Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models

Jacob Marott Sundram

University of Copenhagen

October 7, 2024

Idea

- **Goal:** Solve for the transition path of (possibly heterogeneous-agent) models in general equilibrium with aggregate shocks.
- **Reiter method:** Linearize and solve linear state space system.
- **MIT shock method:** Assume perfect foresight and solve non-linear sequence space system.
- These are the same for small shocks due to certainty equivalence!
- **This paper:** Solve linear sequence space system $\Rightarrow$ Faster.

Setup

- Infinite horizon, discrete time: $t = 0, 1, \ldots$
- A model is nothing but a system of non-linear equations,

$$\boldsymbol{F}_t(\boldsymbol{X}, \boldsymbol{Z}) = \boldsymbol{0}.$$

  - $\boldsymbol{X} = (X_0, X_1, \ldots)$ with $\boldsymbol{X}_t = (X_t^1, \ldots, X_t^{n_x})'$ an $n_x \times 1$ vector of variables.
  - $\boldsymbol{Z} = (Z_0, Z_1, \ldots)$ with $\boldsymbol{Z}_t = (Z_t^1, \ldots, Z_t^{n_z})'$ an $n_z \times 1$ vector of shocks.
- $\boldsymbol{X^i} = (X_0^i, X_1^i, \ldots)'$ and $\boldsymbol{Z^i} = (Z_0^i, Z_1^i, \ldots)'$ are *sequences* $\Rightarrow$ We are representing the model in the sequence space (rather than the state space).

Example: Ramsey model

$$
\begin{aligned}
r_t^K &= \alpha \Gamma_t K_t^{\alpha-1} L_t^{1-\alpha}, && \text{(Firm FOC for capital)} \\
w_t &= (1-\alpha) \Gamma_t K_t^{\alpha} L_t^{-\alpha}, && \text{(Firm FOC for labor)} \\
r_t &= r_t^K - \delta, && \text{(Depreciation)} \\
A_t &= K_t, && \text{(Asset supply)} \\
A_t^{hh} + C_t^{hh} &= (1+r_t) A_{t-1}^{hh} + w_t, && \text{(Budget constraint)} \\
(C_t^{hh})^{-\sigma} &= \beta (1 + r_{t+1}) (C_{t+1}^{hh})^{-\sigma}, && \text{(Euler equation)} \\
A_t &= A_t^{hh}, && \text{(Asset market clearing)} \\
L_t &= 1. && \text{(Labor market clearing)}
\end{aligned}
$$

Move all terms to left-hand side

$$
\begin{aligned}
r_t^K - \alpha \Gamma_t K_t^{\alpha-1} L_t^{1-\alpha} &= 0, && \text{(Firm FOC for capital)} \\
w_t - (1-\alpha)\Gamma_t K_t^{\alpha} L_t^{-\alpha} &= 0, && \text{(Firm FOC for labor)} \\
r_t - (r_t^K - \delta) &= 0, && \text{(Depreciation)} \\
A_t - K_t &= 0, && \text{(Asset supply)} \\
A_t^{hh} + C_t^{hh} - \left[(1+r_t)A_{t-1}^{hh} + w_t\right] &= 0, && \text{(Budget constraint)} \\
(C_t^{hh})^{-\sigma} - \beta(1+r_{t+1})(C_{t+1}^{hh})^{-\sigma} &= 0, && \text{(Euler equation)} \\
A_t - A_t^{hh} &= 0, && \text{(Asset market clearing)} \\
L_t - 1 &= 0. && \text{(Labor market clearing)}
\end{aligned}
$$

Stack as a vector

$$\begin{pmatrix} r_t^K - \alpha \Gamma_t K_t^{\alpha-1} L_t^{1-\alpha} \\ w_t - (1-\alpha)\Gamma_t K_t^{\alpha} L_t^{-\alpha} \\ r_t - (r_t^K - \delta) \\ A_t - K_t \\ A_t^{hh} + C_t^{hh} - \left[(1+r_t)A_{t-1}^{hh} + w_t\right] \\ (C_t^{hh})^{-\sigma} - \beta(1+r_{t+1})(C_{t+1}^{hh})^{-\sigma} \\ A_t - A_t^{hh} \\ L_t - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \boldsymbol{F}_t(\boldsymbol{X}, \boldsymbol{Z}) = \boldsymbol{0}.$$

$n_x = 8$ variables $\boldsymbol{X}_t = (r_t^K, w_t, L_t, K_t, r_t, A_t, C_t^{hh}, A_t^{hh})$, $n_z = 1$ shocks $\boldsymbol{Z}_t = (\Gamma_t)$.

Representation on a computer

- **Goal:** We want to solve the model on a computer.
- **Problem:** The time horizon is infinite.
- **Solution:** Truncate at some T, say $T = 500$.
- In this case, stack $\boldsymbol{F}_t(\boldsymbol{X}, \boldsymbol{Z})$ for $t = 0, 1, \ldots$ to get

$$\boldsymbol{F}(\boldsymbol{X}, \boldsymbol{Z}) = \boldsymbol{0}.$$

  - $\boldsymbol{F}(\boldsymbol{X}, \boldsymbol{Z})$ has dimension $n_x \times T$.
  - $\boldsymbol{X} = (X_0, X_1, \ldots)$ has dimension $n_x \times T$.
  - $\boldsymbol{X} = (Z_0, Z_1, \ldots)$ has dimension $n_z \times T$.

Dimension reduction

- **Problem:** Many endogenous variables $\Rightarrow$ Huge system of equations.
- **Solution:** Reduce dimension. Suppose $\boldsymbol{F}(\boldsymbol{X}, \boldsymbol{Z}) = \mathbf{0}$ can be separated into

$$\boldsymbol{F}_1(\boldsymbol{X}, \boldsymbol{Z}) = \mathbf{0} \quad \text{and} \quad \boldsymbol{F}_2(\boldsymbol{X}, \boldsymbol{Z}) = \mathbf{0},$$

  where the second equation can be solved as a function of some smaller vector $\boldsymbol{U}$ of $n_u < n_x$ unknowns: $\boldsymbol{X} = \boldsymbol{M}(\boldsymbol{U}, \boldsymbol{Z})$.
- Define then $\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{Z}) \equiv \boldsymbol{F}_1(\boldsymbol{M}(\boldsymbol{U}, \boldsymbol{Z}), \boldsymbol{Z})$ to get the reduced system:

$$\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{Z}) = \mathbf{0}.$$

Example: Ramsey model

- In the Ramsey model, set $\boldsymbol{U}_t = (K_t)$, i.e. $n_u = 1$.
- We then get $\boldsymbol{X} = \boldsymbol{M}(\boldsymbol{U}, \boldsymbol{Z})$ directly without any algebra.
- The target is then reduced to:

$$H_t(\boldsymbol{U}, \boldsymbol{Z}) = A_t(K_t) - A_t^{hh}(\boldsymbol{X}(K_t)).$$

Solving for the transition path

- **Goal:** Transition path, $\boldsymbol{X}$, given shocks, $\boldsymbol{Z}$.
- Implicit function theorem on $\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{Z}) = \boldsymbol{0}$ gives

$$\boldsymbol{H}_{\boldsymbol{U}} d\boldsymbol{U} + \boldsymbol{H}_{\boldsymbol{Z}} d\boldsymbol{Z} = \boldsymbol{0},$$

  where $\boldsymbol{A}_{\boldsymbol{B}}$ is short-hand for $\partial \boldsymbol{A} / \partial \boldsymbol{B}$ for any $\boldsymbol{A}, \boldsymbol{B}$.
- Solve for $d\boldsymbol{U}$ and then $d\boldsymbol{X}$ follows:

$$\begin{aligned} d\boldsymbol{U} &= -\boldsymbol{H}_{\boldsymbol{U}}^{-1} \boldsymbol{H}_{\boldsymbol{Z}} d\boldsymbol{Z}, \\ d\boldsymbol{X} &= \boldsymbol{M}_{\boldsymbol{U}} d\boldsymbol{U} + \boldsymbol{M}_{\boldsymbol{Z}} d\boldsymbol{Z} = \boldsymbol{G} d\boldsymbol{Z}. \end{aligned}$$

Sequence space Jacobians

- **Takeaway:** Finding the transition path amounts to finding *sequence space Jacobians* ($\boldsymbol{H_U}$, $\boldsymbol{H_Z}$, $\boldsymbol{M_U}$, and $\boldsymbol{M_Z}$).
- There are multiple ways of doing so:
  1. By hand.
  2. Composition of Jacobians of smaller blocks along directed acyclical graph.
  3. Automatic differentiation.
  4. Numerical differentiation.
- Either way, the transition paths are *linearized*. But we can also use Jacobians to get full *non-linear* transition path.

Non-linear transition paths: Idea

Input: Paths of shocks.

1. Guess on paths of n_u unknowns.
2. Get paths of all n_x endogenous variables using analytical solution.
3. Return paths for n_u targets.
4. If all targets $= \mathbf{0}$, the model is solved. Otherwise, return to 1.

Example: Ramsey model

We know shock: $\mathbf{\Gamma} = (\Gamma_0, \Gamma_1, \dots)'$.

1. Guess on $\boldsymbol{K} = (K_0, K_1, \dots)'$.

Example: Ramsey model

2. Get paths of all other endogenous variables for $t = 0, 1, \dots, T-1$:

$$
\begin{aligned}
L_t &= 1, \\
A_t &= K_t, \\
r_t^K &= \alpha \Gamma_t K_t^{\alpha-1} L_t^{1-\alpha}, \\
w_t &= (1-\alpha) \Gamma_t K_t^{\alpha} L_t^{-\alpha}, \\
r_t &= r_t^K - \delta, \\
(C_t^{hh}) &= \beta (1 + r_{t+1}) (C_{t+1}^{hh})^{-\sigma}, \quad \text{backwards}, \\
A_t^{hh} &= (1 + r_t) A_{t-1}^{hh} + w_t - C_t^{hh}, \quad \text{forwards}.
\end{aligned}
$$

Example: Ramsey model

3. Return $(\boldsymbol{A} - \boldsymbol{A^{hh}}) = (A_0 - A_0^{hh}, A_1 - A_1^{hh}, \dots)'$.

- A decent starting guess is $\boldsymbol{K} = (K_{ss}, K_{ss}, \dots)'$.
- **Problem:** In the presence of shocks, $A_t \neq A_t^{hh}$.
- How to update the guess for the next iteration?

Non-linear transition paths: Updating the guess

- We are trying to solve $\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{Z}) = \mathbf{0}$ for $\boldsymbol{U}$ with some fixed $\boldsymbol{Z}$.
- First-order approximation in $\boldsymbol{U}$ around guess $\boldsymbol{U} = \boldsymbol{U_g}$:

$$\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{Z}) \approx \boldsymbol{H}(\boldsymbol{U_g}, \boldsymbol{Z}) + \boldsymbol{H_U}(\boldsymbol{U} - \boldsymbol{U_g}).$$

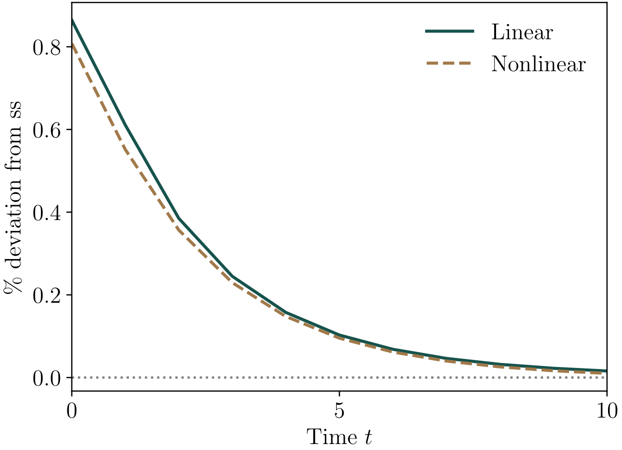
- Solve for $\boldsymbol{U}$:

$$\boldsymbol{U} \approx \boldsymbol{U_g} - \boldsymbol{H}_{\boldsymbol{U}}^{-1}\boldsymbol{H}(\boldsymbol{U_g}, \boldsymbol{Z}).$$

- Use this as the next guess and repeat. Can use Broyden's method to speed up.

Linear vs. non-linear transition paths

(a) Consumption after shocks to Taylor rule



Heterogeneous agent models

- What about heterogeneous agent (HA) model?
- **Difference:** We do not know A_t^{hh} analytically.
- **Solution:** Write

$$
\begin{aligned}
A_t^{hh} &= \boldsymbol{a}_t' \boldsymbol{D}_t, \\
\boldsymbol{D}_{t+1} &= \Lambda_t \boldsymbol{D}_t.
\end{aligned}
$$

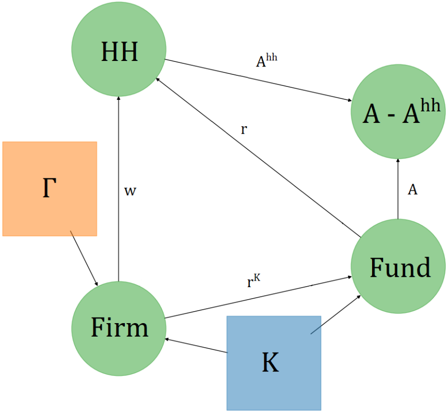
  where a_t is the savings policy function, D_t is the distribution of households, and Λ_t is the transition matrix for the distribution.
- a_t and D_t can be found using standard methods.
- How to get $\boldsymbol{H_U}$? Can do it numerically. Fast way: Use DAG + fake news.

DAGs and blocks

- Directed acyclical graph (DAG): How shocks and unknowns propagate through the model to targets.
- Components:
  - Shocks (exogenous).
  - Unknowns (endogenous).
  - Blocks: Ordered partition of model that takes inputs and returns outputs.
- Key properties of blocks:
  - Each output is unique to a block.
  - The first block has only shocks and unknowns as inputs.
  - Later blocks only additionally take outputs of previous.

Example: HA-Ramsey

$$
\begin{aligned}
\boldsymbol{w}, \boldsymbol{r}^{\boldsymbol{K}} &= \text{Firm}(\boldsymbol{K}), \\
\boldsymbol{A}, \boldsymbol{r} &= \text{Mutual Fund}(\boldsymbol{K}, \boldsymbol{r}^{\boldsymbol{K}}), \\
\boldsymbol{A}^{\boldsymbol{hh}} &= \text{HH}(\boldsymbol{r}, \boldsymbol{w}), \\
\boldsymbol{A} - \boldsymbol{A}^{hh} &= \boldsymbol{H}(\boldsymbol{A}, \boldsymbol{A}^{\boldsymbol{hh}}).
\end{aligned}
$$



Example: HA-Ramsey DAG

- **Notation:** $\mathcal{J}^{y,x}$ is the Jacobian of y w.r.t. x
- Chain rule along DAG:

$$\begin{aligned}
\boldsymbol{H_K} &= \mathcal{J}^{A-A^{hh},K} \\
&= \mathcal{J}^{A-A^{hh},A}\,\mathcal{J}^{A,K} \\
&\quad + \mathcal{J}^{A-A^{hh},A^{hh}}\,\mathcal{J}^{A^{hh},r}\,\mathcal{J}^{r,r^K}\,\mathcal{J}^{r^K,K} \\
&\quad + \mathcal{J}^{A-A^{hh},A^{hh}}\,\mathcal{J}^{A^{hh},w}\,\mathcal{J}^{w,K}.
\end{aligned}$$

Jacobians of household problem

- How to get Jacobians of household problem, i.e. of $\boldsymbol{Y^{hh}} = \boldsymbol{HH}(\boldsymbol{i}_1, \ldots, \boldsymbol{i}_n)$?
- **Naive approach:** For each input i into HH block:
  - For each $s = 0, 1, \ldots, T-1$:
    1. Shock each input i in time s by small $\epsilon > 0$.
    2. Solve household problem backwards along transition path.
    3. Simulate households forward along transition path.
    4. Calculate column s, row t as

$$\frac{\partial \mathcal{J}_t^{Y^{hh},i}}{\partial i_s} = \frac{Y_t^{hh} - Y_{ss}^{hh}}{\epsilon}.$$

- This is very slow. **Solution:** Fake news algorithm.

Fake news algorithm: Goal

- Consider household block: $\boldsymbol{Y}^{hh} = \boldsymbol{HH}(\boldsymbol{i}_1, \ldots, \boldsymbol{i}_n)$.
- HA-Ramsey: $\boldsymbol{A}^{hh} = \boldsymbol{HH}(\boldsymbol{w}, \boldsymbol{r})$.
- We are interested in HH Jacobian, for instance w.r.t. r:

$$\mathcal{J} \equiv \frac{\partial \boldsymbol{A}^{hh}}{\partial \boldsymbol{r}}.$$

Fake news algorithm: Fake news matrix

- Define auxiliary *fake news matrix* $\mathcal{F}$ by:

$$\mathcal{F}_{t,s} \equiv \begin{cases} \frac{\partial A_t^{hh}}{\partial r_s}, & \text{if } s = 0 \text{ or } t = 0 \\ \frac{\partial A_t^{hh}}{\partial r_s} - \frac{\partial A_{t-1}^{hh}}{\partial r_{s-1}} & \text{else} \end{cases}.$$

- At time $t = 0$, there is a news shock that r changes in period s: $\mathcal{F}_{0,s} = \frac{\partial A_0^{hh}}{\partial r_s}$.
- At time $t = 1$, there is a news shock that r does not change after all:

$$\mathcal{F}_{1,s} = \frac{\partial A_1^{hh}}{\partial r_s} - \frac{\partial A_0^{hh}}{\partial r_{s-1}}.$$

Fake news algorithm: A trick

- Why is this useful? If we have $\mathcal{F}$, we we have $\mathcal{J}$:

$$\mathcal{F} = \begin{pmatrix} \mathcal{J}_{00} & \mathcal{J}_{01} & \mathcal{J}_{02} & \cdots \\ \mathcal{J}_{10} & \mathcal{J}_{11} - \mathcal{J}_{00} & \mathcal{J}_{12} - \mathcal{J}_{01} & \cdots \\ \mathcal{J}_{20} & \mathcal{J}_{21} - \mathcal{J}_{10} & \mathcal{J}_{22} - \mathcal{J}_{11} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad \Rightarrow \quad \mathcal{J} = \begin{pmatrix} \mathcal{J}_{00} & \mathcal{J}_{01} & \mathcal{J}_{02} & \cdots \\ \mathcal{J}_{10} & \mathcal{J}_{11} & \mathcal{J}_{12} & \cdots \\ \mathcal{J}_{20} & \mathcal{J}_{21} & \mathcal{J}_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

- This is only useful if it is easier to get $\mathcal{F}$ than $\mathcal{J}$.
- **It is!** $\mathcal{F}$ requires only a single backwards and forwards iteration.
- To see this, consider how to get $\mathcal{F}$, i.e. the Jacobian w.r.t. fake news shocks.

Fake news algorithm: Backwards iteration

- **Claim:** Only need a single backwards iteration.
- Why? Only time $s - t$ until perturbation matters:

$$\boldsymbol{a}_t^s = \begin{cases} \boldsymbol{a}_{ss} & \text{if } t > s, \\ \boldsymbol{a}_{t+j}^{s+j} & \text{if } t \leq s \end{cases},$$

  where $\boldsymbol{a}_t^s$ is the policy function at time t following a shock at time s.
- **Intuitively:** The response of the policy function at time t to a shock at s is the same as the response at time $t + j$ to a shock $s + j$.
- This is really just using recursivity from the dynamic programming problem.

Fake news algorithm: Backwards iteration

- **Implication:** Only need a single backward iteration with $s = T - 1$ to get all $\boldsymbol{a}_t^s$.
- To see this, do backwards iteration with $s = T - 1$ to get:

$$(\boldsymbol{a}_0^{T-1}, \boldsymbol{a}_1^{T-1}, \ldots, \boldsymbol{a}_{T-1}^{T-1}).$$

- Then we get directly:

$$\boldsymbol{a}_t^s = \begin{cases} \boldsymbol{a}_{ss} & \text{if } t > s, \\ \boldsymbol{a}_{(T-1)-(s-t)}^{T-1} & \text{if } t \leq s. \end{cases}$$

for all t, s.

Fake news algorithm: Forwards iteration

- **Claim:** We only need one forwards iteration.
- To see this, note that at $t = 0$, the distribution is in steady state, $\boldsymbol{D}_0^s = \boldsymbol{D}_{ss}$. Thus, we get savings,

$$(A_0^{hh})^s = (\boldsymbol{a}_0^s)'\boldsymbol{D}_{ss}, \quad \text{for } s = 0, 1, \ldots,$$

  and resulting distribution, $\boldsymbol{D}_1^s$.
- At $t = 1$, policies are in steady state $\Rightarrow$ The transition matrix for the distribution is the steady state one. Iterate forward *once* to get remaining distribution for $t = 2, 3, \ldots$: $\boldsymbol{D}_t^s = \Lambda_{ss}\boldsymbol{D}_{t-1}^s$.

Fake news algorithm: Finishing up

- We then get

$$(A_t^{hh})^s = (\boldsymbol{a}_t^s)'\boldsymbol{D}_t^s.$$

- Thus, we have savings at all points in time, $t = 0, 1, \ldots,$ for all possible news shocks at time $s = 0, 1, \ldots$. The entries of the fake news matrix are then

$$\mathcal{F}_{t,s} = \frac{(A_t^{hh})^s - A_{ss}^{hh}}{\epsilon},$$

  where ϵ is the pertubation to r_s.

- As discussed before, $\mathcal{J}$ then follows from $\mathcal{F}$.

Simulation and estimation

- Once you have $\boldsymbol{G}$, many things follow trivially.
- **Simulation:** Can simulate very long time series very fast.
- **Moments:** Can compute auto-covariances very fast.
- **Maximum likelihood:** Can evaluate likelihood very fast and therefore do maximum likelihood estimation fast.
- **Bayesian estimation:** Can do Bayesian estimation fast.

Summary

- To solve for transition paths, we need *sequence space Jacobians*.
- Auclert et. al (2021) introduce an efficient algorithm to do so.
- **Central insights:**
  - Compose Jacobians of blocks using the chain rule.
  - Exploit structure of dynamic programming problem (fake news algorithm).
- Everything follows: Linear IRFs, non-linear IRFs, simulation, estimation, etc.