Solving Hetero-Agent Model with Aggregate Uncertainty Using Projection + Perturbation

Thomas Winberry

September 7, 2021

Canonical Model

- Households optimization:

$$c(\varepsilon,k;Z,\mu)^{-\sigma} = \beta E[(1+r')c(\varepsilon',k';Z',\mu')^{-\sigma}] \quad (1)$$

budget constraint:

$$c(\varepsilon,k;Z,\mu) = \underbrace{(1-\tau)w}_{\text{effective wage}}\varepsilon + \underbrace{bw}_{transfer}(1-\varepsilon) + (1+r')k - k' \quad (2)$$

borrowing constraint

$$k \geq \underline{k} = 0 \quad (3)$$

- Firms:

$$w = (1-\alpha)Z_t(\frac{K_t}{L_t})^{\alpha} \quad (4)$$

$$r = \alpha Z_t(\frac{K_t}{L_t})^{\alpha-1} - \delta \quad (5)$$

- Government:

$$\tau_t = \frac{bU_t}{L_t} = \frac{b(1-L_t)}{L_t}$$

Canonical Model

- Aggregate states:

$$log(Z_{t+1}) = \rho_z log(Z_t) + \sigma_z \omega_{t+1}, \omega \sim N(0,1) \quad (6)$$

- Idiosyncratic states: employed / unemployed:
  - $\varepsilon_t = 1$, if employed;
  - $\varepsilon_t = 0$, if unemployed;
- Transition probabilities (constant over time[1])

ε/ε'	u	e
u	$\pi(u\|u)$	$\pi(e\|u)$
e	$\pi(u\|e)$	$\pi(e\|e)$

- Evolution of distribution: for all measurable sets Δ_k

$$\mu'(\varepsilon', \Delta_k | Z, \mu) = \sum_{\tilde{\varepsilon}} \pi(\varepsilon'|\varepsilon) \int 1\{k'(\varepsilon, k; Z, \mu) \in \Delta_k\} \mu(\varepsilon, dk) \quad (7)$$

system

[1] This implies L_t is constant over time, so is tax rate τ_t.

Computational Challenges

$$c(\varepsilon, k; Z, \mu)^{-\sigma} = \beta E[(1 + r'(Z', \mu'(Z, \mu)))c(\varepsilon', k'; Z', \mu')^{-\sigma}]$$

"Infinite-dimensional Fixed Point Problem":

- aggregate distribution (μ') $\rightarrow$ r' $\rightarrow$ individual decision (c)
- individual decision (c) $\rightarrow$ k' $\rightarrow$ aggregate distribution (μ')

Literature: Algorithm Solving Hetero-Agent Model

- Projection: parameterization of the cross-sectional distribution [2]
    - Algan, Allais, and Den Haan, 2008, 2010 ; Reiter, 2010
    - pros: globally accurate
    - cons: slow computation
- Perturbation
    - Kim, Kollmann, and Kim, 2010; Preston and Roca, 2007 (pure)
    - pros: fast, really fast
    - cons: not suited for model with OBCs etc.
- Hybrid:
    - Projection and Simulation (i.e., Krusell-Smith Algorithm)
        - Maliar, Maliar, and Valli, 2010 (stochastic simulation)
        - Young, 2010 (non-stochastic simulation)
    - Projection and Perturbation
        - Reiter, 2009 (projection: "fine" histogram)
        - Winberry, 2018 (projection: parameterization coefficient)

Handle the Challenges

Individual problem:

- idiosyncratic shocks (uncertainty): large (0 or 1);
- nonlinear and high dimensional individual problem
- perturbation (local solution) probably a bad idea
- projection is needed (as in KS algorithm etc.)

Aggregate problem:

- aggregate shocks (uncertainty): relatively small (1 s.d.);
- linear or almost linear aggregate problem
- perturbation (local solution) probably works

Solution: Projection+Perturbation.

Krusell and Smith (1998)

Projection + Perturbation Algorithm

- Step 1: Approximate the model's equilibrium objects - the distribution, law of motion, factor prices, decision rules (often infinite-dimensional)- using *finite dimensional global approximations* w.r.t. individual state variables.
- Step 2: Compute the stationary equilibrium of the approximated model without aggregate shocks but still with idiosyncratic shocks.
- Step 3: Compute the aggregate dynamic of the approximated model by perturbing it around the stationary equilibrium.

Step 1: Approximation

a. Approximate[3] distribution of household over asset holding (k) with

$$g_{\varepsilon,t}(k) \simeq g^0_{\varepsilon,t} exp\{g^1_{\varepsilon,t}(k - m^1_{\varepsilon,t}) + \sum_{i=2}^{n_g} g^i_{\varepsilon,t}[(k - m^1_{\varepsilon,t})^i - m^i_{\varepsilon,t}]\} \quad (8)$$

where

- n_g: the order of approximation
- $g^j_{\varepsilon,t}$: parameters (solved from system of moment-equations below)
- $m^j_{\varepsilon,t}$: centralized moments of distribution:

$$m^1_{\varepsilon,t} = \int k g_{\varepsilon,t}(k) dk \quad (9)$$

$$m^i_{\varepsilon,t} = \int (k - m^1_{\varepsilon,t})^i g_{\varepsilon,t}(k) dk, \quad \text{for i=2,3,...,} n_g. \quad (10)$$

(issue: occasionally binding constraint) ▸ system

Step 1: Approximation

a. (cont-) Approximate distribution of household over capital holding (k) *at the borrowing constraint* (k=$\underline{k}$).

$\rightarrow$ *positive* mass at $\underline{k}$

$\rightarrow$ denote the mass at constraint with productivity ε as $\hat{m}_{\varepsilon}$.

$\Rightarrow$ Law of motion of the mass :

$$\hat{m}_{\varepsilon,t+1} = \frac{1}{\pi(\varepsilon)}[\sum_{\tilde{\varepsilon}} \hat{m}_{\tilde{\varepsilon},t}\pi(\tilde{\varepsilon})\pi(\varepsilon|\tilde{\varepsilon})1\{k'(\tilde{\varepsilon},\underline{k}) = \underline{k}\}+$$

$$\sum_{\tilde{\varepsilon}}(1-\hat{m}_{\tilde{\varepsilon},t})\pi(\tilde{\varepsilon})\pi(\varepsilon|\tilde{\varepsilon})\int 1\{k'(\tilde{\varepsilon},k) = \underline{k}\}g_{\tilde{\varepsilon},t}(k)dk] \quad (11)$$

where $\pi(\varepsilon)$ is the mass of households with prod. ε;

Step 1: Approximation

b. Approximate LOM of distribution by LOM of moments.[4]

$$m^1_{\varepsilon,t+1} = \frac{1}{\pi(\varepsilon)}[\sum_{\tilde{\varepsilon}} \hat{m}_{\tilde{\varepsilon},t}\pi(\tilde{\varepsilon})\pi(\varepsilon|\tilde{\varepsilon})k'(\tilde{\varepsilon},\underline{k})+$$

$$\sum_{\tilde{\varepsilon}}(1-\hat{m}_{\tilde{\varepsilon},t})\pi(\tilde{\varepsilon})\pi(\varepsilon|\tilde{\varepsilon})\int k'(\tilde{\varepsilon},k)g_{\tilde{\varepsilon},t}(k)dk] \quad (12)$$

$$m^j_{\varepsilon,t+1} = \frac{1}{\pi(\varepsilon)}[\sum_{\tilde{\varepsilon}} \hat{m}_{\tilde{\varepsilon},t}\pi(\tilde{\varepsilon})\pi(\varepsilon|\tilde{\varepsilon})[k'(\tilde{\varepsilon},\underline{k})-m^1_{\varepsilon,t+1}]^j+$$

$$\sum_{\tilde{\varepsilon}}(1-\hat{m}_{\tilde{\varepsilon},t})\pi(\tilde{\varepsilon})\pi(\varepsilon|\tilde{\varepsilon})\int [k'(\tilde{\varepsilon},k)-m^1_{\varepsilon,t+1}]^j g_{\tilde{\varepsilon},t}(k)dk] \quad (13)$$

▸ system

Step 1: Approximation

c. Compute aggregate capital stock from *approximated* distribution:

$$K_t = \sum_{\varepsilon} \pi(\varepsilon) \sum_{j=1}^{m_g} \omega_j k_j g_{\varepsilon,t}(k_j) \tag{14}$$

And approximate factor prices:

$$w = (1-\alpha) Z_t (\frac{K_t}{L_t})^{\alpha} \tag{15}$$

$$r = \alpha Z_t (\frac{K_t}{L_t})^{\alpha-1} - \delta \tag{16}$$

▸ system

Step 1: Approximation

d. Deal with expectation term:

- define conditional expectation term as:

$$\psi(\varepsilon,k) = E[\beta(1+r')c_{t+1}(\varepsilon',k')^{-\sigma}]$$

- saving policy rule:

$$k'(\varepsilon,k) = \max\{\underline{k}, w[(1-\tau)\varepsilon + b(1-\varepsilon)] + (1+r)k - \psi(\varepsilon,k)^{\frac{1}{\sigma}}\} \quad (17)$$

- consumption policy rule:

$$c(\varepsilon,k) = w[(1-\tau)\varepsilon + b(1-\varepsilon)] + (1+r)k - k'(\varepsilon,k) \quad (18)$$

▸ system

Step 1: Approximation

d. (cont-) Deal with expectation term:

- approximate conditional expectation with Chebyshev polynomials:

$$\hat{\psi}(\varepsilon,k) \simeq exp\{\sum_{i=1}^{n_\psi} \theta_{\varepsilon i,t} T_i(\xi(k))\}$$

where T_i is i-th order Chebyshev polynomial of transformed nodes $\xi(k)$ on [-1, 1].

- approximate saving policy rule:

$$\hat{k}'(\varepsilon,k) = \max\{\underline{k}, w[(1-\tau)\varepsilon + b(1-\varepsilon)] + (1+r)k - \hat{\psi}(\varepsilon,k)^{\frac{1}{\sigma}}\} \quad (19)$$

- approximate consumption policy rule:

$$\hat{c}_t(\varepsilon,k) = w[(1-\tau)\varepsilon + b(1-\varepsilon)] + (1+r)k - \hat{k}'(\varepsilon,k) \quad (20)$$

system

Step 1: Approximation

Approximate model is characterized by:

- LoM of aggregate state (Z_t): (6)
- approximate moments m: (9) and (10)
- LoM of μ: (7) $\rightarrow$ LoM of m: (12) and (13)
- factor prices: (4) and (5) $\rightarrow$ approx' factor price: (15) and (16)
- policy rule: (1), (2) $\rightarrow$ approx' policy rule: (19) and (20)

Define a residual function based on the approximate model:

$$E_t[f(y_t, y_{t+1}, x_t, x_{t+1}; \chi)] = 0 \tag{21}$$

Step 2: Stationary Equilibrium

The stationary equilibrium is a system of (nonlinear) equations:

$$f(y^*, y^*, x^*, x^*; 0) = 0 \tag{22}$$

(This can be very difficult to solve due to the large size!)
Author's strategy: write s.s. in terms of K.

- compute factor prices: **r** and **w**. (by assumption L is constant)
- solve the approximated expectation term
- solve the decision rules
- solve for invariant distribution of moments **m** and implied parameters **g**.
- update aggregate capital K' from (14)

$$K' = \sum_{\varepsilon} \pi(\varepsilon) \sum_{j=1}^{m_g} \omega_j k'(\varepsilon, k_j) g_{\varepsilon}(k)$$

- return K'-K and solve for a zero of this equation

Step 3: Perturbation on Aggregate Dynamics

- Import the model to **Dynare**[5], and **Dynare** will:
- differentiate these equations;
- evaluate them at steady state;
- solve the resulting system at first order;
- perform default analysis.

[5]This step is not trivial. Dynare does not accept matrix expressions used heavily in the matlab codes to solve steady state. We need to re-write teh matrix expressions as loops over scalar variables using Dynare's macro-processor.
For details on macro-processor, see: http://www.dynare.org/manual/index_37.html; or at: http://www.dynare.org/DynareShanghai2013/macroprocessor.pdf

Khan and Thomas (2008)

Model

Production function:

$$y_{jt} = e^{z_t} e^{\varepsilon_{jt}} k_{jt}^{\theta} n_{jt}^{\nu}, \quad \theta + \nu < 1 \tag{23}$$

aggregate shock:

$$z_{t+1} = \rho_z z_t + \sigma_z \omega_{t+1}^z, \quad \text{where } \omega_{t+1}^z \sim N(0,1) \tag{24}$$

idiosyncratic shock:

$$\varepsilon_{jt+1} = \rho_\varepsilon \varepsilon_{jt} + \sigma_\varepsilon \omega_{jt+1}^\varepsilon, \quad \text{where } \omega_{jt+1}^\varepsilon \sim N(0,1) \tag{25}$$

capital law of motion:

$$k_{jt+1} = (1-\delta)k_{jt} + i_{jt}$$

If $\frac{i_{jt}}{k_{jt}} \notin [-a, a]$, firm must pay a fixed adjustment cost ξ_{jt} in units of labor.

Model

Bellman equation:

$$
\begin{aligned}
v(\varepsilon, k, \xi; \mathbf{s}) =& \lambda(\mathbf{s}) \max_n \left\{ e^z e^{\varepsilon} k^{\theta} n^{\nu} - w(\mathbf{s}) n \right\} \\
& + \max \{ v^a(\varepsilon, k; \mathbf{s}) - \xi \lambda(\mathbf{s}) w(\mathbf{s}), v^n(\varepsilon, k; \mathbf{s}) \}
\end{aligned} \tag{26}
$$

where for adjusting firm:

$$
v^a(\varepsilon, k; \mathbf{s}) = \max_{k' \in \mathbb{R}} -\lambda(\mathbf{s}) \left(k' - (1-\delta) k \right) + \beta \mathbb{E} \left[\widehat{v} \left(\varepsilon', k'; \mathbf{s}' \left(z'; \mathbf{s} \right) \mid \varepsilon, k; \mathbf{s} \right] \right.,
$$

and for non-adjusting firm (i.e. $k' \in [(1-\delta-a)k, (1-\delta+a)k]$)

$$
v^n(\varepsilon, k; \mathbf{s}) = \max -\lambda(\mathbf{s}) \left(k' - (1-\delta) k \right) + \beta \mathbb{E} \left[\widehat{v} \left(\varepsilon', k'; \mathbf{s}' \left(z'; \mathbf{s} \right) \mid \varepsilon, k; \mathbf{s} \right] \right.
$$

Unique threshold value of the fixed cost ξ below which firm adjusts

$$
\widehat{\xi}(\varepsilon, k; \mathbf{s}) = \frac{v^a(\varepsilon, k; \mathbf{s}) - v^n(\varepsilon, k; \mathbf{s})}{\lambda(\mathbf{s}) w(\mathbf{s})}
$$

Model

Ex ante value function:

$$\widehat{v}(\varepsilon,k;\mathbf{s}) = \lambda(\mathbf{s})\max_{n}\left\{e^{z}e^{\varepsilon}k^{\theta}n^{\nu} - w(\mathbf{s})n\right\} + \left(1-\frac{\widehat{\xi}(\varepsilon,k;\mathbf{s})}{\bar{\xi}}\right)v^{n}(\varepsilon,k;\mathbf{s}) + \frac{\widehat{\xi}(\varepsilon,k;\mathbf{s})}{\bar{\xi}}\left(v^{a}(\varepsilon,k;\mathbf{s}) - \lambda(\mathbf{s})w(\mathbf{s})\frac{\widehat{\xi}(\varepsilon,k;\mathbf{s})}{2}\right) \tag{27}$$

Law of motion of distribution:

$$g'\left(\varepsilon',k';\mathbf{s}\right) = \iiint \left[1\left\{\rho_{\varepsilon}\varepsilon + \sigma_{\varepsilon}\omega_{\varepsilon}' = \varepsilon'\right\} \times \left[\frac{\widehat{\xi}(\varepsilon,k;\mathbf{s})}{\bar{\xi}}1\left\{k^{a}(\varepsilon,k;\mathbf{s}) = k'\right\} + \left(1-\frac{\widehat{\xi}(\varepsilon,k;z,\mathbf{m})}{\bar{\xi}}\right)1\left\{k^{n}(\varepsilon,k;\mathbf{s}) = k'\right\}\right]\right] \times p\left(\omega_{\varepsilon}'\right)g(\varepsilon,k)d\omega_{\varepsilon}'d\varepsilon dk \tag{28}$$

Projection + Perturbation Algorithm

- Step 1: Approximate the model's equilibrium objects - the distribution, law of motion, firm value, Bellman equation (often infinite-dimensional)- using *finite dimensional global approximations* w.r.t. individual state variables.
- Step 2: Compute the stationary equilibrium of the approximated model without aggregate shocks but still with idiosyncratic shocks.
- Step 3: Compute the aggregate dynamic of the approximated model by perturbing it around the stationary equilibrium.

Step 1a: Approximate density function

a. Approximate[6] density function over individual states (ε, k) with

$$g(\varepsilon,k) \approx \hat{g}(\varepsilon,k) = g_0 \exp\left\{ g_1^1\left(\varepsilon - m_1^1\right) + g_1^2\left(\log k - m_1^2\right) + \sum_{i=2}^{n_g}\sum_{j=0}^{i} g_i^j\left[\left(\varepsilon - m_1^1\right)^{i-j}\left(\log k - m_1^2\right)^j - m_i^j\right]\right\} \quad (29)$$

- n_g: the order of approximation (setting $n_g = 2$ sufficient)
- $\vec{g} : g_1^1, g_1^2, g_i^j$: coefficients (to be solved) [7]
- $\vec{m} : m_1^1, m_1^2, m_i^j$: centralized moments of distribution (parameter)

$$m_1^1 = E(\varepsilon) = \iint \varepsilon \hat{g}(\varepsilon,k) d\varepsilon dk$$

$$m_1^2 = E(\log k) = \iint \log k \hat{g}(\varepsilon,k) d\varepsilon dk \text{ and} \quad (30)$$

$$m_i^j = \iint \left(\varepsilon - m_1^1\right)^{i-j}\left(\log k - m_1^2\right)^j \hat{g}(\varepsilon,k) d\varepsilon dk$$

[6] In practice we approximate the integrals using Gauss-Legendre quadrature.

[7] g_0 is set to ensure total mass of p.d.f. is 1.

Step 1a: Approximate density function

- $g(\varepsilon, k) \rightarrow \hat{g}(\varepsilon, k) \sim \vec{m} + \vec{g}$
- aggregate state variable: $(z, g(\varepsilon, k)) \rightarrow (z, \vec{m})$.
- lom of distribution $g(\varepsilon, k) \rightarrow$ lom of moment $\vec{m}$ (*next page*)

Step 1b: Approximate law of motion

b. Approximate LOM of distribution by LOM of moments.[8] i.e.

$$m_1^{1\prime} = E(\varepsilon') = \iiint \left(\rho_\varepsilon \varepsilon + \omega_\varepsilon'\right) p\left(\omega_\varepsilon'\right) \hat{g}(\varepsilon, k) d\omega_\varepsilon' d\varepsilon dk \tag{31}$$

where $\hat{g}(\varepsilon, k)$ can be approximated by $\vec{m}$ (and coefficient $\vec{g}$).

$$\begin{aligned} m_1^{2\prime} = E(\log k') = \iiint & \left[\frac{\hat{\xi}(\varepsilon, k; z, \vec{m})}{\bar{\xi}} \log k^a(\varepsilon, k; z, \vec{m}) + \right. \\ & \left. \left(1 - \frac{\widehat{\xi}(\varepsilon, k; z, \vec{m})}{\bar{\xi}}\right) \log k^n(\varepsilon, k; z, \vec{m}) \right] p\left(\omega_\varepsilon'\right) \hat{g}(\varepsilon, k) d\omega_\varepsilon' d\varepsilon dk \end{aligned} \tag{32}$$

and

$$m_i^{j\prime} = \ldots$$

[8] In practice we approximate integrals using Gauss-Legendre quadrature at order m_g.

Step 1b: Approximate law of motion

b. Solve for steady state $\vec{m}^*$ by iterating on this LOM of moments.

Step 1: Approximate value function

c. Approximate *ex ante* firm value function $\hat{v}(\varepsilon, k; z, \vec{m})$:

$$\hat{v}(\varepsilon, k; z, \vec{m}) \approx \sum_{i=1}^{n_\varepsilon} \sum_{j=1}^{n_k} \theta_{ij}(z, \vec{m}) T_i(\varepsilon) T_j(k) \tag{33}$$

where

- n_ε, n_k: the order of approximation
- $T_i(\varepsilon), T_j(k)$: Chebyshev polynomials (or other polynomials)
- $\theta_{ij}(z, \vec{m})$: coefficients, depending on the aggregate states

Step 1: Approximate value function

c. Approximate Bellman equation:

$$
\begin{aligned}
&\widehat{v}\left(\varepsilon_i,\ k_j; z, \vec{m}\right) \\
&\quad = \lambda(z,\vec{m}) \max_n \left\{ e^z e^{\varepsilon_i} k_j^{\theta} n^{\nu} - w(z,\vec{m}) n \right\} + \lambda(z,\vec{m})(1-\delta) k_j \\
&\qquad + \left(\frac{\widehat{\xi}\left(\varepsilon_i, k_j; z, \vec{m}\right)}{\bar{\xi}} \right) \left(-\lambda(z,\vec{m}) \left(k^a\left(\varepsilon_i, k_j; z, \vec{m}\right) \right.\right. \\
&\qquad \left. + w(z,\vec{m}) \frac{\widehat{\xi}\left(\varepsilon_i, k_j; z, \vec{m}\right)}{2} \right) \\
&\qquad \left. + \beta \mathbb{E}_{z'|z} \left[\int \widehat{v}\left(\rho_\varepsilon \varepsilon_i + \sigma_\varepsilon \omega_\varepsilon', k^a\left(\varepsilon_i, k_j; z, \vec{m}\right); z', \vec{m}'(z,\vec{m}) \right) p\left(\omega_\varepsilon'\right) d\omega_\varepsilon' \right] \right) \\
&\qquad + \left(1 - \frac{\widehat{\xi}\left(\varepsilon_i, k_j; z, \vec{m}\right)}{\bar{\xi}} \right) \left(-\lambda(z,\vec{m}) k^n\left(\varepsilon_i, k_j; z, \vec{m}\right) \right. \\
&\qquad \left. + \beta \mathbb{E}_{z'|z} \left[\int \widehat{v}\left(\rho_\varepsilon \varepsilon_i + \sigma_\varepsilon \omega_\varepsilon', k^n\left(\varepsilon_i, k_j; z, \vec{m}\right); z', \vec{m}'(z,\vec{m}) \right) p\left(\omega_\varepsilon'\right) d\omega_\varepsilon' \right] \right)
\end{aligned} \tag{34}
$$

Step 1: Summary

Approximated model is characterized by:

- LoM of aggregate state (Z_t): (24)
- Aggregate state $(z, g(\varepsilon, k))$: $\rightarrow$ approx'ed aggregate state $(z, \vec{m})$
- LoM of $g(\varepsilon, k)$: (28) $\rightarrow$ LoM of $\vec{m}$: i.e. (31) and (32) etc.
- Bellman equation (26) $\rightarrow$ approx'ed Bellman equation (34)

Define a residual function based on the approximate model:

$$E_t[f(y_t, y_{t+1}, x_t, x_{t+1}; \chi)] = 0 \tag{35}$$

Step 2: Stationary Equilibrium

- 1. Given initial guess for w_0
    - compute firm's value function by iterating on the Bellman equation (34)
    - compute invariant distribution $\vec{m}^*$ by iterating on law of motion (31), using firm's decision rules;
    - aggregate individual firm's labor demand
    - compute residual from labor market clearing condition

  solve w_0^* (in*computeLMCResidualHistogram.m*).
- Solve moments $\vec{m}$ from histogram.
- Solve parameters $\vec{g}$ from histogram.
- Solve refined w^* using polynomial given w_0^*, $\vec{m}$ and $\vec{g}$ (in *computeLMCResidualPolynomial.m*)
- Compute steady state objects with refined w^*.
- Compute coefficients of policy rules (θ_{ij}).
- Compute aggregate variables

Step 3: Perturbation on Aggregate Dynamics

- Import the model to **Dynare**[9], and **Dynare** will:
- differentiate these equations;
- evaluate them at steady state;
- solve the resulting system at first order;
- perform default analysis.

[9]This step is not trivial. Dynare does not accept matrix expressions used heavily in the matlab codes to solve steady state. We need to re-write teh matrix expressions as loops over scalar variables using Dynare's macro-processor.
For details on macro-processor, see: http://www.dynare.org/manual/index_37.html; or at: http://www.dynare.org/DynareShanghai2013/macroprocessor.pdf

Reference and Further Reading

Reference

- Winberry, T. (2018). A method for solving and estimating heterogeneous agent macro models. Quantitative Economics.

which partly builds upon the algorithm in:

- Reiter, M. (2009). Solving heterogeneous-agent models by projection and perturbation. Journal of Economic Dynamics and Control, 33(3), 649-665.
- Algan, Y., Allais, O., & Den Haan, W. J. (2010). Solving the incomplete markets model with aggregate uncertainty using parameterized cross-sectional distributions. Journal of Economic Dynamics and Control, 34(1), 59-68.

Reference and Further Reading

Further Reading for P+P Algorithm/Application

- (review) Terry, S. J. (2017). Alternative methods for solving heterogeneous firm models. Journal of Money, Credit and Banking, 49(6), 1081-1111.
- (review) Algan, Y., Allais, O., Den Haan, W. J., & Rendahl, P. (2014). Solving and simulating models with heterogeneous agents and aggregate uncertainty. In Handbook of Computational Economics (Vol. 3, pp. 277-324). Elsevier.
- (application) Winberry, T. (2020). Lumpy Investment, Business Cycles, and Stimulus Policy. American Economic Review.