Supplement to "How to solve dynamic stochastic models computing expectations just once": Appendices

(*Quantitative Economics*, Vol. 8, No. 3, November 2017, 851–893)

Kenneth L. Judd
Hoover Institution, Stanford University

Lilia Maliar
Department of Economics, Stanford University

Serguei Maliar
Department of Economics, Santa Clara University

Inna Tsener
Department of Applied Economics, University of the Balearic Islands

Appendix A: Bellman and Euler equation algorithms for solving a growth model with inelastic labor supply

In Algorithms A1–A6, we provide a description of three Bellman and two Euler equation algorithms, which we use to solve the neoclassical stochastic growth model with inelastic labor supply described in Section 2.

Kenneth L. Judd: kennethjudd@mac.com
Lilia Maliar: maliarl@stanford.edu
Serguei Maliar: smaliar@scu.edu
Inna Tsener: inna.tcener@uib.es

DOI: 10.3982/QE329

Algorithm 1a. Value function iteration

Initialization.

a. Choose an approximating function $\widehat{V}(\cdot; b) \approx V$.
b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1, \dots, J$.
c. Construct a grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of V on the grid.

For $m = 1, \dots, M$:
a. Solve for k'_m satisfying
$u'((1-\delta)k_m + z_m f(k_m) - k'_m) = \beta \sum_{j=1}^{J} \omega_j \widehat{V}_1(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})$.
b. Find c_m satisfying
$c_m = (1-\delta)k_m + z_m f(k_m) - k'_m$.
c. Find value function on the grid
$\widehat{v}_m \equiv u(c_m) + \beta \sum_{j=1}^{J} \omega_j \widehat{V}(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})$.

Step 2. Computation of b that fits value function on the grid.

Run a regression to find $\widehat{b}$:
$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{v}_m - \widehat{V}(k_m, z_m; b)\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if
$\frac{1}{M} \sum_{m=1}^{M} |\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}| < 10^{-9}$.
b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi)b^{(i)} + \xi \widehat{b}$.

Algorithm 1b. Value function iteration with precomputation

Initialization.

...
b. Precompute $\{\mathcal{I}_0, \dots, \mathcal{I}_n\}$ using (12).
...

Step 1. Computation of values of V on the grid.

At iteration i, for $m = 1, \dots, M$:
a. Given $b^{(i)}$, find $b'^{(i)}$ from (11) and compute $\widehat{V}_1(k'_m, z_m^{\rho}; b'^{(i)})$.
b. Solve for k'_m satisfying
$u'((1-\delta)k_m + z_m f(k_m) - k'_m) = \beta \widehat{V}_1(k'_m, z_m^{\rho}; b'^{(i)})$.
...
d. Find value function on the grid
$\widehat{v}_m \equiv u(c_m) + \beta \widehat{V}(k'_m, z_m^{\rho}; b'^{(i)})$.

...

Algorithm 2a. Endogenous grid method

Initialization.

a. Choose an approximating function $\widehat{V}(\cdot; b) \approx V$.
b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1, \dots, J$.
c. Construct grid $\Gamma = \{k'_m, z_m\}_{m=1}^{M}$.
d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of V on the grid.

For $m = 1, \dots, M$:
a. Compute $\widehat{W}(k'_m, z_m; b^{(i)}) \equiv \sum_{j=1}^{J} \omega_j \widehat{V}(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})$ and $\widehat{W}_1(k'_m, z_m; b^{(i)}) \equiv \sum_{j=1}^{J} \omega_j \widehat{V}_1(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})$.
b. Find $c_m = u'^{-1}[\beta \widehat{W}_1(k'_m, z_m; b^{(i)})]$.
c. Use a solver to find k_m satisfying
$(1-\delta)k_m + z_m f(k_m) = c_m + k'_m$.
d. Find value function on the grid
$\widehat{v}_m \equiv u(c_m) + \beta \widehat{W}(k'_m, z_m; b^{(i)})$.

Step 2. Computation of b that fits value function on the grid.

Run a regression to find $\widehat{b}$:
$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{v}_m - \widehat{V}(k_m, z_m; b)\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if
$\frac{1}{M} \sum_{m=1}^{M} \left| \frac{(k_m)^{(i)} - (k_m)^{(i-1)}}{(k_m)^{(i-1)}} \right| < 10^{-9}$.
b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi) b^{(i)} + \xi \widehat{b}$.

Algorithm 2b. Endogenous grid method with precomputation

Initialization.

...
b. Precompute $\{\mathcal{I}_0, \dots, \mathcal{I}_n\}$ using (12).
...

Step 1. Computation of values of V on the grid.

At iteration i, for $m = 1, \dots, M$:
a. Given $b^{(i)}$, find $b'^{(i)}$ from (11) and compute $\widehat{W}(k'_m, z_m^{\rho}; b'^{(i)})$ and $\widehat{W}_1(k'_m, z_m^{\rho}; b'^{(i)})$.
b. Find $c_m = u'^{-1}[\beta \widehat{W}_1(k'_m, z_m^{\rho}; b'^{(i)})]$.
...
d. Find value function on the grid
$\widehat{v}_m \equiv u(c_m) + \beta \widehat{W}(k'_m, z_m^{\rho}; b'^{(i)})$.

...

Algorithm 3a. Envelope condition method

Initialization.

a. Choose an approximating function $\widehat{V}(\cdot; b) \approx V$.
b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1, \dots, J$.
c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of V on the grid.

For $m = 1, \dots, M$:
a. Use $b^{(i)}$ to compute $\widehat{V}_1(k_m, z_m; b^{(i)})$.
b. Compute the corresponding values of c_m using
$c_m = u'^{-1}[\widehat{V}_1(k_m, z_m; b^{(i)})(1 - \delta + z_m f'(k_m))^{-1}]$.
c. Find k'_m using
$k'_m = (1 - \delta)k_m + z_m f(k_m) - c_m$.
d. Find value function on the grid
$\widehat{v}_m \equiv u(c_m) + \beta \sum_{j=1}^{J} \omega_j \widehat{V}(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})$.

Step 2. Computation of b that fits the value function on the grid.

Run a regression to find $\widehat{b}$:
$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{v}_m - \widehat{V}(k_m, z_m; b)\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if
$\frac{1}{M} \sum_{m=1}^{M} |\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}| < 10^{-9}$.
b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1 - \xi)b^{(i)} + \xi\widehat{b}$.

Algorithm 3b. Envelope condition method with precomputation

Initialization.

...
b. Precompute $\{\mathcal{I}_0, \dots, \mathcal{I}_n\}$ using (12).
...

Step 1. Computation of values of V on the grid.

At iteration i, for $m = 1, \dots, M$:
a. Given $b^{(i)}$, find $b'^{(i)}$ from (11); use $b^{(i)}$ to compute $\widehat{V}_1(k_m, z_m; b^{(i)})$ and use $b'^{(i)}$ to compute $\widehat{V}(k'_m, z_m^{\rho}; b'^{(i)})$.
...
d. Find value function on the grid
$\widehat{v}_m \equiv u(c_m) + \beta \widehat{V}(k'_m, z_m^{\rho}; b'^{(i)})$.

...

Algorithm 4a. Euler equation algorithm parameterizing Q

Initialization.

a. Choose an approximating function $\widehat{Q}(\cdot; b) \approx Q$.
b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1, \ldots, J$.
c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of Q on the grid.

For $m = 1, \ldots, M$:
a. Use $b^{(i)}$ to compute $\widehat{Q}(k_m, z_m; b^{(i)})$.
b. Find k'_m using
$k'_m = (1-\delta)k_m + z_m f(k_m) - u'^{-1}\left(\frac{\widehat{Q}(k_m, z_m; b^{(i)})}{1-\delta+z_m f(k_m)}\right)$.
c. Find the values of q_m on the grid
$\widehat{q}_m \equiv \beta \sum_{j=1}^{J} \omega_j \widehat{Q}(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})[1-\delta+zf'(k_m)]$.

Step 2. Computation of b that fits the Q function on the grid.

Run a regression to find $\widehat{b}$:
$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{q}_m - \widehat{Q}(k_m, z_m; b)\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if
$\frac{1}{M} \sum_{m=1}^{M} \left|\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}\right| < 10^{-9}$.
b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi)b^{(i)} + \xi\widehat{b}$.

Algorithm 4b. Euler equation algorithm parameterizing Q with precomputation

Initialization.

. . .
b. Precompute $\{\mathcal{I}_0, \ldots, \mathcal{I}_n\}$ using (12).
. . .

Step 1. Computation of values of Q on the grid.

At iteration i, for $m = 1, \ldots, M$:
a. Given $b^{(i)}$, find $b'^{(i)}$ from (11); compute $\widehat{Q}(k_m, z_m; b^{(i)})$ and $\widehat{Q}(k'_m, z_m^{\rho}; b'^{(i)})$.
. . .
c. Find the values of $\widehat{q}_m$ on the grid
$\widehat{q}_m \equiv \beta \widehat{Q}(k'_m, z_m^{\rho}; b'^{(i)})[1-\delta+zf'(k_m)]$.

. . .

Algorithm 5a. Euler equation algorithm parameterizing Q and K

Initialization.

a. Choose approximating functions $\widehat{K}(\cdot; v) \approx K$ and $\widehat{Q}(\cdot; b) \approx Q$.
b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1, \dots, J$.
c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
d. Make an initial guess on $v^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of Q on the grid.

For $m = 1, \dots, M$:
a. Use $v^{(i)}$ to compute $\widehat{K}(k_m, z_m; v^{(i)})$.
b. Find c_m using
$c_m = (1-\delta)k_m + z_m f(k_m) - \widehat{K}(k_m, z_m; v^{(i)})$.
c. Find the values of q_m on the grid
$\widehat{q}_m \equiv u'(c_m)[1 - \delta + z_m f'(k_m)]$.

Step 2. Computation of v that fits the capital function on the grid.

a. Run a regression to find $\widehat{b}$:
$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{q}_m - \widehat{Q}(k_m, z_m; b)\|$,
and compute $\widehat{Q}(k_m, z_m; \widehat{b})$.
b. Compute the values of next-period capital on the grid
$\widehat{k}'_m \equiv \beta \frac{\sum_{j=1}^{J} \omega_j \widehat{Q}(\widehat{K}(k_m, z_m; v^{(i)}), z_m^{\rho} \exp(\varepsilon_j); \widehat{b})}{\widehat{Q}(k_m, z_m; \widehat{b})} (1 - \delta + z f'(k_m)) \widehat{K}(k_m, z_m; v^{(i)})$.
c. Run a regression to find $\widehat{v}$:
$\widehat{v} = \arg\min_v \sum_{m=1}^{M} \|\widehat{k}'_m - \widehat{K}(k_m, z_m; v^{(i)})\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if
$\frac{1}{M} \sum_{m=1}^{M} |\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}| < 10^{-9}$.
b. Use damping with $\xi = 0.15$ to compute $v^{(i+1)} = (1-\xi)v^{(i)} + \xi\widehat{v}$.

Algorithm 5b. Euler equation algorithm parameterizing Q and K with precomputation

Initialization.

...

b. Precompute $\{\mathcal{I}_0, \ldots, \mathcal{I}_n\}$ using (12).

...

Step 2. Computation of v that fits the capital function on the grid.

a. Run a regression to find $\widehat{b}$,

$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{q}_m - \widehat{Q}(k_m, z_m; b)\|$,

and compute $\widehat{Q}(k_m, z_m; \widehat{b})$.

Given $\widehat{b}$, find $\widehat{b}'$ from (11), and compute $\widehat{Q}(\widehat{K}(k_m, z_m; v^{(i)}), z_m^{\rho}; \widehat{b}')$.

b. Compute the value of next-period capital on the grid

$\widehat{k}'_m \equiv \beta \frac{\widehat{Q}(\widehat{K}(k_m, z_m; v^{(i)}), z_m^{\rho}; \widehat{b}')}{\widehat{Q}(k_m, z_m; \widehat{b})} (1 - \delta + z f'(k_m)) \widehat{K}(k_m, z_m; v^{(i)})$.

...

Algorithm 6. Euler equation algorithm parameterizing K (not compatible with precomputation)

Initialization.

a. Choose approximating functions $\widehat{K}(\cdot; v) \approx K$.

...

Step 1. Computation of values of $\widehat{k}'$ on the grid.

For $m = 1, \ldots, M$:

a. Use $v^{(i)}$ to compute $\widehat{K}(k_m, z_m; v^{(i)})$ and

$k''_{m,j} = \widehat{K}(\widehat{K}(k_m, z_m; v^{(i)}), z_m^{\rho} \exp(\epsilon_j); v^{(i)})$, $j = 1, \ldots, J$.

b. Find $c'_{m,j}$ using

$c'_{m,j} = (1 - \delta)\widehat{K}(k_m, z_m; v^{(i)}) + z_m f(\widehat{K}(k_m, z_m; v^{(i)})) - k''_{m,j}$.

c. Find the values of c_m on the grid

$u'(c_m) = \beta \sum_{j=1}^{J} \omega_j u'(c'_{m,j})[1 - \delta + z_m^{\rho} \exp(\epsilon_j) f'(\widehat{K}(k_m, z_m; v^{(i)}))]$.

d. Find the values of k'_m on the grid

$\widehat{k}'_m = (1 - \delta) k_m + z_m f(k_m) - c_m$.

Step 2. Computation of v that fits the capital function on the grid.

a. Run a regression to find $\widehat{v}$:

$\widehat{v} = \arg\min_v \sum_{m=1}^{M} \|\widehat{k}'_m - \widehat{K}(k_m, z_m; v^{(i)})\|$.

...

Appendix B: ECM and Euler equation algorithms for solving a growth model with elastic labor supply

In Algorithms 7 and 8, we provide a description of ECM and Euler equation algorithms, which we use to solve the neoclassical stochastic growth model with valued leisure described in Section 2.

Algorithm 7a. Envelope condition method

Initialization.

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

b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1, \ldots, J$.

c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.

d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of V on the grid.

For $m = 1, \ldots, M$:

a. Use $b^{(i)}$ to compute $\widehat{V}_1(k_m, z_m; b^{(i)})$.

b. Solve for l_m that satisfies

$B(1-l_m)^{-\mu}[1-\delta + z_m f_1(k_m, l_m)] = V_1(k_m, z_m; b^{(i)}) z_m f_2(k_m, l_m)$.

c. Compute the corresponding values of c_m using

$c_m = u_1^{-1}[\frac{B(1-l_m)^{-\mu}}{z_m f_2(k_m, l_m)}]$.

d. Find k'_m using

$k'_m = (1-\delta)k_m + z_m f(k_m, l_m) - c_m$.

e. Find value function on the grid

$\widehat{v}_m \equiv u(c_m, l_m) + \beta \sum_{j=1}^{J} \omega_j \widehat{V}(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})$.

Step 2. Computation of b that fits the value function on the grid.

Run a regression to find $\widehat{b}$:

$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{v}_m - \widehat{V}(k_m, z_m; b)\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if

$\frac{1}{M} \sum_{m=1}^{M} |\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}| < 10^{-9}$.

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

Algorithm 7b. Envelope condition method with precomputation

Initialization.

...

b. Precompute $\{\mathcal{I}_0, \ldots, \mathcal{I}_n\}$ using (12).

...

Step 1. Computation of values of V on the grid.

At iteration i, for $m = 1, \ldots, M$:

a. Given $b^{(i)}$, find $b'^{(i)}$ from (11); use $b^{(i)}$ to compute $\widehat{V}_1(k_m, z_m; b^{(i)})$ and use $b'^{(i)}$ to compute $\widehat{V}(k'_m, z^{\rho}_m; b'^{(i)})$.

...

e. Find value function on the grid $\widehat{v}_m \equiv u(c_m, l_m) + \beta \widehat{V}(k'_m, z^{\rho}_m; b'^{(i)})$.

...

Algorithm 8a. Euler equation algorithm parameterizing Q

Initialization.

a. Choose an approximating function $\widehat{Q}(\cdot; b) \approx Q$.
b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1, \dots, J$.
c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of Q on the grid.

For $m = 1, \dots, M$:
a. Use $b^{(i)}$ to compute $\widehat{Q}(k_m, z_m; b^{(i)})$.
b. Solve for l_m that satisfies
$B(1-l_m)^{-\mu}[1-\delta + z_m f_1(k_m, l_m)] = \widehat{Q}(k_m, z_m; b^{(i)}) z_m f_2(k_m, l_m)$.
c. Compute the corresponding values of c_m using
$c_m = u_1^{-1}[\frac{B(1-l_m)^{-\mu}}{z_m f_2(k_m, l_m)}]$.
d. Find k'_m using
$k'_m = (1-\delta)k_m + z_m f(k_m, l_m) - c_m$.
e. Find the values of q_m on the grid
$\widehat{q}_m \equiv \beta \sum_{j=1}^{J} \omega_j \widehat{Q}(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)})[1-\delta + z f_1(k_m, l_m)]$.

Step 2. Computation of b that fits the Q function on the grid.

Run a regression to find $\widehat{b}$:
$\widehat{b} = \arg\min_b \sum_{m=1}^{M} \|\widehat{q}_m - \widehat{Q}(k_m, z_m; b)\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if
$\frac{1}{M} \sum_{m=1}^{M} |\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}| < 10^{-9}$.
b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi) b^{(i)} + \xi \widehat{b}$.

Algorithm 8b. Euler equation algorithm parameterizing Q with precomputation

Initialization.

...

b. Precompute $\{\mathcal{I}_0, \ldots, \mathcal{I}_n\}$ using (12).

...

Step 1. Computation of values of Q on the grid.

At iteration i, for $m = 1, \ldots, M$:

a. Given $b^{(i)}$, find $b'^{(i)}$ from (11); compute $\widehat{Q}(k_m, z_m; b^{(i)})$ and $\widehat{Q}(k'_m, z^{\rho}_m; b'^{(i)})$.

...

e. Find the values of $\widehat{q}_m$ on the grid

$\widehat{q}_m \equiv \beta \widehat{Q}(k'_m, z^{\rho}_m; b'^{(i)})[1 - \delta + z f_2(k_m, l_m)]$.

...

Appendix C: Euler equation algorithm for solving the multicountry model

In this section, we describe the Euler equation methods that we use to analyze the multicountry model in Section 3.

Algorithm 9a. Euler equation algorithm parameterizing Q^h and K^h

---

Initialization.

---

a. Choose approximating functions $K^h(\cdot; v^h) \approx K^h$, $h = 1, \dots, N$, and $Q^h(\cdot; b^h) \approx Q^h$.
b. Choose integration nodes, $\boldsymbol{\varepsilon}_j = (\varepsilon_j^1, \dots, \varepsilon_j^N)$, and weights, ω_j, $j = 1, \dots, J$.
c. Fix simulation length $T = 2000$ and $(\boldsymbol{k}_0, \boldsymbol{z}_0) = (\mathbf{1}, \mathbf{1})$, where $\mathbf{1} \equiv (1, \dots, 1) \in \mathbb{R}^N$.
d. Draw and fix a sequence of productivity levels $\{\boldsymbol{z}_t\}_{t=1,\dots,T}$ using (40).
e. Construct integration nodes, $\boldsymbol{z}_{t+1,j} = (z_{t+1,j}^1, \dots, z_{t+1,j}^N)$ with $z_{t+1,j}^h = (z_t^h)^\rho \exp(\varepsilon_j^h)$.
f. Make an initial guess on $(v^1)^{(1)}, \dots, (v^h)^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, given $(v^1)^{(i)}, \dots, (v^h)^{(i)}$:
Step 1. Computation of values of Q on the simulated points.
For $t = 1, \dots, T$:
a. Use $k_{t+1}^h = \widehat{K}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; (v^h)^{(i)})$, $h = 1, \dots, N$, to recursively calculate $\{\boldsymbol{k}_{t+1}\}_{t=0,\dots,T}$.
b. Compute $\{\boldsymbol{c}_t\}_{t=0,\dots,T}$ satisfying $c_t^h = (\sum_{h=1}^N c_t^h)/N$.
c. Compute q_t^h from
$\widehat{q}_t^h \equiv u'(c_t^h)[1 - \delta + z_t^h f'(k_t^h)]$.

Step 2. Computation of v^h that fits the values of capital on the simulated points.
a. Run a regression to find $\widehat{b}^h$,
$\widehat{b}^h = \arg\min_{b^h} \sum_{m=1}^M \| \widehat{q}_t^h - \widehat{Q}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; b^h) \|$,
and compute $\widehat{Q}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; \widehat{b}^h)$.
b. Compute the values of next-period capital on the simulated points
$\widehat{k}_{t+1}^h \equiv \beta \frac{\sum_{j=1}^J \omega_j \widehat{Q}^h(\boldsymbol{k}_{t+1}, \boldsymbol{z}_{t+1,j}; \widehat{b}^h)}{\widehat{Q}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; \widehat{b}^h)} [1 - \delta + z_t^h f'(k_t^h)] K^h(\boldsymbol{k}_t, \boldsymbol{z}_t; v^h)$, $h = 1, \dots, N$.
c. Run a regression to find $\widehat{v}^h$:
$\widehat{v}^h \equiv \arg\min_{v^h} \sum_{t=1}^T \| \widehat{k}_{t+1}^h - \widehat{K}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; (v^h)^{(i)}) \|$.

Step 3. Convergence check and fixed-point iteration.
a. Check for convergence for $i \geq 2$; end Step 2 if
$\frac{1}{TN} \sum_{t=1}^T \sum_{h=1}^N \left| \frac{(k_{t+1}^h)^{(i)} - (k_{t+1}^h)^{(i-1)}}{(k_{t+1}^h)^{(i-1)}} \right| < 10^{-10}$.
b. Use damping with $\xi = 0.1$ to compute $(v^h)^{(i+1)} = (1 - \xi)(v^h)^{(i)} + \xi \widehat{v}^h$.

Algorithm 9b. Euler equation algorithm parameterizing Q^h and K^h with precomputation

Initialization.

...

b. Precompute $\{\mathcal{I}_0, \ldots, \mathcal{I}_n\}$ using (53).

...

Step 2. Computation of v^h that fits the values of capital on the simulated points.

a. Run a regression to find $\widehat{b}^h$,

$\widehat{b}^h = \arg\min_b \sum_{m=1}^M \|\widehat{q}_t^h - \widehat{Q}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; b^h)\|$,

and compute $\widehat{Q}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; \widehat{b}^h)$.

Given $\widehat{b}$, find $\widehat{b}'$ from (47), and compute $\widehat{Q}^h(\boldsymbol{k}_{t+1}, \boldsymbol{z}_t^\rho; (\widehat{b}^h)')$.

b. Compute the values of next-period capital on the simulated points

$\widehat{k}_{t+1}^h \equiv \beta \frac{\widehat{Q}^h(\boldsymbol{k}_{t+1}, \boldsymbol{z}_t^\rho; (\widehat{b}^h)')}{\widehat{Q}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; \widehat{b}^h)} [1 - \delta + z_t^h f'(k_t^h)] K^h(\boldsymbol{k}_t, \boldsymbol{z}_t; v^h)$, $h = 1, \ldots, N$.

...

Algorithm 10. Euler equation algorithm parameterizing K^h (not compatible with precomputation)

Initialization.

a. Choose approximating functions $K^h(\cdot; v^h) \approx K^h$, $h = 1, \ldots, N$.

...

Step 1. Computation of values of $\widehat{k}_{t+1}^h$ on the simulated points.

For $t = 1, \ldots, T$:

a. Use $k_{t+1}^h = \widehat{K}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; (v^h)^{(i)})$, $h = 1, \ldots, N$, to recursively calculate $\{\boldsymbol{k}_{t+1}\}_{t=0,\ldots,T}$.

b. Compute $\{\boldsymbol{c}_t\}_{t=0,\ldots,T}$ satisfying $c_t^h = (\sum_{h=1}^N c_t^h)/N$.

c. Compute $k_{t+2,j}^h = \widehat{K}^h(\boldsymbol{k}_{t+1}, z_t^\rho \exp(\boldsymbol{\epsilon}_j); (v^h)^{(i)})$.

d. Find $\{\boldsymbol{c}_{t+1,j}\}_{t=0,\ldots,T}$ satisfying $c_{t+1,j}^h = (\sum_{h=1}^N c_{t+1,j}^h)/N$.

Step 2. Computation of v^h that fits the values of capital on the simulated points.

a. Compute the values of next-period capital on the simulated points for $h = 1, \ldots, N$:

$\widehat{k}_{t+1}^h \equiv \beta \sum_{j=1}^J \omega_j u'(c_{t+1,j}^h)(u'(c_t^h))^{-1}[1 - \delta + (z_t^h)^\rho \exp(\epsilon_j^h) f'(k_{t+1}^h)] K^h(\boldsymbol{k}_t, \boldsymbol{z}_t; v^h)$.

b. Run a regression to find $\widehat{v}^h$:

$\widehat{v}^h \equiv \arg\min_{v^h} \sum_{t=1}^T \|\widehat{k}_{t+1}^h - \widehat{K}^h(\boldsymbol{k}_t, \boldsymbol{z}_t; (v^h)^{(i)})\|$.

...

TABLE C.1. Accuracy and cost of the Euler equation algorithm for the multicountry model: GSSA parameterizing K.

Polynomial	Precomputation			M1			M2			GH(2)		
Degree	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU	L_1	L_∞	CPU
1st	−2.77	−1.81	61	−2.77	−1.80	92	−2.77	−1.80	105	−2.77	−1.80	86
2nd	−3.88	−2.61	223	−3.88	−2.61	418	−3.88	−2.61	621	−3.88	−2.61	389
3rd	−4.94	−3.55	382	−4.95	−3.55	614	−4.95	−3.55	818	−4.95	−3.55	567
4th	−6.05	−4.68	574	−6.04	−4.67	840	−6.04	−4.67	1499	−6.04	−4.67	1241
5th	−7.15	−5.79	738	−7.15	−5.78	938	−7.15	−5.78	1875	−7.15	−5.78	1551
$N = 5$												
1st	−2.86	−1.94	75	−2.86	−1.93	184	−2.86	−1.93	687	−2.86	−1.93	396
2nd	−3.99	−2.81	319	−3.99	−2.82	958	−3.99	−2.82	3955	−3.99	−2.82	2456
3rd	−5.10	−3.84	2550	−5.13	−3.83	8893	−5.13	−3.83	7349	−5.13	−3.83	12,221
4th	−6.18	−4.89	7033	−6.31	−4.92	16,640	–	–	–	–	–	–
$N = 10$												
1st	−2.87	−1.89	125	−2.87	−1.89	515	−2.87	−1.89	4889	−2.87	−1.89	18,263
2nd	−4.00	−2.80	666	−4.01	−2.78	2681	−4.01	−2.78	23,610	–	–	–
3rd	−4.99	−3.88	14,837	−5.18	−3.92	14,232	–	–	–	–	–	–
$N = 20$												
1st	−3.12	−2.09	152	−3.12	−2.08	1803	−3.12	−2.08	32434	–	–	–
2nd	−4.36	−3.26	3303	−4.40	−3.32	13,204	–	–	–	–	–	–
$N = 30$												
1st	−3.15	−2.08	221	−3.16	−2.08	4688	–	–	–	–	–	–
2nd	−4.22	−3.22	13,543	–	–	–	–	–	–	–	–	–

Note: The main result columns correspond to variants of GSSA that evaluate integrals by using the precomputation method, the monomial integration methods with $2N$ and $2N^2+1$ nodes, and the Gauss Hermite quadrature method with two nodes, respectively; the statistics L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality condition and test points (in log 10 units) on a stochastic simulation of 10,000 observations; CPU is the time necessary for computing a solution (in seconds).

Appendix D: Euler equation algorithm for solving the Aiyagari model

In this section, we describe a solution algorithm for Aiyagari's (1994) model analyzed in Section 4. We focus on the algorithm for solving the individual problem in which the precomputation results appear. The rest of the algorithm is standard and relies on stochastic simulation and a bisection technique as in Aiyagari (1994).

Algorithm 11a. Euler equation algorithm parameterizing Q and K

Initialization.

a. Given K, compute $R = \alpha K^{\alpha-1} - \delta$ and $W = (1-\alpha)(K)^{\alpha}$.
b. Choose an approximating function $\widehat{K}(\cdot; b^{\ell}) \approx K$ and $\widehat{Q}(\cdot; v^{\ell}) \approx Q$ for $\ell \in \{1, \dots, J\}$.
c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
d. Make an initial guess on $K(\cdot; b^{\ell})^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of K on the grid.

For $m = 1, \dots, M$:
a. Find $c_{m,\ell}$ using
$c_{m,\ell} = W z_{\ell} + (1+R)k_m - \widehat{K}(\cdot; b^{\ell})$.
b. Compute $q_{m,\ell} = c_{m,\ell}^{-\gamma}(1+R)$.
c. Run a regression to find $\widehat{v}^{\ell}$ for $\ell \in \{1, \dots, J\}$:
$\widehat{v}^{\ell} = \arg\min_{v^{l}} \sum_{m=1}^{M} \|\widehat{q}_{m,\ell} - \widehat{Q}(k_m; v)\|$.
d. Find $\widehat{k}'_{m,\ell}$ using
$\widehat{k}'_{m,\ell} = W z_{\ell} + (1+R)k_m - u'^{-1}[\beta \sum_j \pi_{jl} \widehat{Q}(k'_{m,\ell}; \widehat{v}^{j})]$.

Step 2. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$; end Step 1 if
$\max_{m,l} |\frac{(c_{m,\ell})^{(i-1)} - (c_{m,\ell})^{(i)}}{(c_{m,\ell})^{(i-1)}}| < 10^{-10}$.
b. Use damping with $\xi = 0.5$ to compute
$(\widehat{K}(\cdot; b^{\ell}))^{(i+1)} = (1-\xi)(\widehat{K}(\cdot; b^{\ell}))^{(i)} + \xi(\widehat{k}'_{m,\ell})^{(i)}$.

Algorithm 11b. Euler equation algorithm parameterizing Q with precomputation

Initialization.

...

Iterative cycle. Computation of a solution.

Step 1. Computation of values of K on the grid.

At iteration i, for $m = 1, \dots, M$:

...

d. Given $\widehat{v}^l$, $l = 1, \dots, J$, find $(\widehat{v}^\ell)'$ from (67).

e. Find $\widehat{k}'_{m,\ell}$ using

$\widehat{k}'_{m,\ell} = W z_\ell + (1 + R) k_m - u'^{-1}[\beta \widehat{Q}(k'_{m,\ell}; (\widehat{v}^\ell)')]$.

...

Algorithm 12. Euler equation algorithm for Aiyagari's (1994) model parameterizing K (not compatible with precomputation)

Initialization.

...

b. Choose an approximating function $\widehat{K}(\cdot; b^\ell) \approx K$ for $\ell \in \{1, \dots, J\}$.

...

Iterative cycle. Computation of a solution.

Step 1. Computation of values of K on the grid.

At iteration i, for $m = 1, \dots, M$:

...

For $\ell = \{1, \dots, J\}$:

b. Compute $k''_{m,j} = \widehat{K}(k'_{m,\ell}; b^j)$ for $j = \{1, \dots, J\}$.

c. Find $c'_{m,j}$ for $j = \{1, \dots, J\}$ using

$c'_{m,j} = W z_j + (1 + R) k'_{m,\ell} - k''_{m,j}$.

d. Find $\widehat{k}'_{m,\ell}$ using

$\widehat{k}'_{m,\ell} = W z_\ell + (1 + R) k_m - u'^{-1}[\beta \sum_j \pi_{jl} (c'_{m,j})^{-\gamma} [1 + R]]$.

...

Co-editor José-Víctor Ríos-Rull handled this manuscript.

Manuscript received 9 November, 2012; final version accepted 6 December, 2016; available online 2 June, 2017.