are either collinear or poorly scaled, a least-squares (LS) problem is ill-conditioned, and the OLS method is numerically unstable. Judd, Maliar and Maliar (2011) describe a variety of approximation methods suitable for dealing with ill-conditioned problems. Such methods include least-squares methods using singular value decomposition and QR factorization, Tikhonov regularization, least-absolute deviations method, and principal component regression method.

Convergence criteria in *Step 2 (iii)* We focus on the convergence of the values of the policy function on the grid, rather than on the convergence of the coefficients of the policy function. The difference in the values of the capital policy function has an economic meaning (it informs us about the size of the error in the capital choice), while the difference in the coefficients has no economic meaning and depends on a specific choice of basis functions (for example, coefficients of Chebyshev polynomials are not equal to coefficients of ordinary polynomials). In all our experiments, the convergence of the coefficients of policy functions implied the convergence of the values on the grid and vice versa.

Procedure for updating the coefficients in *Step 2 (iv)* Fixed-point iteration is a derivative-free method, unlike time iteration and quasi-Newton methods, which are two other iterative schemes for finding fixed-point coefficients; see Judd (1998, pp. 553-558 and 103-119, respectively). To attain numerical stability, fixed-point iteration requires setting the damping parameter ξ to a small value. As a result, fixed-point iteration might need a larger number of iterations for convergence than do time-iteration and quasi-Newton methods, however, it might still have a smaller overall cost because of a much smaller per-iteration cost. Fixed-point iteration is particularly well suited for high-dimensional problems in which the cost of finding derivatives (Jacobian and Hessian) is prohibitive. In all our experiments, fixed-point iteration was numerically stable under appropriate damping.

4.3 Numerical experiments

In this section, we investigate the performance of CGA in the context of the representative-agent model.

4.3.1 Implementation details

Parameters, computational techniques, software and hardware We parameterize the model (8)–(10) by assuming $u(c_t) = \frac{c_t^{1-\gamma}-1}{1-\gamma}$ with $\gamma \in \left\{\frac{1}{5}, 1, 5\right\}$ and $f(k_t) = k_t^{\alpha}$ with $\alpha = 0.36$. We set $\beta = 0.99$, $\delta = 0.025$, $\rho = 0.95$ and $\sigma = 0.01$. We normalize the steady state of capital to one by assuming $A = \frac{1/\beta-(1-\delta)}{\alpha}$. The simulation length is $T = 10,000$, the damping parameter in (18) is $\xi = 0.1$, and the convergence parameter in (17) is $\varpi = 10^{-11}$. We parameterize the capital policy function using complete ordinary polynomials of degrees up to 5. We use a 10-node Gauss-Hermite quadrature rule in the formula (13) for approximating the conditional expectation; see Judd (1998, p. 261). We compute the regression coefficients using an LS method based on QR factorization. We construct a cluster grid using the agglomerative hierarchical algorithm with Ward's distance. We use MATLAB software, version 7.6.0.324 (R2008a) and a desktop computer ASUS with Intel(R) Core(TM)2 Quad CPU Q9400 (2.66 GHz), RAM 4MB.

Accuracy check We generate a new random draw of $10,200$ points and discard the first 200 points. At each point (k_τ, a_τ), we compute a Euler-equation error in a unit-free form by using a 10-node Gauss-Hermite quadrature rule, $\mathcal{E}(k_\tau, a_\tau) \equiv \sum_{j=1}^{J^{\text{test}}} \omega_j^{\text{test}} \cdot \left[\beta \frac{u'\left(c'_{\tau,j}\right)}{u'(c_\tau)} \left[1-\delta+a_\tau^{\rho} \exp\left(\varepsilon_j^{\text{test}}\right) A f'\left(k'_\tau\right)\right]\right] - 1$, where c_τ and $c'_{\tau,j}$ are defined similarly to c_m and $c'_{m,j}$ in (14) and (15), respectively. We report the mean and maximum of absolute value of $\mathcal{E}(k_\tau, a_\tau)$.

Initial guess The ergodic set is unknown before the model is solved. We initialize CGA using (arbitrary) initial guess $k_{t+1} = 0.95k_t + 0.05a_t$ (this guess matches the steady state level of capital equal to one). Given this initial guess, we simulate the model, construct the clusters and compute a first-degree polynomial solution on the constructed grid of the clusters' centers; we repeat this procedure one more time using the obtained solution as an initial guess. To compute polynomial approximations of degrees higher than 1, we use the cluster grid derived from the polynomial approximation of degree 1, and we use the coefficients vector obtained from the polynomial approximation of the previous degree.

4.3.2 Accuracy and speed of the benchmark CGA algorithm

In Table 1, we provide the results under the grid of $M = 25$ points. The accuracy of solutions delivered by CGA is comparable to the highest accuracy

attained in the related literature. Approximation errors decrease with each polynomial degree by one or more orders of magnitude. For the fifth-degree polynomials, the largest unit-free error in our least accurate solution is still less than 10^{-6} (see the experiment with high degree of risk aversion $\gamma = 5$). Most of the cost of CGA comes from the clustering routine (the time for constructing clusters twice is included in the total time for computing the polynomial solution of degree 1 and is about 18 seconds). Computing high-degree polynomial solutions is relatively fast (a few seconds) for a given grid. We performed sensitivity experiments in which we varied the number of clusters, recomputed the cluster grid iteratively a large number of times and modified the concept of distance between clusters. The results were robust to all modifications considered. We also tried to vary the number of nodes in the Gauss-Hermite quadrature rule, and we found 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 was somewhat lower). This result is in line with the finding of Judd (1992) that in the context of the standard growth model, even few quadrature nodes lead to very accurate solutions.

4.3.3 Overidentification versus collocation

To compute the results in Table 1, we use a grid that overidentifies the polynomial coefficients, namely, we use the same grid of 25 points for polynomial degrees from 1 to 5 (the number of polynomial coefficients ranges from 3 to 21). An alternative technique used in the related literature is collocation, when the number of grid points is the same as the number of polynomial terms, and the polynomial coefficients are identified exactly. In Table 2, we use collocation to recompute the solutions reported in Table 1.

The comparison of the results in Tables 1 and 2 indicates that collocation is not a good choice in the context of CGA. First, computing a separate cluster grid for each polynomial degree increases the cost. Second, our overidentifying grid generally leads to more accurate solutions than the collocation grid. Finally, collocation is less numerically stable than overidentification (CGA with collocation failed to converge under $\gamma = 5$ for the polynomial degrees 4 and 5). Collocation is designed for approximating smooth functions on hypercube domains using orthogonal polynomials and becomes fragile when we deviate from this case; see Judd (1992) for a discussion.

4.3.4 Autocorrection of the cluster grid

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

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

In our analysis, the cluster grid was always autocorrecting, however, there is no guarantee that this will be always the case. It might happen that in the presence of strong non-linearities and kinks in policy functions, we are stuck in a computational self-confirming equilibrium.[9] Furthermore, some models (for example, dynamic games) might have ergodic sets consisting of multiple recurrent classes. In those cases, we must train the algorithm to focus on the relevant recurrent class by imposing appropriate equilibrium restrictions such as monotonicity, concavity, continuity, steady state, etc. In addition, we must check the accuracy of solutions not only on a stochastic simulation but also on deterministic sets of points representing different areas of the state space; see Juillard and Villemot (2011) for examples of accuracy tests on deterministic sets of points.

4.3.5 Cluster grid versus Smolyak grid

Krueger and Kubler (2004), and Malin, Krueger and Kubler (2011) develop a projection method that relies on a Smolyak space grid. Like conventional projection methods, Smolyak's method operates on a hypercube domain (and, hence, the size of the domain grows exponentially with the dimensionality of

[9] An interesting case to explore would be a model with occasionally binding borrowing constraints. Christiano and Fisher (2000) show how projection methods could be used to solve such a model.

the state space). However, it uses a specific discretization of the hypercube domain which yields a sparse grid of carefully selected points (the number of points in the Smolyak grid grows only polynomially with the dimensionality of the state space).

We now compare the accuracy of solutions under the Smolyak and cluster grids.[10] We construct the Smolyak grid as described in Malin et al. (2011), namely, we use the interval for capital $[0.8, 1.2]$, and we use the interval for productivity $\left[\exp\left(-\frac{0.8}{1-\rho}\right), \exp\left(\frac{0.8}{1-\rho}\right)\right]$. The Smolyak grid has 13 points, so we use the same number of points in the cluster grid; the two grids are shown in Figures 4a and 4b, respectively. With 13 grid points, we can identify the coefficients in ordinary polynomials up to degree 3. In this case, we evaluate the accuracy of solutions not only on a stochastic simulation but also on a set of 100×100 points which are uniformly spaced on the same domain as the one used by Smolyak's method for finding a solution. The results are shown in Table 3.

In the test on a stochastic simulation, the cluster grid leads to considerably more accurate solutions than the Smolyak grid. This is because under the cluster grid, we fit a polynomial directly in the ergodic set, while under the Smolyak grid, we fit a polynomial in a larger rectangular domain and face a trade-off between the fit inside and outside the ergodic set. In the test on the rectangular domain, however, the Smolyak grid produces significantly smaller maximum errors than the cluster grid. This is because CGA is designed to be accurate in the ergodic set and its accuracy decreases more rapidly away from the 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 had the same regularities. These regularities are also observed in high-dimensional applications.[11]

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

[11] Kollmann et al. (2011) compare the accuracy of solutions produced by several solution methods, including the CGA algorithm introduced in the present paper and Smolyak's algorithm of Krueger and Kubler (2004) (see Maliar, Maliar and Judd, 2011, and Malin et al., 2011, for implementation details of the respective methods in the context of those models). Their comparison is performed using a collection of 30 real-business cycle models with up to 10 heterogeneous agents. Their findings are the same as ours: on the ergodic set and near the steady state, the CGA solutions are more accurate than the Smolyak

In some problems (such as, e.g., dynamic games), in order to have an accurate solution in the ergodic set, we must have a sufficiently accurate solution at the boundaries of the ergodic set (points that are not visited in equilibrium but which can communicate with points that are visited). In particular, different sets of perceptions on what occurs outside of a recurrent class can support different recurrent classes as "equilibria". We find that the accuracy range of CGA can be expanded by using the following technique: we increase the variance of exogenous shocks when simulating series for constructing clusters. This fattens up the cloud of simulated data and expands the solution domain. In our experiments, this technique increased accuracy outside the ergodic set at a cost of a moderate accuracy loss inside the ergodic set. The "fattened-up ergodic set" is still far smaller in large-scale problems than the conventional hypercube domain.

4.4 CGA in problems with high dimensionality

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

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 policy functions to approximate. The number of terms in an approximating polynomial function increases and we need to increase the number of grid points to identify the polynomial coefficients. The number of nodes in integration formulas increases. Finally, operating with large data sets can slow down computations or can lead to a memory congestion. If a solution method relies on product-rule constructions (of grids, integration nodes, derivatives,

solutions whereas the situation reverses for large deviations from the steady state.

etc.), the cost increases exponentially (curse of dimensionality) as is in the case of conventional projection methods such as a Galerkin method of Judd (1992). Below, we show that the cost of CGA grows at a relatively moderate rate.

Cost of constructing clusters We first assess how the cost of constructing clusters depends on the dimensionality of the problem. In Table 4, we report the time necessary for constructing $M = 3$, 30, 300 clusters under three simulation lengths $T = 1000$, 3000, $10,000$ with the number of countries ranging from $N = 1$ to $N = 200$. The cost of constructing clusters depends primarily on T. An increase in T by one order of magnitude (from 1000 to $10,000$) increases the clustering time by about two orders of magnitude. In turn, an increase in N by two orders of magnitude (from 1 to 100) only triplicates the clustering time. In the most expensive case, $N = 200$ and $T = 10,000$, the clustering time is around one minute. Given that clusters must be constructed just few times, we do not explore possibilities of reducing the cost of constructing clusters.[12]

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 5. We consider four alternative integration rules such as the Gauss-Hermite product rule with 2^N nodes, denoted by $Q(2)$, the monomial rule with $2N^2 + 1$ nodes, denoted by $M2$, the monomial rule with $2N$ nodes, denoted by $M1$, (see Judd, 1998, formulas 7.5.9–7.5.11), and the Gauss-Hermite rule with one node, denoted by $Q(1)$.

The accuracy of solutions here is similar to that we had for the one-agent model. For the polynomial approximations of degrees 1, 2 and 3, the errors are typically smaller than 0.1%, 0.01% and 0.001%, respectively. A specific integration method used plays only a minor role in the accuracy of solutions. For the polynomial approximation of degree 1, all the integration methods considered lead to virtually the same accuracy. For the polynomial approximation of degree 2, $Q(2)$, $M2$ and $M1$ lead to the approximation errors which are identical up to the fourth digit, while $Q(1)$ yields the errors which are $5 - 10\%$ larger. These regularities are robust to variations in the model's parameters such as the volatility and persistence of shocks and the

[12] A K-means clustering algorithm is a cheaper alternative to the hierarchical clustering algorithm used in our analysis. K-means clustering starts with K random clusters, and then moves observations between those clusters (with the aim of minimizing variability within clusters and maximizing variability between clusters). A drawback of K-means clustering is that it can give different clusters with each run.

degrees of risk aversion (see Table 8 in Judd, Maliar and Maliar, 2010, a working-paper version of the present paper).

The running time ranges from 30 seconds to 24 hours depending on the number of countries, the polynomial degree and the integration technique used. In particular, CGA was able to compute quadratic solutions to the models with up to 40 countries and linear solutions to the models with up to 200 countries when using inexpensive (monomial and one-node quadrature) integration rules. Thus, CGA was able to solve much larger problems than those studied in the related literature. A proper coordination between the choices of approximating function and integration technique is needed to make CGA cost-efficient. An example of such a coordination is a combination of a flexible second-degree polynomial with a cheap one-node Gauss-Hermite quadrature rule (as opposed to an inefficient combination of a rigid first-degree polynomial with expensive product integration formulas).

5 A new Keynesian model with the ZLB

In this section, we use CGA to solve a stylized new Keynesian model with Calvo-type price frictions and a Taylor (1993) rule. Our setup builds on the models considered in Christiano, Eichenbaum and Evans (2005), Smets and Wouters (2003, 2007), Del Negro et al. (2007). This literature estimates their models using the data on actual economies, while we use their parameter estimates and compute solutions numerically. We solve two versions of the model, one in which we allow for negative nominal interest rates and the other in which we impose a zero lower bound (ZLB) on nominal interest rates. Our setup has eight state variables. It is large scale in the sense that it is expensive to solve or even intractable under conventional global solution methods that rely on product rules.

The literature that finds numerical solutions to new Keynesian models typically relies on local perturbation solution methods or applies expensive global solution methods to low-dimensional problems. As for perturbation, most papers compute linear approximations, and some papers compute quadratic approximations (e.g., Kollmann, 2002, and Schmitt-Grohé and Uribe, 2007) or cubic approximations (e.g., Rudebusch and Swanson, 2008). Few papers use global solution methods; see, e.g., Adam and Billi (2006), Anderson, Kim and Yun (2010), and Adjemian and Juillard (2011). The above papers have at most 4 state variables and employ simplifying assumptions.[13]

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

Finally, Fernández-Villaverde, Posch and Rubio-Ramírez (2011) formulate and study a continuous-time version of the new Keynesian model.

5.1 The set up

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

Households The representative household solves

$$\max_{\{C_t, L_t, B_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma} - 1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta} - 1}{1+\vartheta}\right] \tag{19}$$

$$\text{s.t. } P_t C_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} + T_t = B_{t-1} + W_t L_t + \Pi_t, \tag{20}$$

where the initial condition $\left(B_0, \eta_{u,0}, \eta_{L,0}, \eta_{B,0}\right)$ is given; C_t, L_t, and B_t are consumption, labor and nominal bond holdings, respectively; P_t, W_t and R_t are the commodity price, nominal wage and (gross) nominal interest rate, respectively; $\eta_{u,t}$ and $\eta_{L,t}$ are exogenous preference shocks to the overall momentary utility and disutility of labor, respectively; $\eta_{B,t}$ is an exogenous premium in the return to bonds; T_t is lump-sum taxes; Π_t is the profit of intermediate-good firms; $\beta \in (0,1)$ is the discount factor; $\gamma > 0$ and $\vartheta > 0$ are the utility-function parameters. The processes for shocks are

$$\eta_{u,t+1} = \rho_u \eta_{u,t} + \epsilon_{u,t+1}, \qquad \epsilon_{u,t+1} \sim \mathcal{N}\left(0, \sigma_u^2\right), \tag{21}$$

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

$$\eta_{B,t+1} = \rho_B \eta_{B,t} + \epsilon_{B,t+1}, \qquad \epsilon_{B,t+1} \sim \mathcal{N}\left(0, \sigma_B^2\right), \tag{23}$$

where ρ_u, ρ_L, ρ_B are the autocorrelation coefficients, and σ_u, σ_L, σ_B are the standard deviations of disturbances.

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

$$\max_{Y_t(i)} \quad P_t Y_t - \int_0^1 P_t(i) Y_t(i) \, di \tag{24}$$

$$\text{s.t. } Y_t = \left(\int_0^1 Y_t(i)^{\frac{\varepsilon-1}{\varepsilon}} di\right)^{\frac{\varepsilon}{\varepsilon-1}}, \tag{25}$$

where (25) is a Dixit-Stigltz aggregator function with $\varepsilon \geq 1$.

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

$$\min_{L_t(i)} \quad \mathrm{TC}\left(Y_t\left(i\right)\right) = \left(1-v\right) W_t L_t\left(i\right) \tag{26}$$

$$\text{s.t. } Y_t\left(i\right) = \exp\left(\eta_{a,t}\right) L_t\left(i\right), \tag{27}$$

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

where $L_t\left(i\right)$ is the labor input; $\exp\left(\eta_{a,t}\right)$ is the productivity level; ρ_a is the autocorrelation coefficient, and σ_a is the standard deviation of the disturbance. The firms are subject to Calvo-type price setting: a fraction $1-\theta$ of the firms sets prices optimally, $P_t\left(i\right) = \widetilde{P}_t$, 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, $P_t\left(i\right) = P_{t-1}\left(i\right)$, for $i \in [0,1]$. A reoptimizing firm $i \in [0,1]$ maximizes the current value of the profit over the time when $\widetilde{P}_t$ remains effective,

$$\max_{\widetilde{P}_t} \quad \sum_{j=0}^{\infty} \beta^j \theta^j E_t \left\{ \Lambda_{t+j} \left[\widetilde{P}_t Y_{t+j}\left(i\right) - P_{t+j} \mathrm{mc}_{t+j} Y_{t+j}\left(i\right) \right] \right\} \tag{29}$$

$$\text{s.t. } Y_t\left(i\right) = Y_t \left(\frac{P_t\left(i\right)}{P_t} \right)^{-\varepsilon}, \tag{30}$$

where (30) is the demand for an intermediate good i following from (24), (25); Λ_{t+j} is the Lagrange multiplier on the household's budget constraint (20); mc_{t+j} is the real marginal cost of output at time $t+j$ (which is identical across the firms).

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

$$T_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} = P_t \frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)} + B_{t-1} + v W_t L_t, \tag{31}$$

where $\frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)} = G_t$ is government spending, $v W_t L_t$ is the subsidy to the intermediate-good firms, and $\eta_{G,t}$ is a government-spending shock,

$$\eta_{G,t+1} = \rho_G \eta_{G,t} + \epsilon_{G,t+1}, \qquad \epsilon_{G,t+1} \sim \mathcal{N}\left(0, \sigma_G^2\right), \tag{32}$$

where ρ_R is the autocorrelation coefficient, and σ_R is the standard deviation of disturbance.

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

$$\Phi_t \equiv R_* \left(\frac{R_{t-1}}{R_*}\right)^{\mu} \left[\left(\frac{\pi_t}{\pi_*}\right)^{\phi_\pi} \left(\frac{Y_t}{Y_{N,t}}\right)^{\phi_y}\right]^{1-\mu} \exp\left(\eta_{R,t}\right), \tag{33}$$

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

$$\eta_{R,t+1} = \rho_R \eta_{R,t} + \epsilon_{R,t+1}, \qquad \epsilon_{R,t+1} \sim \mathcal{N}\left(0, \sigma_R^2\right), \tag{34}$$

where ρ_R is the autocorrelation coefficient, and σ_R is the standard deviation of disturbance. When the ZLB is not imposed, the Taylor rule is $R_t = \Phi_t$.

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

$$\max_{\{C_t, L_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma} - 1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta} - 1}{1+\vartheta}\right] \tag{35}$$

$$\text{s.t. } C_t = \exp\left(\eta_{a,t}\right) L_t - G_t, \tag{36}$$

where $G_t = \frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)}$ is given, and $\eta_{u,t+1}$, $\eta_{L,t+1}$, $\eta_{a,t+1}$, and $\eta_{G,t}$ follow the processes (21), (22), (28), and (32), respectively. The FOCs of the problem (35), (36) imply that $Y_{N,t}$ depends only on exogenous shocks,

$$Y_{N,t} = \left[\frac{\exp\left(\eta_{a,t}\right)^{1+\vartheta}}{\left[\exp\left(\eta_{G,t}\right)\right]^{-\gamma} \exp\left(\eta_{L,t}\right)}\right]^{\frac{1}{\vartheta+\gamma}}. \tag{37}$$

5.2 Summary of equilibrium conditions

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

$$S_t = \frac{\exp\left(\eta_{u,t} + \eta_{L,t}\right)}{\exp\left(\eta_{a,t}\right)} L_t^{\vartheta} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}, \tag{38}$$

$$F_t = \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon-1} F_{t+1}\right\}, \tag{39}$$

$$\frac{S_t}{F_t} = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}, \tag{40}$$

$$\Delta_t = \left[(1-\theta)\left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{\varepsilon}{\varepsilon-1}} + \theta\frac{\pi_t^{\varepsilon}}{\Delta_{t-1}}\right]^{-1}, \tag{41}$$

$$C_t^{-\gamma} = \frac{\beta\exp\left(\eta_{B,t}\right)R_t}{\exp\left(\eta_{u,t}\right)}E_t\left[\frac{C_{t+1}^{-\gamma}\exp\left(\eta_{u,t+1}\right)}{\pi_{t+1}}\right], \tag{42}$$

$$Y_t = \exp\left(\eta_{a,t}\right)L_t\Delta_t, \tag{43}$$

$$C_t = \left(1-\frac{\overline{G}}{\exp\left(\eta_{G,t}\right)}\right)Y_t, \tag{44}$$

where the variables S_t and F_t are introduced for a compact representation of the profit-maximization condition of the intermediate-good firm and are defined recursively; $\pi_{t+1} \equiv \frac{P_{t+1}}{P_t}$ is the gross inflation rate between t and $t+1$; Δ_t is a measure of price dispersion across firms (also referred to as efficiency distortion). The conditions (38)–(44) correspond to (C17), (C18), (C23), (C33), (C3), (C27) and (C36) in the online Appendix C.

An interior equilibrium satisfies 8 equilibrium conditions (38)–(44), the Taylor rule (with or without the ZLB) in which Φ_t and $Y_{N,t}$ are given by (33) and (37), respectively, and 6 processes for exogenous shocks, (21)–(23), (28), (34) and (32). The 8 equilibrium conditions must be solved with respect to 8 unknowns $\{C_t, L_t, Y_t, \pi_t, \Delta_t, R_t, S_t, F_t\}$. There are 2 endogenous state variables, $\{\Delta_{t-1}, R_{t-1}\}$, and 6 exogenous state variables, $\left\{\eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{a,t}, \eta_{R,t}, \eta_{G,t}\right\}$.

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 parameters. We approximate numerically the policy functions for S_t, F_t and $C_t^{-\gamma}$ using the Euler equations (38), (39) and (42), respectively, and 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 CGA2 and CGA3, 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; see http://www.dynare.org). When solving the model with the ZLB by CGA, 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 ap-

proach 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 at 1). A detailed description of the methodology of our numerical analysis is provided in the online Appendix C.

Accuracy and cost of solutions In Table 6, we report the results for 3 experiments: in the first 2 experiments, we allow for negative net interest rates and consider two alternative values of the target inflation rate, $\pi_* = 1$ (a zero target net inflation rate) and $\pi_* = 1.0598$ (the estimate of Del Negro et al., 2007), and in the last experiment, we introduce the ZLB in the model with $\pi_* = 1$.

The perturbation methods are fast: the time necessary for computing both PER1 and PER2 is about 10 seconds. CGA is more costly: the computational time ranges from 6 to 25 minutes; this cost is modest given that we use MATLAB and a standard desktop computer. Our results for the multicountry model indicate that CGA would be tractable in much larger new Keynesian models.

The accuracy of PER1 is low. The approximation errors in some equilibrium conditions can be as large as 17% (see the maximum error of $10^{-0.76}$ for the model with $\pi_* = 1.0598$ in the table). The accuracy of PER2 depends on the model considered. When $\pi_* = 1$ and the ZLB is not imposed, PER2 is almost as accurate as CGA2 and has the approximation errors smaller than 2%. However, when the ZLB is imposed, the accuracy of PER2 deteriorates dramatically: the errors in some equilibrium conditions can reach 9%. In turn, CGA2 has the errors of less than 2% in all the experiments considered, and CGA3 has the errors that are smaller than 0.1% and 1% when the ZLB is not imposed and is imposed, respectively.

To appreciate how much the equilibrium quantities differ across the methods, we report the maximum percentage differences between variables produced by CGA3 and those produced by the other methods on a simulation of 10,000 observations. The regularities are similar to those we observed for the approximation errors. The difference between the series produced by PER1 and CGA3 can be larger than 13%; the difference between the series produced by PER2 and CGA3 depends on the model: it is less than 2% when $\pi_* = 1$ and the ZLB is not imposed but can be in excess of 9% when the ZLB is imposed; and finally, the difference between the series produced by CGA2 and CGA3 is relatively small in all models (1.27% at most). Generally, the supplementary variables S_t and F_t differ more across methods than such economically relevant variables as Y_t, L_t and C_t.

Economic importance of the ZLB In the table, we report the minimum and maximum values of R_t on a stochastic simulation, as well as a percentage number of periods in which $R_t \leq 1$. Under $\pi_* = 1$, the interest rate was as low as 0.9801, and the frequency of $R_t \leq 1$ was as large as 8%. In contrast, under $\pi_* = 1.0598$, the interest rate never got below 0.9922, and the frequency of $R_t \leq 1$ was just 0.13%. Thus, a negative net interest rate is a relatively frequent and economically significant event in a low- but not high-inflation economy.[14]

When the ZLB is not imposed, the properties of the interest rate (i.e., ranges of values for R_t and frequencies of $R_t \leq 1$) are similar under the CGA and perturbation methods. However, when the ZLB is imposed, such properties differ: the frequency of reaching the ZLB (i.e., $R_t = 1$) is about 20% smaller under the perturbation methods than under the CGA methods; see Table 6. These regularities are illustrated in Figure 5 using a fragment of a stochastic simulation. When the ZLB is not imposed, all methods predict 4 consecutive periods of negative (net) interest rates (see periods 3-7 in Figure 5a). When the ZLB is imposed, the CGA method predicts a zero interest rate in those 4 periods, while the perturbation methods predict a zero interest rate just in 1 period (this is true for both PER1 and PER2). The way we deal with the ZLB in the perturbation solution misleads the agents about the true state of the economy. To be specific, when we chop the interest rate at zero in the simulation procedure, agents perceive the drop in the interest rate as being small and respond by an immediate recovery. In contrast, under CGA, agents accurately perceive the drop in the interest rate as being large and respond by 4 consecutive periods of a zero net interest rate (which correspond to 4 consecutive periods of negative net interest rates predicted in the case when the ZLB is not imposed).

Lessons In new Keynesian models, local (perturbation) and global solution methods may produce qualitatively different results. This is not a hypothetical possibility but something we observe in a stylized model under empirically plausible parameterizations. When the ZLB is ignored, the accuracy of perturbation methods can be increased by using higher order approximations. When the ZLB is imposed, the accuracy depends critically on the way we deal with the ZLB. The approximation errors are large if we neglect the ZLB in the solution procedure and introduce it just in simulation. This is because we use one set of equations to solve the model, and we use another set of

[14] Chung, Laforte, Reifschneider, and Williams (2011) provide estimates of the incidence of the ZLB in the US economy. Christiano, Eichenbaum and Rebelo (2009) study the economic significance of the ZLB in the context of a similar model.

equations to simulate the model afterwords. Increasing the order of perturbation solutions will not fix this problem. We need global solution methods that can handle policy functions with kinks.

6 Conclusion

The ergodic set of a typical economic model is a tiny fraction of the hypercube domains, normally used by global solution methods. We propose the cluster-grid algorithm, CGA, which provides a simple way of solving dynamic economic models on their ergodic sets. Unlike perturbation methods, CGA can handle applications with strong non-linearities and kinks in policy functions (an example is a new Keynesian model with the ZLB). We combine the efficient choice of solution domain with other computational techniques that are particularly suitable for large-scale applications, namely, nonproduct (monomial and 1-node quadrature) integration rules, and a derivative-free fixed-point iteration method for computing the coefficients of policy functions. In combination, these techniques enable us to accurately solve economic models with hundreds of state variables using a desktop computer and MATLAB software. The speed of CGA can be substantially increased by using more powerful hardware and software, as well as parallelization techniques.

References

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

[2] Adjemian, S. 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.

[3] 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.

[4] 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.

[5] 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.

[6] Christiano, L., M. Eichenbaum and S. Rebelo, (2009). When is the government spending multiplier large? NBER Working Paper 15394.

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

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

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

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

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

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

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

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

[15] Fernández-Villaverde, J., O. Posch and J. Rubio-Ramírez, (2011). Solving the new Keynesian model in continuous time. Manuscript.

[16] Fernández-Villaverde, J. and J. Rubio-Ramírez, (2007). Estimating Macroeconomic Models: A Likelihood Approach. Review of Economic Studies 74, 1059-1087.

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

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

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

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

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

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

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

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

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

[26] Kollmann, R., S. Kim and J. Kim, (2011a). Solving the multi-country real business cycle model using a perturbation method. Journal of Economic Dynamics and Control 35, 203-206.

[27] 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.

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

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

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

[31] 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.

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

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

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

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

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

[37] Santos, M., (1999). Numerical solution of dynamic economic models, in: J. Taylor and M. Woodford (Eds.), Handbook of Macroeconomics, Amsterdam: Elsevier Science, pp. 312-382.

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

[39] 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.

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

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

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

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

Supplement to "A Cluster-Grid Algorithm: Solving Problems With High Dimensionality": Appendices

Kenneth L. Judd
Lilia Maliar
Serguei Maliar

Appendix A: A numerical example of implementing the agglomerative hierarchical clustering algorithm

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

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

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

Nearest-neighbor measure of distance The nearest-neighbor measure of distance between the clusters A and B is the distance between the closest pair of observations $x_i \in A$ and $y_j \in B$, i.e., $D(A,B) = \min\limits_{x_i \in A,\ y_j \in B} d(x_i, y_j)$.

Let $d(x_i, y_j) = \left[\left(x_i^1 - y_j^1\right)^2 + \left(x_i^2 - y_j^2\right)^2\right]^{1/2} \equiv d_{ij}$ be the Euclidean distance (2).

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

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

The smallest non-zero distance for the five observations in S_1 is $d_{13} = 0.5$. Thus, we merge observations (singleton clusters) 1 and 3 into one cluster and call the obtained cluster $\{1, 3\}$. The distances for the four resulting clusters $\{1, 3\}$, 2, 4, and 5, are shown in a matrix S_2,

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

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

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

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

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

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

Ward's measure of distance We now construct clusters using Ward's measure of distance (3). As an example, consider the distance between the singleton clusters 1 and 2, i.e., $D(1,2)$. The center of the cluster $\{1,2\}$ is $\overline{x}_{\{1,2\}} = \left(\overline{x}^1_{\{1,2\}}, \overline{x}^2_{\{1,2\}}\right) = (1.5, 1.75)$, and $SSD(1) = SSD(2) = 0$. Thus, we have

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

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

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

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

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

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

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

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

We have

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

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

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

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

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

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

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

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

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

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

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

Appendix B: The multicountry model

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

$$\max_{\left\{c_t^h, k_{t+1}^h\right\}_{t=0,...,\infty}^{h=1,...,N}} E_0 \sum_{h=1}^{N} \lambda^h \left[\sum_{t=0}^{\infty} \beta^t u^h\left(c_t^h\right)\right] \tag{B1}$$

subject to the aggregate resource constraint,

$$\sum_{h=1}^{N} c_t^h + \sum_{h=1}^{N} k_{t+1}^h = \sum_{h=1}^{N} k_t^h (1-\delta) + \sum_{h=1}^{N} a_t^h A f^h \left(k_t^h\right), \tag{B2}$$

where the initial condition $\left\{k_0^h, a_0^h\right\}^{h=1,...,N}$ is given; E_t is the operator of conditional expectation; c_t^h, k_t^h, a_t^h and λ^h are, respectively, consumption, capital, productivity level and welfare weight of a country $h \in \{1, ..., N\}$; $\beta \in (0,1)$ is the discount factor; $\delta \in (0,1]$ is the depreciation rate; A is the normalizing constant in the production function. The utility and production functions, u^h and f^h, respectively, are increasing, concave and continuously differentiable. The process for the productivity level of country h is given by

$$\ln a_{t+1}^h = \rho \ln a_t^h + \epsilon_{t+1}^h, \tag{B3}$$

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,

$$u'(c_t) = \beta E_t \left\{ u'(c_{t+1}) \left[1 - \delta + a_{t+1}^h A f'\left(k_{t+1}^h\right)\right] \right\}, \tag{B4}$$

where u' and f' denote the derivatives of u and f, respectively. Thus, the planner's solution is determined by the process for shocks $(B3)$, the resource constraint $(B2)$, and the set of Euler equations $(B4)$.

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

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

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

where b^h is a vector of coefficients. We rewrite the Euler equation $(B4)$ as

$$k_{t+1}^h = E_t \left\{ \beta \frac{u'\left(c_{t+1}\right)}{u'\left(c_t\right)} \left[1 - \delta + a_{t+1}^h A f'\left(k_{t+1}^h\right)\right] k_{t+1}^h \right\}. \tag{B5}$$

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

The steps of the CGA algorithm here are similar to those described in Section 4.2 for the one-agent model. However, we now iterate on N policy functions of heterogenous countries instead of just one policy function of the representative agent. That is, we make an initial guess on N coefficients vectors $\left\{b^h\right\}^{h=1,...,N}$, approximate N conditional expectations in *Step 2 (i)* and run N regressions in *Step 2 (ii)*. The damping parameter in (18) is $\xi = 0.1$, and the convergence parameter in (17) is $\varpi = 10^{-8}$. In Stage 2, we evaluate the size of Euler equation errors on a stochastic simulation of length $T^{\text{test}} = 10,200$ (we discard the first 200 observations to eliminate the effect of initial condition). To test the accuracy of solutions, we use the Gauss-Hermite quadrature product rule $Q\left(2\right)$ for N up to 12, use the monomial rule $M2$ for N from 12 to 20, and use the monomial rule $M1$ for N larger than 20. We use the same values of the parameters in the multicountry model as in the one-agent model; in particular, we assume $\gamma = 1$.

Appendix C: The new Keynesian model

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

Households The FOCs of the household's problem (19)–(23) with respect to C_t, L_t and B_t are

$$\Lambda_t = \frac{\exp\left(\eta_{u,t}\right) C_t^{-\gamma}}{P_t}, \tag{C1}$$

$$\exp\left(\eta_{u,t} + \eta_{L,t}\right) L_t^{\vartheta} = \Lambda_t W_t, \tag{C2}$$

$$\exp\left(\eta_{u,t}\right) C_t^{-\gamma} = \beta \exp\left(\eta_{B,t}\right) R_t E_t \left[\frac{\exp\left(\eta_{u,t+1}\right) C_{t+1}^{-\gamma}}{\pi_{t+1}}\right], \tag{C3}$$

where Λ_t is the Lagrange multiplier associated with the household's budget constraint (20). After combining (C1) and (C2), we get

$$\exp\left(\eta_{L,t}\right) L_t^{\vartheta} C_t^{\gamma} = \frac{W_t}{P_t}. \tag{C4}$$

Final-good producers The FOC of the final-good producer's problem (24), (25) with respect to $Y_t(i)$ yields the demand for the ith intermediate good

$$Y_t(i) = Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon}. \tag{C5}$$

Substituting the condition (C5) into (25), we obtain

$$P_t = \left(\int_0^1 P_t(i)^{1-\varepsilon} di \right)^{\frac{1}{1-\varepsilon}}. \tag{C6}$$

Intermediate-good producers The FOC of the cost-minimization problem (26)–(28) with respect to $L_t(i)$ is

$$\Theta_t = \frac{(1-v) W_t}{\exp\left(\eta_{a,t}\right)}, \tag{C7}$$

where Θ_t is the Lagrange multiplier associated with (27). The derivative of the total cost in (26) is the nominal marginal cost, $\mathrm{MC}_t(i)$,

$$\mathrm{MC}_t(i) \equiv \frac{d\mathrm{TC}\left(Y_t(i)\right)}{dY_t(i)} = \Theta_t. \tag{C8}$$

The conditions (C7) and (C8) taken together imply that the real marginal cost is the same for all firms,

$$\mathrm{mc}_t(i) = \frac{(1-v)}{\exp\left(\eta_t^a\right)} \cdot \frac{W_t}{P_t} = \mathrm{mc}_t. \tag{C9}$$

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

$$E_t \sum_{j=0}^{\infty} (\beta\theta)^j \Lambda_{t+j} Y_{t+j} P_{t+j}^{\varepsilon+1} \left[\frac{\widetilde{P}_t}{P_{t+j}} - \frac{\varepsilon}{\varepsilon - 1} \mathrm{mc}_{t+j} \right] = 0 \tag{C10}$$

From the household's FOC (C1), we have

$$\Lambda_{t+j} = \frac{\exp\left(\eta_{u,t+j}\right) C_{t+j}^{-\gamma}}{P_{t+j}}. \tag{C11}$$

Substituting (C11) into (C10), we get

$$E_t\sum_{j=0}^{\infty}(\beta\theta)^j\exp\left(\eta_{u,t+j}\right)C_{t+j}^{-\gamma}Y_{t+j}P_{t+j}^{\varepsilon}\left[\frac{\widetilde{P}_t}{P_{t+j}}-\frac{\varepsilon}{\varepsilon-1}\mathrm{mc}_{t+j}\right]=0. \tag{C12}$$

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

$$\chi_{t,j}\equiv\begin{cases}1 \text{ if } j=0\\ \frac{1}{\pi_{t+j}\cdot\pi_{t+j-1}\cdots\pi_{t+1}} \text{ if } j\geq 1\end{cases}. \tag{C13}$$

Then $\chi_{t,j}=\chi_{t+1,j-1}\cdot\frac{1}{\pi_{t+1}}$ for $j>0$. Therefore, (C12) becomes

$$E_t\sum_{j=0}^{\infty}(\beta\theta)^j\exp\left(\eta_{u,t+j}\right)C_{t+j}^{-\gamma}Y_{t+j}\chi_{t,j}^{-\varepsilon}\left[\widetilde{\widetilde{p}}_t\chi_{t,j}-\frac{\varepsilon}{\varepsilon-1}\mathrm{mc}_{t+j}\right]=0, \tag{C14}$$

where $\widetilde{\widetilde{p}}_t\equiv\frac{\widetilde{P}_t}{P_t}$. We express $\widetilde{\widetilde{p}}_t$ from (C14) as follows

$$\widetilde{\widetilde{p}}_t=\frac{E_t\sum_{j=0}^{\infty}(\beta\theta)^j\exp\left(\eta_{u,t+j}\right)C_{t+j}^{-\gamma}Y_{t+j}\chi_{t,j}^{-\varepsilon}\frac{\varepsilon}{\varepsilon-1}\mathrm{mc}_{t+j}}{E_t\sum_{j=0}^{\infty}(\beta\theta)^j\exp\left(\eta_{u,t+j}\right)C_{t+j}^{-\gamma}Y_{t+j}\chi_{t,j}^{1-\varepsilon}}\equiv\frac{S_t}{F_t}. \tag{C15}$$

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

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

Substituting mc_t from $(C9)$ into the above recursive formula for S_t, we have

$$S_t = \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t \frac{(1-v)}{\exp\left(\eta_{a,t}\right)} \cdot \frac{W_t}{P_t} + \beta\theta E_t \left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}. \quad \text{(C16)}$$

Substituting $\frac{W_t}{P_t}$ from (C4) into (C16), we get

$$S_t = \frac{\varepsilon}{\varepsilon - 1} \exp\left(\eta_{u,t}\right) Y_t \frac{(1-v)}{\exp\left(\eta_{a,t}\right)} \cdot \exp\left(\eta_{L,t}\right) L_t^{\vartheta} + \beta\theta E_t \left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}. \quad \text{(C17)}$$

For F_t, the corresponding recursive formula is

$$F_t = \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t + \beta\theta E_t \left\{\pi_{t+1}^{\varepsilon-1} F_{t+1}\right\}. \quad \text{(C18)}$$

Aggregate price relationship The condition (C6) can be rewritten as

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

$$\left[\int_{\text{reopt.}} P_t\left(i\right)^{1-\varepsilon} di + \int_{\text{non-reopt.}} P_t\left(i\right)^{1-\varepsilon} di\right]^{\frac{1}{1-\varepsilon}}, \quad \text{(C19)}$$

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

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

$$\int_{\text{non-reopt.}} P_t\left(i\right)^{1-\varepsilon} di = \int_0^1 \theta P\left(j\right)^{1-\varepsilon} \omega_{t-1}\left(j\right) dj = \theta P_{t-1}^{1-\varepsilon}. \quad \text{(C20)}$$

Substituting (C20) into (C19) and using the fact that all reoptimizers set $\widetilde{P}_t^{1-\varepsilon}$, we get

$$P_t = \left[\left(1-\theta\right)\widetilde{P}_t^{1-\varepsilon} + \theta P_{t-1}^{1-\varepsilon}\right]^{\frac{1}{1-\varepsilon}}. \quad \text{(C21)}$$

We divide both sides of (C21) by P_t,

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

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

$$\widetilde{\widetilde{p}}_t = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}. \tag{C22}$$

Combining (C22) and (C15), we obtain

$$\frac{S_t}{F_t} = \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}. \tag{C23}$$

Aggregate output Let us define aggregate output

$$\overline{Y}_t \equiv \int_0^1 Y_t(i)\,di = \int_0^1 \exp\left(\eta_{a,t}\right) L_t(i)\,di = \exp\left(\eta_{a,t}\right) L_t, \tag{C24}$$

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 (C5) into (C24) to get

$$\overline{Y}_t = \int_0^1 Y_t \left(\frac{P_t(i)}{P_t}\right)^{-\varepsilon} di = Y_t P_t^{\varepsilon} \int_0^1 P_t(i)^{-\varepsilon}\,di. \tag{C25}$$

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

$$\left(\overline{P}_t\right)^{-\varepsilon} \equiv \int_0^1 P_t(i)^{-\varepsilon}\,di. \tag{C26}$$

Substituting (C24) and (C26) into (C25) gives us

$$Y_t \equiv \overline{Y}_t \left(\frac{\overline{P}_t}{P_t}\right)^{\varepsilon} = \exp\left(\eta_{a,t}\right) L_t \Delta_t, \tag{C27}$$

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

$$\Delta_t \equiv \left(\frac{\overline{P}_t}{P_t}\right)^{\varepsilon}. \tag{C28}$$

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 (C21), the variable $\overline{P}_t$, defined in (C26), satisfies

$$\overline{P}_t = \left[(1-\theta)\,\widetilde{P}_t^{-\varepsilon} + \theta\left(\overline{P}_{t-1}\right)^{-\varepsilon}\right]^{-\frac{1}{\varepsilon}}. \tag{C29}$$

Using (C29) in (C28), we get

$$\Delta_t = \left(\frac{\left[(1-\theta) \widetilde{P}_t^{-\varepsilon} + \theta \left(\overline{P}_{t-1} \right)^{-\varepsilon} \right]^{-\frac{1}{\varepsilon}}}{P_t} \right)^{\varepsilon}. \tag{C30}$$

This implies

$$\Delta_t^{\frac{1}{\varepsilon}} = \left[(1-\theta) \left(\frac{\widetilde{P}_t}{P_t} \right)^{-\varepsilon} + \theta \left(\frac{\overline{P}_{t-1}}{P_t} \right)^{-\varepsilon} \right]^{-\frac{1}{\varepsilon}}. \tag{C31}$$

In terms of $\widetilde{\widetilde{p}}_t \equiv \frac{\widetilde{P}_t}{P_t}$, the condition (C31) can be written as

$$\Delta_t = \left[(1-\theta) \widetilde{\widetilde{p}}_t^{-\varepsilon} + \theta \frac{\overline{P}_{t-1}^{-\varepsilon}}{P_t^{-\varepsilon}} \cdot \frac{P_{t-1}^{-\varepsilon}}{P_{t-1}^{-\varepsilon}} \right]^{-1}. \tag{C32}$$

By substituting $\widetilde{\widetilde{p}}_t$ from (C22) into (C32), we obtain the law of motion for Δ_t,

$$\Delta_t = \left[(1-\theta) \left[\frac{1-\theta \pi_t^{\varepsilon-1}}{1-\theta} \right]^{-\frac{\varepsilon}{1-\varepsilon}} + \theta \frac{\pi_t^{\varepsilon}}{\Delta_{t-1}} \right]^{-1}. \tag{C33}$$

Aggregate resource constraint Combining the household's budget constraint (20) with the government budget constraint (31), we have the aggregate resource constraint

$$P_t C_t + P_t \frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)} = (1-v) W_t L_t + \Pi_t. \tag{C34}$$

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

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

where $P_t Y_t = \int_0^1 P_t(i) Y_t(i)\, di$ follows by a zero-profit condition of the final-good firms. Hence, (C34) can be rewritten as

$$P_t C_t + P_t \frac{\overline{G}}{\exp\left(\eta_{G,t}\right)} Y_t = P_t Y_t. \tag{C35}$$

In real terms, the aggregate resource constraint (C35) becomes

$$C_t = \left(1 - \frac{\overline{G}}{\exp\left(\eta_{G,t}\right)} \right) Y_t. \tag{C36}$$

Equilibrium conditions The conditions (39)–(44) in the main text correspond to the conditions (C18), (C23), (C33), (C3), (C27) and (C36) in the present appendix. The condition (38) in the main text follows from (C17) under the additional assumption $\frac{\varepsilon}{\varepsilon-1}(1-v)=1$ which ensures that the model admits a deterministic steady state (this assumption is commonly used in the related literature; see, e.g., Christiano et al. 2009).

Calibration procedure Most of the parameters are calibrated using the estimates of Del Negro et al. (2007, Table 1, column "DSGE posterior"); namely, we assume $\gamma=1$ and $\vartheta=2.09$ in the utility function (19); $\phi_y=0.07$, $\phi_\pi=2.21$, and $\mu=0.82$ in the Taylor rule (33); $\varepsilon=4.45$ in the production function of the final-good firm (25); $\theta=0.83$ (the fraction of the intermediate-good firms affected by price stickiness); $\overline{G}=0.23$ in the government budget constraint (31); and $\rho_a=0.2$, $\rho_u=0.92$, $\rho_G=0.95$, $\sigma_a=0.82\%$, $\sigma_u=0.54\%$ and $\sigma_G=0.38\%$ in the processes for shocks (21), (28) and (32). From Smets and Wouters (2007), we take the values of $\rho_B=0.23$, $\rho_R=0.15$, $\sigma_B=0.22\%$ and $\sigma_R=0.24\%$ in the processes for shocks (23) and (34). Finally, from Smets and Wouters (2003), we take the value of $\rho_L=0.881$ in the process for shock (22), however, we use the value of $\sigma_L=0.6\%$, which is lower than their estimate of $\sigma_L=3.818\%$ (the latter estimate leads to an excessive volatility in the model). The above parameterization leads to the output volatility of around 3%, which is grossly consistent with the data on actual economies.

We set the discount factor at $\beta=0.99$. To parameterize the Taylor rule (33), 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, Table 1, column "DSGE posterior").

Solution procedure The CGA algorithm for the new Keynesian model is similar to the one described in Section 4.2 for the neoclassical growth model. We approximate the policy functions $S_t=S(\varkappa_t)$, $F_t=F(\varkappa_t)$ and $C_t^{-\gamma}=\mathrm{MU}(\varkappa_t)$, where $\varkappa_t=\left\{\Delta_{t-1},R_{t-1},\eta_{u,t},\eta_{L,t},\eta_{B,t},\eta_{a,t},\eta_{R,t},\eta_{G,t}\right\}$ is a set of current state variables. We parameterize such functions with flexible functional forms $S(\varkappa_t)\approx\widehat{S}\left(\varkappa_t;b^S\right)$, $F(\varkappa_t)\approx\widehat{F}\left(\varkappa_t;b^F\right)$, $\mathrm{MU}(\varkappa_t)\approx\widehat{\mathrm{MU}}\left(\varkappa_t;b^{\mathrm{MU}}\right)$, where b^S, b^F and b^{MU} are the coefficients vectors. To approximate the policy functions, we use the family of ordinary polynomials. We solve for the rest of the variables analytically: given S_t, F_t and $C_t^{-\gamma}$, we find π_t, Δ_t, Y_t and L_t from (40), (41), (44) and (43), respectively. To compute the conditional expectations in the Euler equations (38), (39) and (42), we use the 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 policy functions and for the grid of clusters). After the initial CGA solution was computed, we reconstruct the clusters and repeat the solution procedure (we checked that the subsequent reconstructions of the cluster grid do not improve the accuracy of solutions).

The simulation length is $T = 10,000$, and the number of clusters is $M = 1000$. The damping parameter in (18) is $\xi = 0.1$, and the convergence parameter in (17) is $\varpi = 10^{-7}$. We compute approximation errors on a stochastic simulation of 10,200 observations (we eliminate the first 200 observations). In the test, we use the monomial rule $M2$ with $2N^2 + 1$ nodes which is more accurate than the rule $M1$ used in the solution procedure. Dynare does not evaluate the accuracy of perturbation solutions. We wrote a MATLAB routine that simulates the perturbation solutions and evaluates their accuracy using the Dynare's representation of the state space which includes the current endogenous state variables $\{\Delta_{t-1}, R_{t-1}\}$, the past exogenous state variables $\left\{\eta_{u,t-1}, \eta_{L,t-1}, \eta_{B,t-1}, \eta_{a,t-1}, \eta_{R,t-1}, \eta_{G,t-1}\right\}$ and the current disturbances $\{\epsilon_{u,t}, \epsilon_{L,t}, \epsilon_{B,t}, \epsilon_{a,t}, \epsilon_{R,t}, \epsilon_{G,t}\}$. We found that the CGA and perturbation solutions are very close to each other when the volatility of shock is small, namely, the maximum difference between solutions produced by the two methods was less than 0.001% when the standard deviation of all shocks was set at 0.01%. By construction, the CGA solutions have non-zero errors only in the Euler equations, while the perturbation solutions have non-zero errors in all equilibrium conditions.

References

[1] Christiano, L., M. Eichenbaum, and S. Rebelo, (2009). When is the government spending multiplier large? NBER Working Paper 15394.

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

[3] 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.

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

Table 1. Accuracy and speed of the CGA algorithm in the one-agent model: 25 clusters. [a]

Polynomial Degree	γ = 1/5			γ = 1			γ = 5		
	ε_{mean}	ε_{mean}	*CPU*	ε_{mean}	ε_{max}	*CPU*	ε_{mean}	ε_{max}	*CPU*
1st	-4,88	-3,92	22,41	-4,29	-3,33	19,67	-3,32	-2,30	19,16
2nd	-6,58	-5,34	0,81	-6,10	-4,86	0,35	-4,86	-3,69	0,19
3rd	-8,15	-6,49	0,40	-7,53	-5,90	0,13	-6,16	-4,60	0,13
4th	-9,44	-7,81	0,19	-8,82	-6,96	0,11	-7,16	-5,36	0,10
5th	-10,03	-8,40	0,26	-9,90	-8,01	0,14	-8,26	-6,37	1,93

[a] ε_{mean} and ε_{max} are, respectively, the average and maximum absolute unit-free Euler equation errors (in log10 units) on a stochastic simulation of 10,000 observations; *CPU* is the time necessary for computing a solution (in seconds); γ is the coefficient of relative risk aversion.

Table 2. Accuracy and speed of the CGA algorithm in the one-agent model: collocation. [a]

Polynomial Degree	γ = 1/5			γ = 1			γ = 5		
	ε_{mean}	ε_{max}	*CPU*	ε_{mean}	ε_{max}	*CPU*	ε_{mean}	ε_{max}	*CPU*
1st	-4,86	-3,87	18,63	-4,34	-3,36	18,45	-3,39	-2,40	18,39
2nd	-6,53	-5,26	9,27	-6,09	-4,84	9,00	-4,81	-3,57	9,00
3rd	-8,05	-6,50	9,30	-7,45	-6,00	9,05	-5,92	-4,55	9,04
4th	-8,77	-7,17	9,53	-8,14	-6,53	9,25	failed to converge		
5th	-9,88	-8,29	10,36	-9,17	-7,64	10,19			

[a] ε_{mean} and ε_{max} are, respectively, the average and maximum absolute unit-free Euler equation errors (in log10 units) on a stochastic simulation of 10,000 observations; *CPU* is the time necessary for computing a solution (in seconds); γ is the coefficient of relative risk aversion.

Table 3. Accuracy and speed in the one-agent model: Smolyak grid versus cluster grid. [a]

Polynomial Degree	Accuracy Test on a Stochastic Simulation				Accuracy Test on a Tensor-Product Grid			
	Smolyak Grid		Cluster Grid		Smolyak Grid		Cluster Grid	
	ε_{mean}	ε_{max}	ε_{mean}	ε_{max}	ε_{mean}	ε_{max}	ε_{mean}	ε_{max}
1st	-3,31	-2,94	-4,28	-3,27	-3,25	-2,54	-3,27	-2,39
2nd	-4,74	-4,17	-6,07	-4,81	-4,32	-3,80	-4,40	-3,26
3rd	-5,27	-5,13	-7,44	-5,87	-5,39	-4,78	-5,40	-4,10

[a] ε_{mean} and ε_{max}, respectively, the average and maximum absolute unit-free Euler equation errors (in log10 units) on a stochastic simulation of 10,000 observations; *CPU* is the time necessary for computing a solution (in seconds). The number of grid points in cluster grid is the same as that in the Smolyak grid, and is equal to 13.

Table 4. Cost of constructing clusters depending on the number of countries *N*.

Simulation Length	Number of Clusters	Time Needed to Construct Clusters (in seconds)							
		N = 1	*N* = 2	*N* = 6	*N* = 10	*N* = 20	*N* = 40	*N* = 100	*N* = 200
T = 1000	*M* = 3	0.07	0.08	0.09	0.11	0.12	0.18	0.39	1.18
	M = 30	0.08	0.08	0.09	0.12	0.13	0.24	0.54	1.35
	M = 300	0.09	0.10	0.12	0.15	0.31	0.83	1.85	3.23
T = 3000	*M* = 3	0.83	0.84	0.83	0.84	1.04	1.42	2.77	4.50
	M = 30	0.80	0.81	0.85	0.86	1.13	1.54	3.08	5.15
	M = 300	0.83	0.86	1.00	1.08	1.92	3.11	7.06	11.73
T = 10,000	*M* = 3	8.93	8.87	9.18	9.51	12.05	17.36	31.45	42.08
	M = 30	8.99	8.94	9.28	9.77	12.46	17.81	31.87	44.23
	M = 300	9.05	9.33	10.08	10.91	14.78	22.40	43.76	66.37

Table 5. Accuracy and speed in the multicountry model depending on the integration method used. [a]

Number of Countries	Polyn. Degree	Number of Coefficients [b]	*M*	Integration Method											
				Q(2)			*M2*			*M1*			*Q(1)*		
				$\mathcal{E}_{mean}$	$\mathcal{E}_{max}$	*CPU*	$\mathcal{E}_{mean}$	$\mathcal{E}_{max}$	*CPU*	$\mathcal{E}_{mean}$	$\mathcal{E}_{max}$	*CPU*	$\mathcal{E}_{mean}$	$\mathcal{E}_{max}$	*CPU*
N = 2	1st	5	300	-4.09	-3.19	38	-4.09	-3.19	53	-4.09	-3.19	44	-4.07	-3.19	45
	2nd	15		-5.45	-4.51	108	-5.45	-4.51	150	-5.45	-4.51	114	-5.06	-4.41	85
	3rd	35		-6,51	-5,29	237	-6,51	-5,29	398	-6,51	-5,29	212	-5,17	-4,92	121
N = 4	1st	9	300	-4.13	-3.15	63	-4.13	-3.15	120	-4.13	-3.15	50	-4.11	-3.16	39
	2nd	45		-5.47	-4.32	287	-5.47	-4.32	517	-5.47	-4.32	206	-4.95	-4.23	90
N = 6	1st	13	300	-4.18	-3.21	222	-4.18	-3.21	232	-4.18	-3.21	68	-4.16	-3.22	42
	2nd	91		-5.51	-4.38	1282	-5.51	-4.38	1440	-5.51	-4.38	301	-4.93	-4.29	97
N = 8	1st	17	300	-4.20	-3.25	947	-4.20	-3.25	468	-4.20	-3.25	114	-4.18	-3.26	44
	2nd	153		-5.49	-4.51	9511	-5.49	-4.51	3774	-5.49	-4.51	422	-4.91	-4.34	109
N = 10	1st	21	400	-	-	-	-4.20	-3.24	1090	-4.20	-3.24	182	-4.18	-3.25	59
	2nd	231		-	-	-	-5.46	-4.50	12503	-5.46	-4.50	970	-4.90	-4.33	191
N = 12	1st	25	400	-	-	-	-4.21	-3.28	1403	-4.21	-3.28	233	-4.19	-3.29	63
	2nd	325		-	-	-	-5.23	-4.30	69025	-5.23	-4.30	1307	-4.88	-4.34	226
N = 16	1st	33	1000	-	-	-	-	-	-	-4.22	-3.29	843	-4.19	-3.29	175
	2nd	561		-	-	-	-	-	-	-5.44	-4.38	6790	-4.88	-4.27	1058
N = 20	1st	41	1000	-	-	-	-	-	-	-4.21	-3.29	1238	-4.17	-3.28	184
	2nd	861		-	-	-	-	-	-	-5.08	-4.17	16895	-4.83	-4.10	1911
N = 30	1st	61	4000	-	-	-	-	-	-	-4.23	-3.31	13985	-4.19	-3.29	3529
	2nd	1891		-	-	-	-	-	-	-	-	-	-4.86	-4.54	36304
N = 40	1st	81	4000	-	-	-	-	-	-	-4.23	-3.31	19043	-4.19	-3.29	5321
	2nd	3321		-	-	-	-	-	-	-	-	-	-4.86	-4.48	87748
N=100	1st	201	1000	-	-	-	-	-	-	-4.09	-3.24	38782	-4.06	-3.23	2174
N=200	1st	401	1000	-	-	-	-	-	-	-	-	-	-3.97	-3.20	6316

[a] $\mathcal{E}_{mean}$ and $\mathcal{E}_{max}$ are, respectively, the average and maximum absolute unit-free Euler equation errors (in log10 units) on a stochastic simulation of 10,000 observations; *CPU* is the time necessary for computing a solution (in seconds); *M* is the number of clusters*; Q(2)* and *Q(1)* are the Gauss-Hermite product rule with 2^N and 1 nodes, respectively, and *M2* and *M1* are the monomial rules with $2N^2+1$ and $2N$ nodes, respectively; [b] In the policy function of 1 country.

Table 6. The new Keynesian model: the CGA algorithm versus perturbation. [a]

Statistic	$\pi^* = 1$				$\pi^* = 1.0598$				$\pi^* = 1$ and ZLB			
	PER1	PER2	CGA2	CGA3	PER1	PER2	CGA2	CGA3	PER1	PER2	CGA2	CGA3
Running time												
CPU	9		363	664	9		802	1401	9		445	914
Absolute errors across optimality conditions												
$\mathcal{E}_{mean}$	-3,05	-3,81	-4,15	-4,26	-2,90	-3,60	-4,22	-4,27	-2,99	-3,40	-3,98	-4,05
$\mathcal{E}_{max}$	-0,89	-1,75	-1,85	-3,14	-0,76	-1,56	-1,77	-3,08	-0,90	-1,05	-1,93	-2,06
Interest rate properties												
R_{min}	0,9826	0,9806	0,9801	0,9804	0,9942	0,9924	0,9922	0,9923	1	1	1	1
R_{max}	1,0402	1,0382	1,0391	1,0380	1,0638	1,0615	1,0629	1,0612	1,0402	1,0382	1,0391	1,0394
Freq(R≤1),%	8,1953	8,1267	8,2737	8,4600	0,0980	0,1274	0,1372	0,1372	6,7836	6,6562	8,6266	8,3423
Difference between time series produced by the method in the given column and CGA3												
dif(*R),%*	0,23	0,05	0,11	0	0,43	0,38	0,17	0	0,90	0,94	0,14	0
dif(Δ),%	2,16	0,36	0,13	0	6,88	4,88	0,23	0	2,09	0,62	0,06	0
dif(*S),%*	8,29	1,54	0,82	0	13,09	5,28	1,17	0	11,17	9,17	1,27	0
dif(*F),%*	2,20	0,26	0,27	0	8,01	3,53	0,48	0	4,82	3,86	0,61	0
dif(*C),%*	1,35	0,18	0,17	0	4,21	2,92	0,35	0	3,22	3,58	1,06	0
dif(*Y),%*	1,36	0,18	0,17	0	4,21	2,92	0,35	0	3,25	3,59	1,06	0
dif(Y_N),%	0,04	0,00	0,00	0	0,04	0,00	0,00	0	0,04	0,00	0,00	0
dif(*L),%*	3,22	0,14	0,24	0	3,87	1,96	0,39	0	4,66	3,61	1,04	0
dif(π),%	0,56	0,06	0,21	0	0,54	0,28	0,36	0	0,98	0,86	0,19	0

[a] $\mathcal{E}_{mean}$ and $\mathcal{E}_{max}$ are, respectively, the average and maximum absolute percentage errors (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 seconds); PER1 and PER2 are the 1st- and 2nd-order perturbation solutions, respectively; CGA2 and CGA3 are 2nd- and 3d-degree CGA polynomial solutions, respectively; R_{min} and R_{max} are, respectively, the minimum and maximum gross nominal interest rates across 10,000 simulated periods; *Freq(R≤1)* is a percentage number of periods in which *R≤1*; dif(*X),%* is maximum absolute percentage difference between time series for variable *X* produced by the method in the given column and CGA3.

Figure 1a. The ergodic set

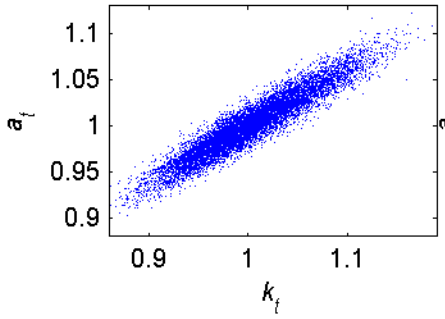

Figure 1b. Four clusters

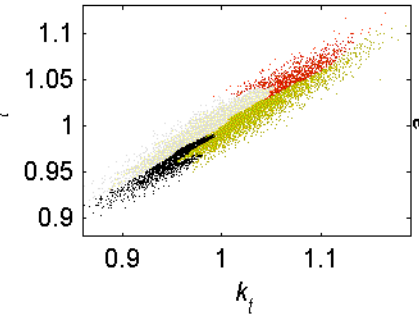

Figure 1c. The clusters' centers

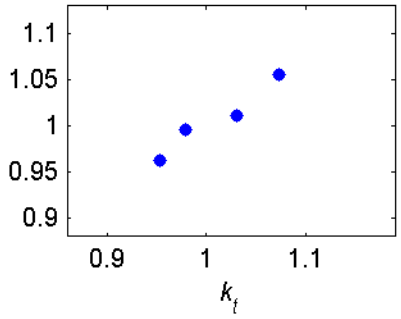

Figure 2a. Normalized ergodic set

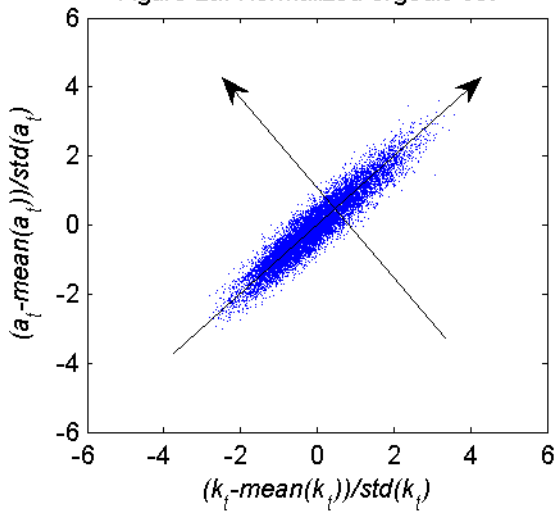

Figure 2b. PCs of the ergodic set

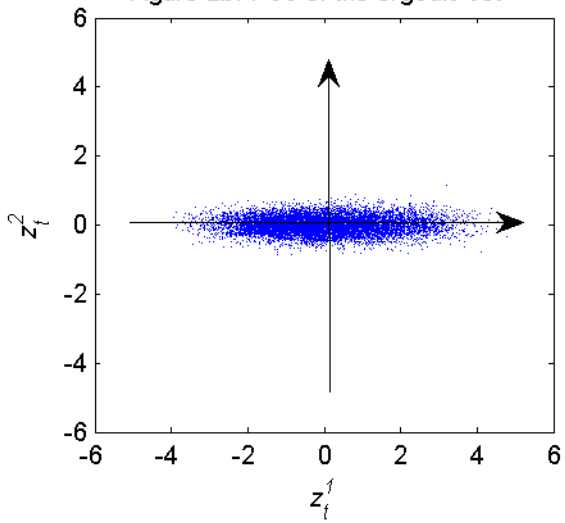

Figure 2c. Normalized PCs of the ergodic set

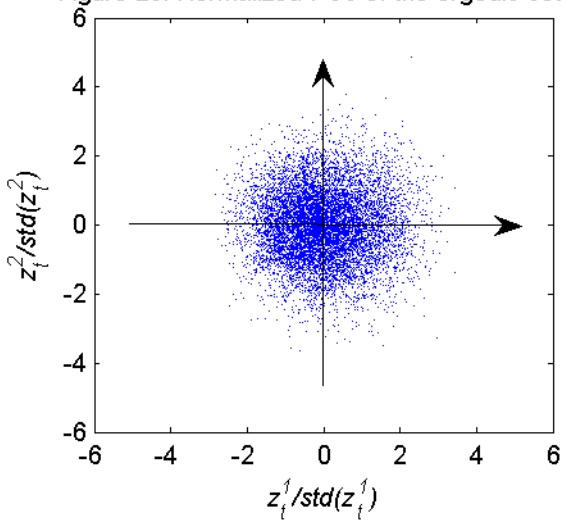

Figure 2d. 4 clusters on the PCs

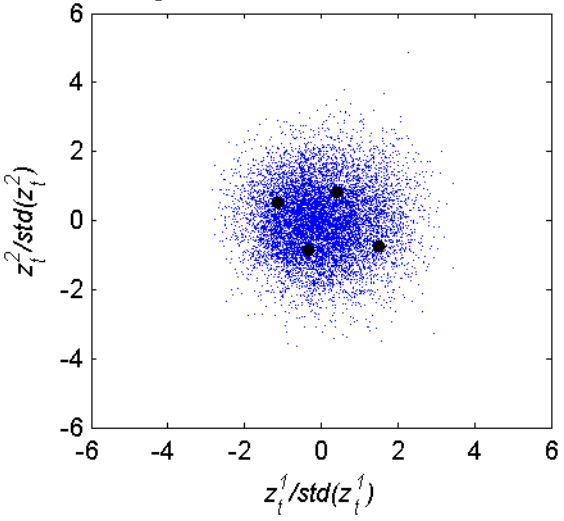

Figure 2e. 10 clusters on the PCs

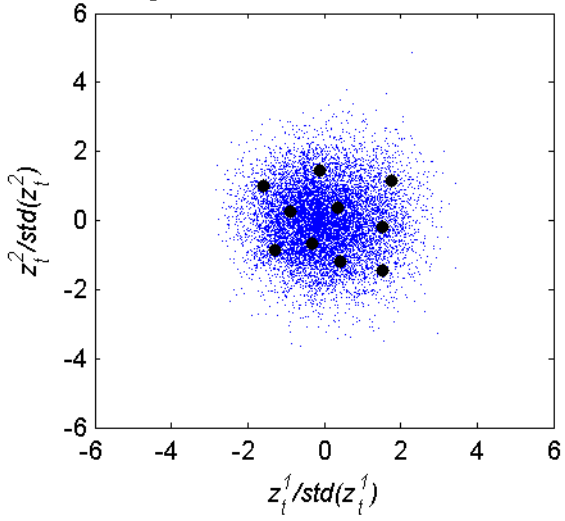

Figure 2f. 100 clusters on the PCs

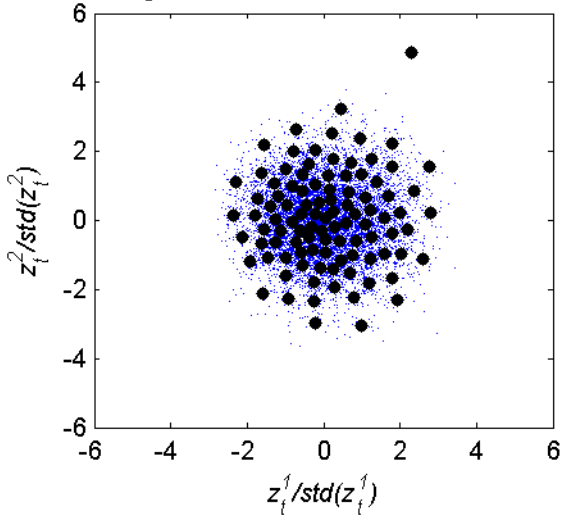

Figure 3. Autocorrection of the cluster grid: initial guess on capital is 10 steady state levels

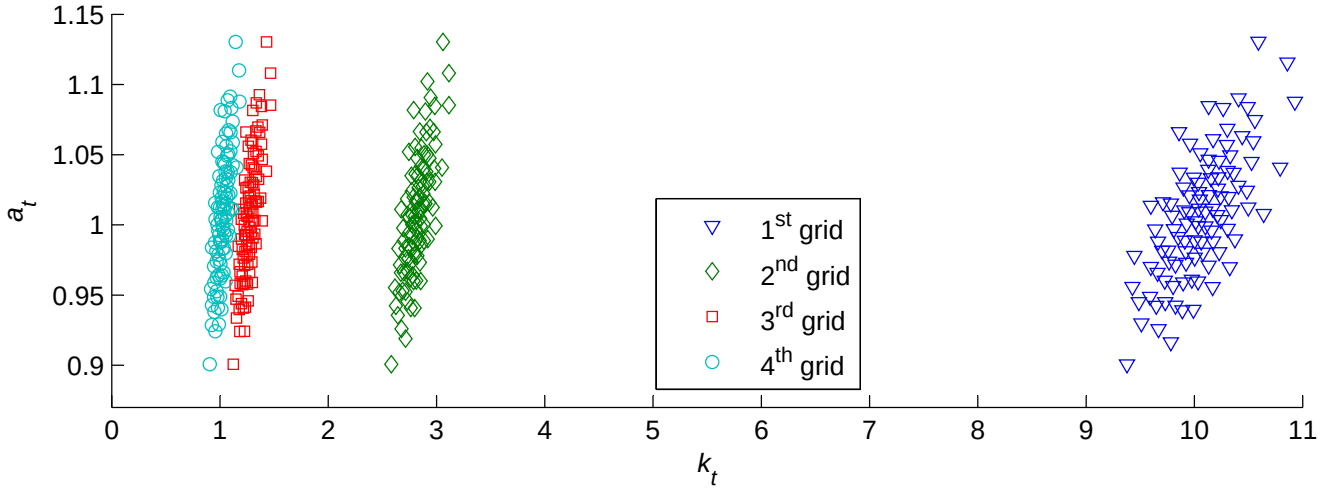

Figure 4a. Smolyak grid

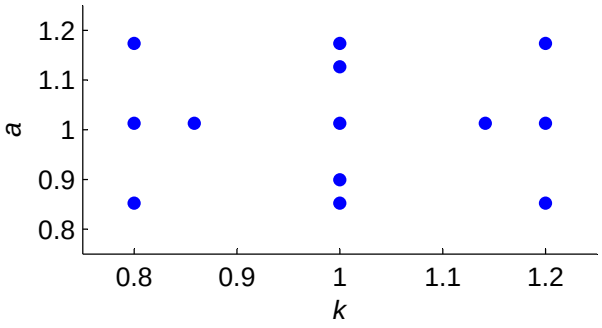

Figure 4b. Cluster grid

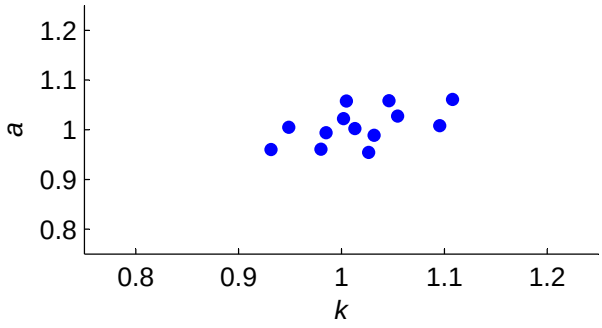

Figure 5a. A time-series solution to the new Keynesian model

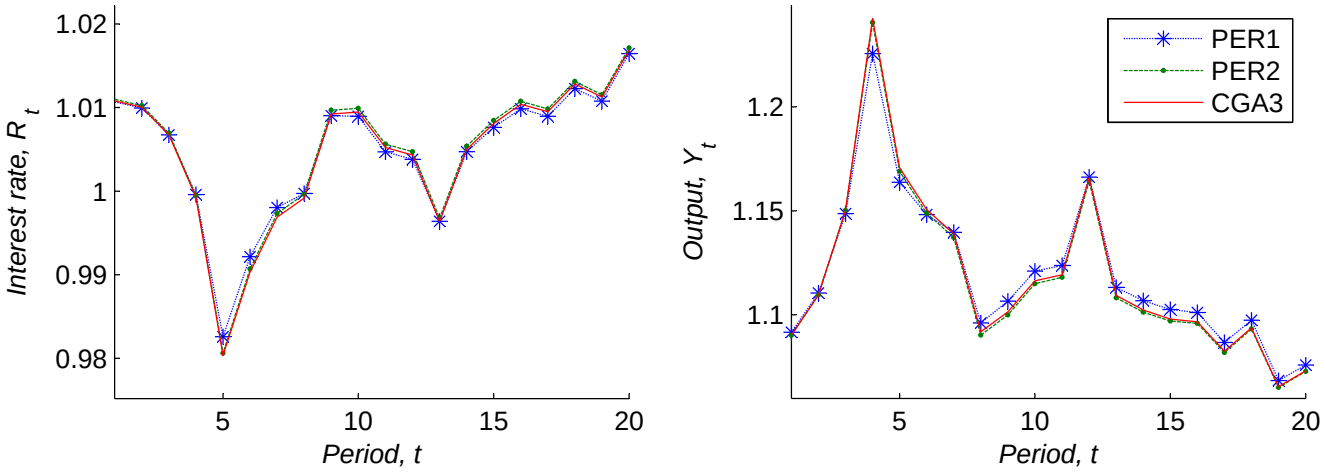

Figure 5b. A time-series solution to the new Keynesian model with ZLB

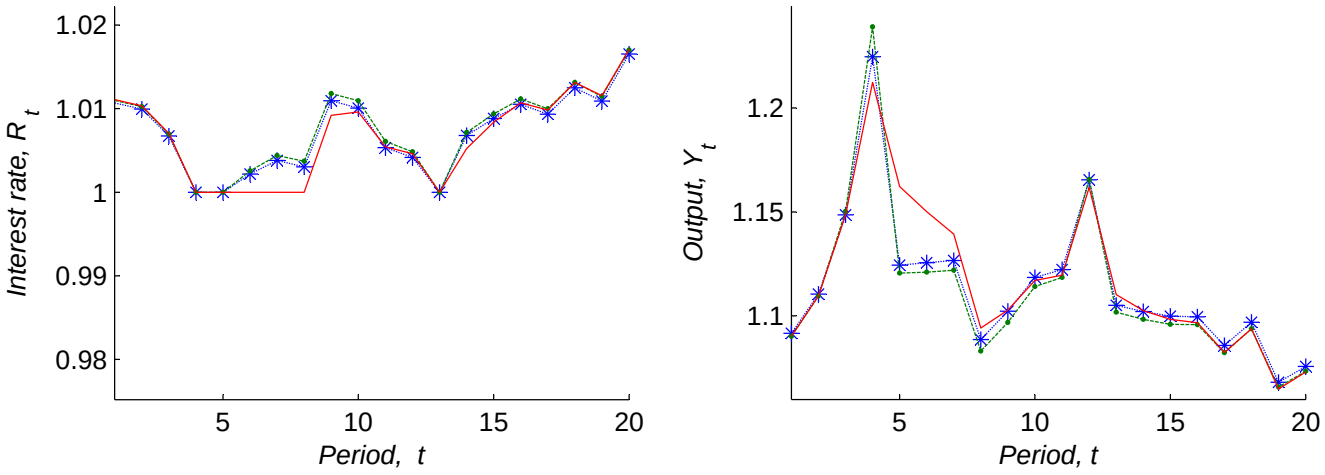

Figure A.1. Agglomerative hierarchical clustering algorithm: an example.