
Using Sequence Space Jacobians to Solve Heterogeneous Agent Models

Auclert, Bardóczy, Rognlie & Straub (2021, ABRS)

741 Macroeconomics
Topic 9

Masao Fukui

Fall 2025

Question

- Modern macro puts emphasis on heterogeneity
 - households' income/wealth/consumption/friction
 - firms' productivity/size/friction
 - space/sectors/occupations
- Many interesting macro questions are about dynamics
 - IRF to monetary/fiscal policy shocks or business cycle shocks
 - Transition dynamics following the technological/policy change
- How do we solve the dynamics of macro models with heterogeneity?

Hugget Model (from 704)

■ Households solve

$$\begin{aligned} V_t(a_t, e_t) = \max_{c_t, a_{t+1} \geq -\phi} & u(c_t) + \beta \mathbb{E}_t V_{t+1}(a_{t+1}, e_{t+1}) \\ \text{s.t. } & c_t + a_{t+1} = (1 + r_{t-1})a_t + e_t Y_t \end{aligned}$$

- Policy functions: $c_t(a, e), a_{t+1}(a, e)$
- Normalize $\mathbb{E}e = 1$ unconditionally

■ Distribution evolves

$$\mu_{t+1}(a, e_j) = \sum_i \mu_t(a_{t+1}^{-1}(a, e_i), e_i) \text{Prob}(e_j | e_i)$$

■ Goods market clears (bonds market clears by Warlas' law):

$$\int c_t(a, e) d\mu_t = Y_t$$

Equilibrium Counterfactuals

Neoclassical:

- Given $\{Y_t\}$, recursive eqm consists of $\{c_t, a_{t+1}, V_t\}$, $\{\mu_t\}$, and $\{r_t\}$ such that
 - $\{c_t, a_{t+1}, V_t\}$ solve Bellman & $\{\mu_t\}$ follows the law of motion
 - Goods market clears

HANK:

- Given $\{r_t\}$, recursive eqm consists of $\{c_t, a_{t+1}, V_t\}$, $\{\mu_t\}$, and $\{Y_t\}$ such that
 - $\{c_t, a_{t+1}, V_t\}$ solve Bellman & $\{\mu_t\}$ follows the law of motion
 - Goods market clears

Q: How do we compute the IRFs in response to $\{Y_t\}$ shocks or $\{r_t\}$ shocks?

Two Approaches

1. State-space approach (Krusell & Smith, 1998; Reiter, 2009)

- Find the relevant state variables and drop time subscript t
- Dominant approach when I was a grad student

2. Sequence-space approach (Boppart, Krusell & Mitman, 2018; ABRS, 2021)

- Carry t and solve for a sequence of $\{r_t\}$ or $\{Y_t\}$
- State-of-the-art method

State-Space Approach in One Slide

Aggregate state variable: distribution $\mu_t(a, e)!$

- No household directly cares about distribution – they care about $\{r_t\}$ and $\{Y_t\}$
- Why is distribution $\mu_t(a, e)$ a state variable?
 - because they need distribution to forecast the future $\{r_t\}$ or $\{Y_t\}$
- μ is an infinite-dimensional object
 - ⇒ infeasible to carry μ_t as a state variable on the computer
- Needs a dimensional reduction:
 1. Krusell-Smith (1998): Replace μ_t with the first K moments ($K = 1$ in practice)
 2. Reiter (2009): Replace μ_t with the histogram & linearize w.r.t. agg. shocks
- Both are computationally expensive

Sequence-Space Approach

Sequence Space Approach

- Define the aggregate consumption function as

$$\mathcal{C}_t(r, Y) = \int c_t(a, e) d\mu_t$$

where $r \equiv [r_t]_t$ and $Y \equiv [Y_t]_t$

- The key observation: \mathcal{C} depends only on a small number of aggregate vectors
 - Crucially, it does not (directly) depend on distribution μ
- The goal is to find r or Y that solve

$$\mathcal{C}_t(r, Y) = Y_t$$

which we can stack across t to write

$$\mathcal{C}(r, Y) = Y$$

Linearized Solution

- We look for the linearized solutions around the steady state w.r.t. agg. shocks:

$$M^r dr + M^Y dY = dY$$

where $M^r \equiv \left[\frac{\partial \mathcal{C}_t}{\partial r_s} \right]_{t,s}$ and $M^Y \equiv \left[\frac{\partial \mathcal{C}_t}{\partial Y_s} \right]_{t,s}$ are **sequence-space Jacobians (SSJ)**

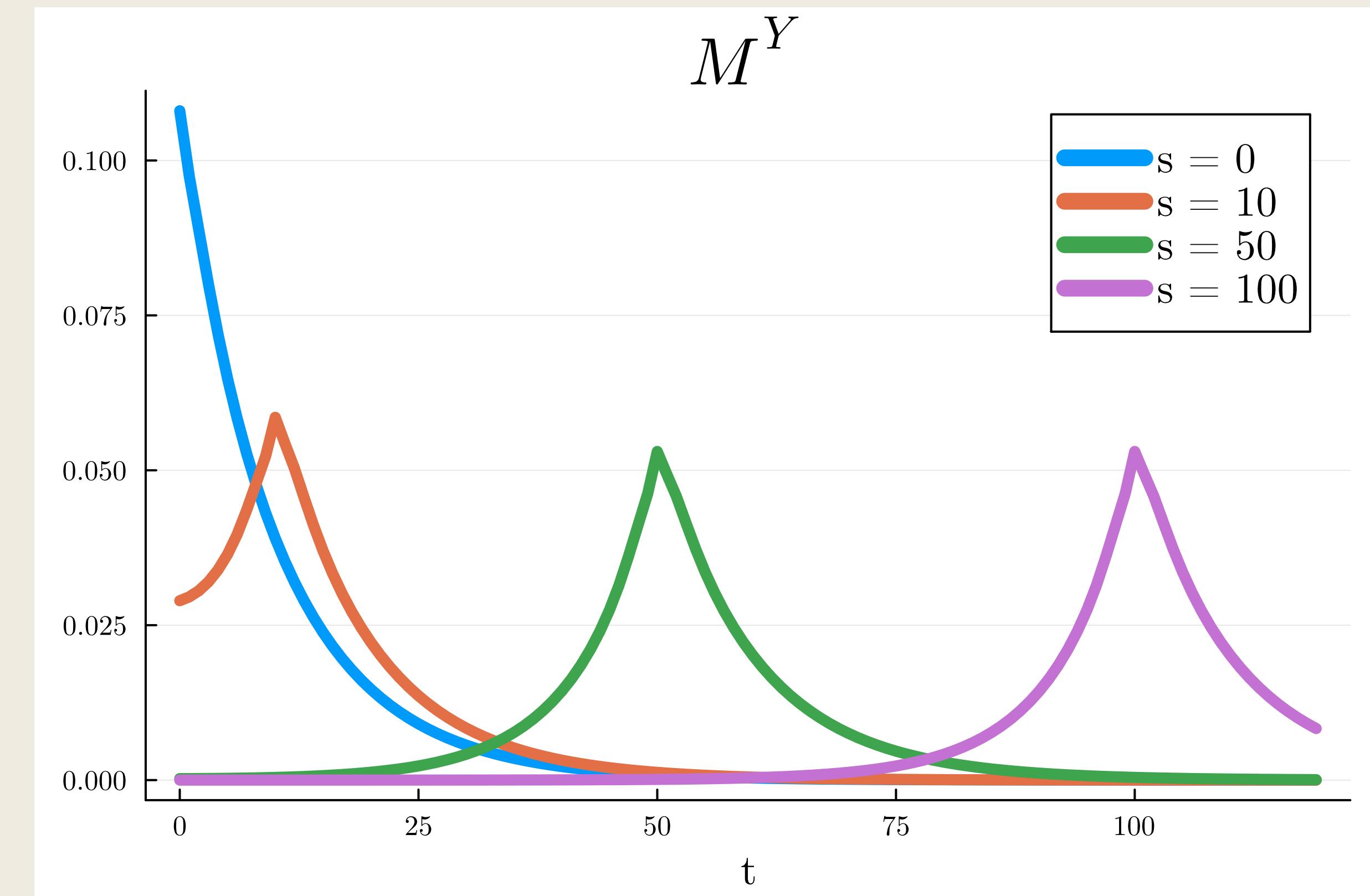
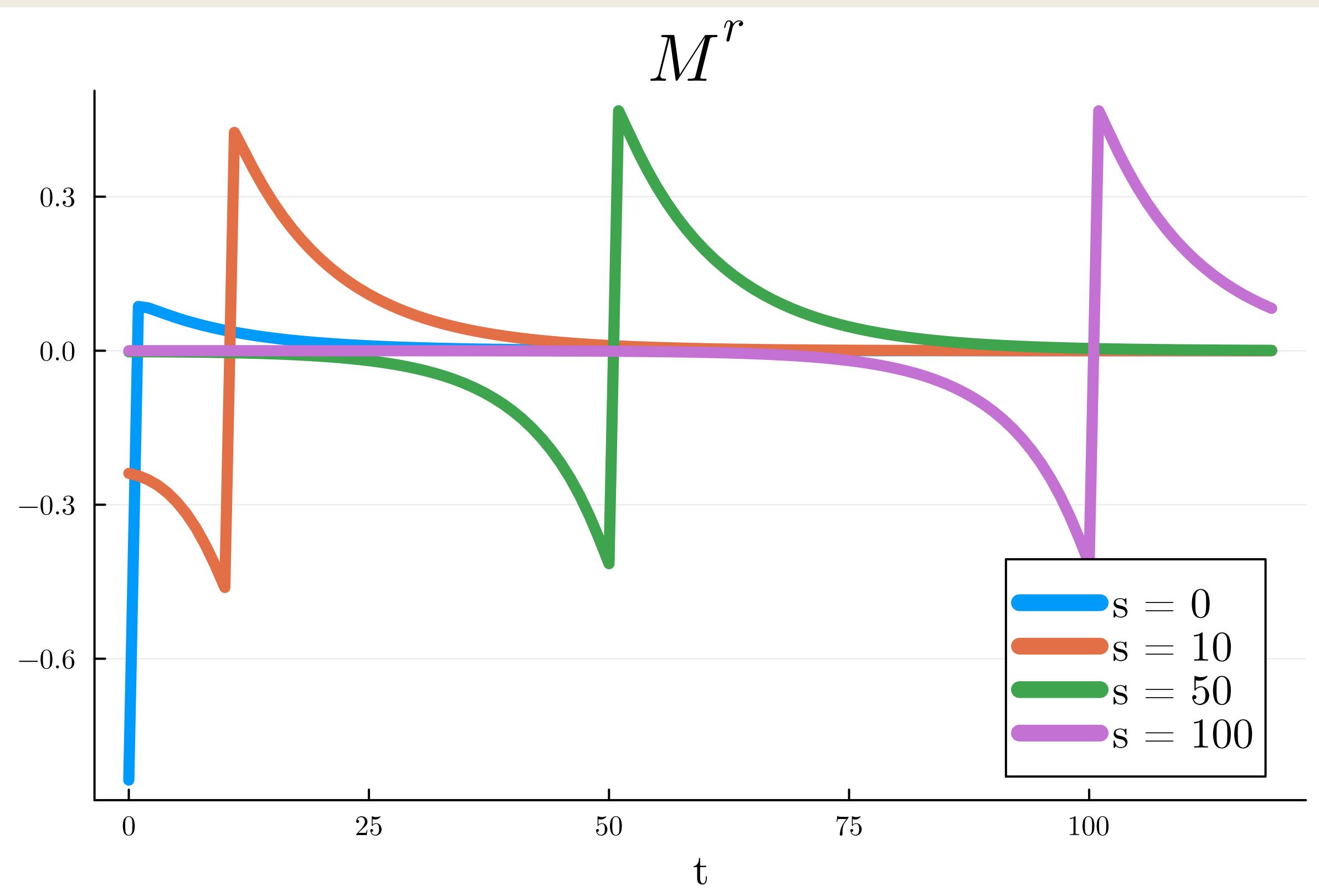
- Both M^r and M^Y have clear economic interpretations:
 - (t, s) element of M^r : PE response of agg. C at time t to an increase in r at time s
 - (t, s) element of M^Y : PE response of agg. C at time t to an increase in Y at time s
- If we know the SSJ, we can solve the linear systems to find the solutions:

$$dr = [M^r]^{-1} [dY - M^Y dY] \quad \text{for neoclassical}$$

or

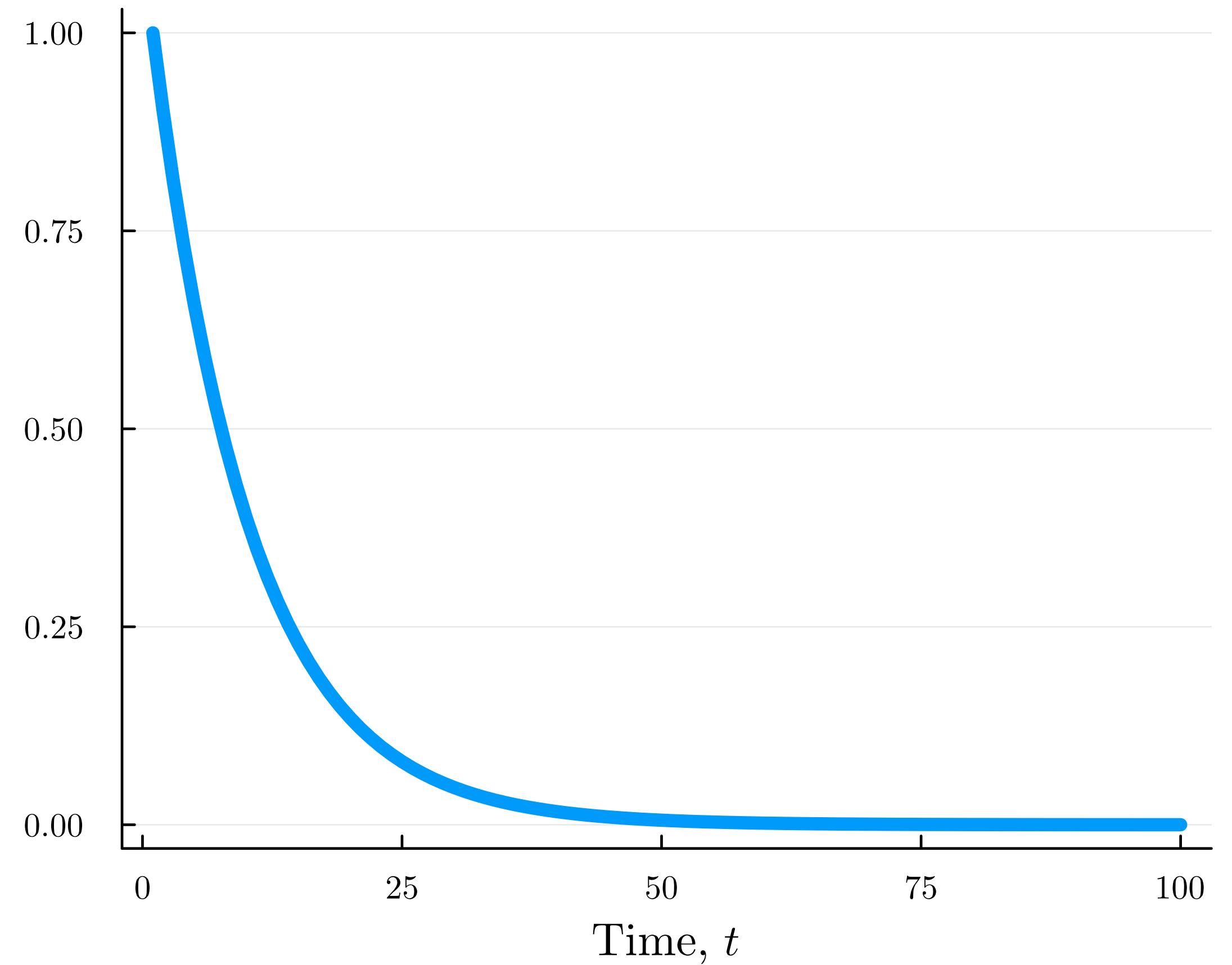
$$dY = [I - M^Y]^{-1} [M^r dr] \quad \text{for HANK}$$

Visualizing M^r and M^Y

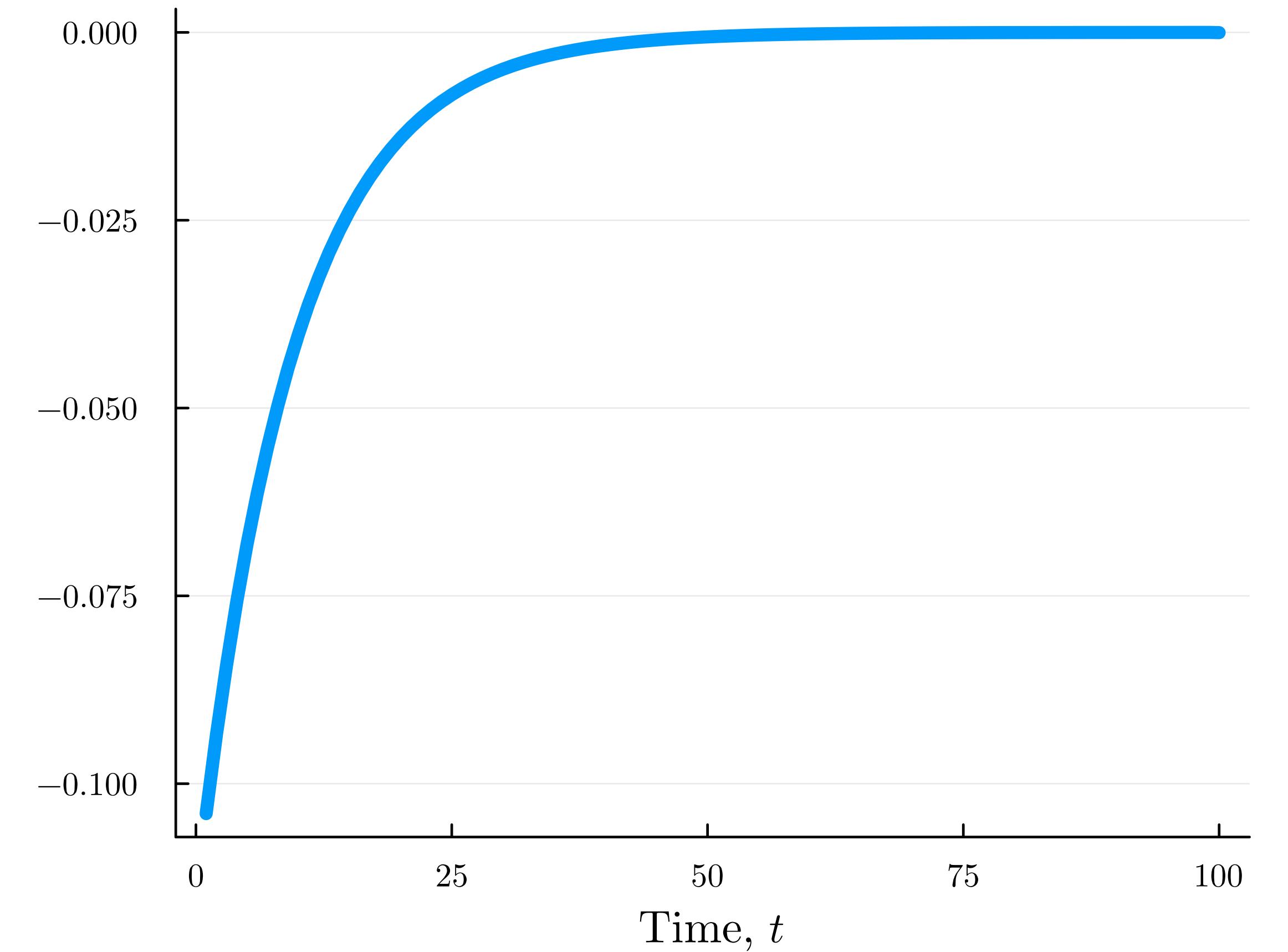


IRF of Neoclassical Economy

Output

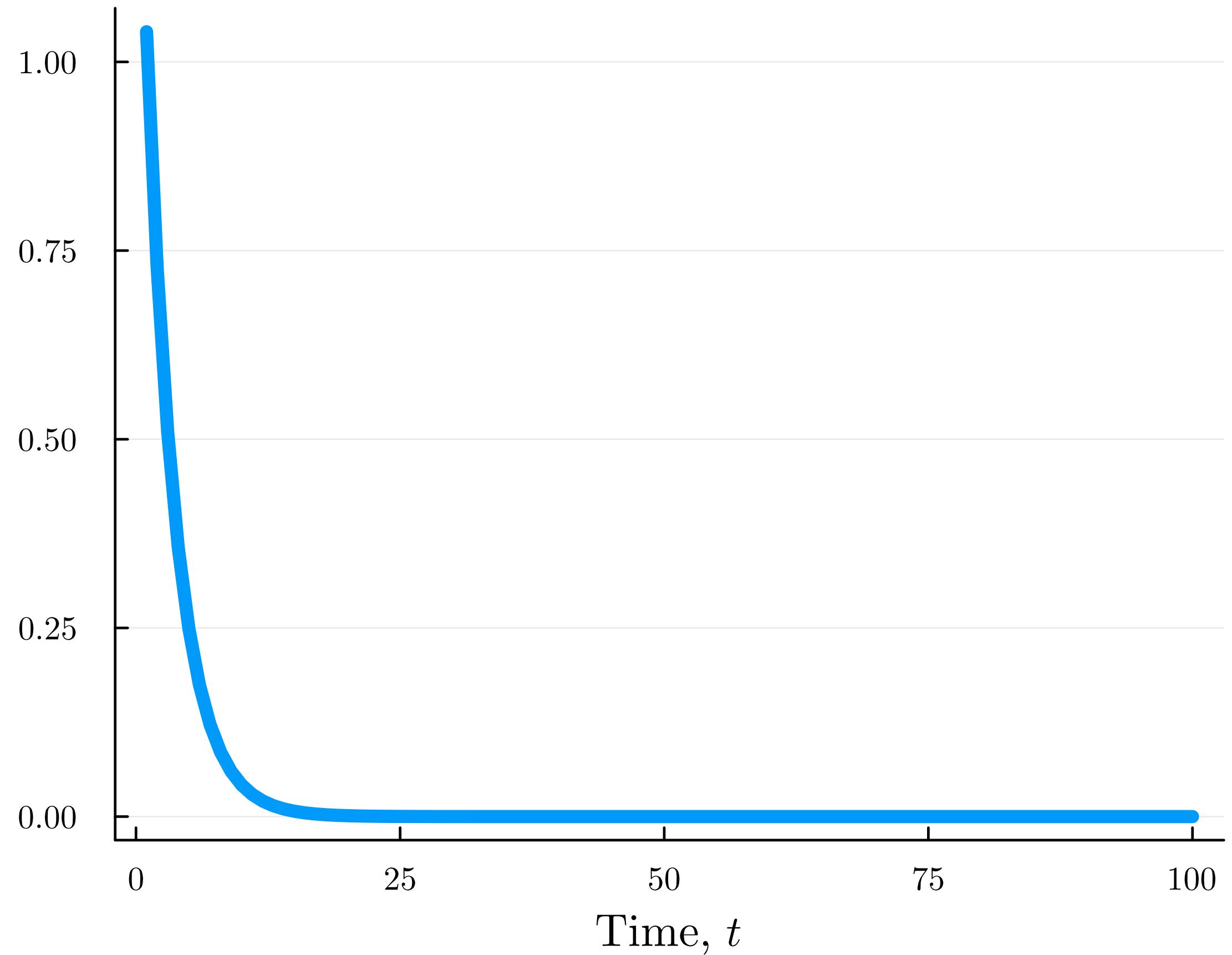


Interest Rate

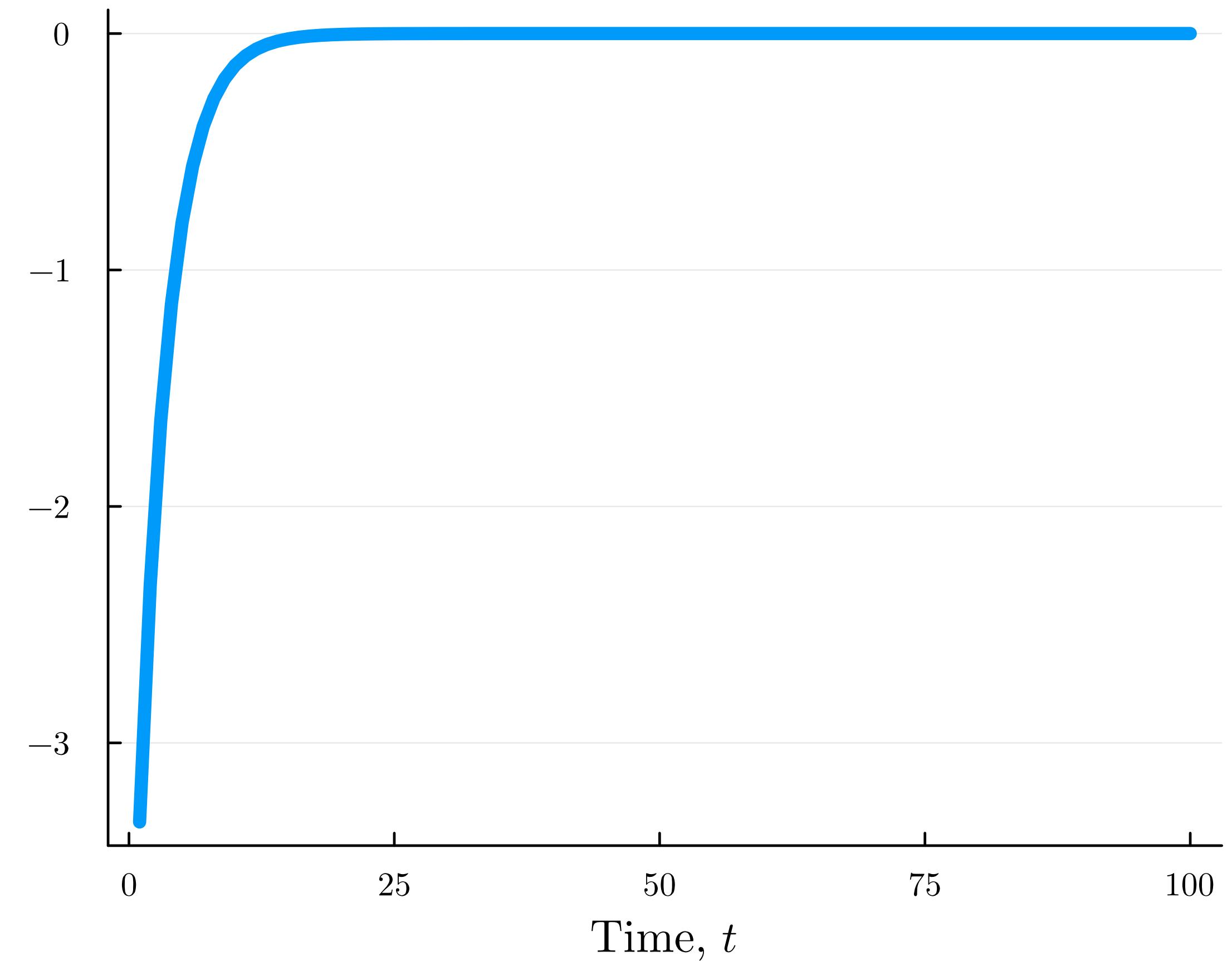


IRF of HANK

Interest Rate



Output



How Do We Know SSJ?

- First, SSJs are infinite-dimensional objects, so truncate to $T \times T$ matrices
 - This is the only approximation involved in SSJ method
- How do we obtain M_r and M_Y ?
- The brute force approach:

$$\mathbf{r} := \begin{bmatrix} r + dr_0 \\ r \\ \vdots \\ r \end{bmatrix}$$

$\overbrace{\hspace{10em}}$
 $T \times 1$ vector

How Do We Know SSJ?

- First, SSJs are infinite-dimensional objects, so truncate to $T \times T$ matrices
 - This is the only approximation involved in SSJ method
- How do we obtain M_r and M_Y ?
- The brute force approach:

$$\mathbf{r} := \begin{bmatrix} r + dr_0 \\ r \\ \vdots \\ r \end{bmatrix} \underbrace{\qquad\qquad\qquad}_{T \times 1 \text{ vector}}$$

Set $c_{T+1} = c_{ss}$
for $t = T : -1 : 1$
backward iteration of
the Euler:
 $u'(c_t) \geq \beta(1 + r_t)\mathbb{E}u'(c_{t+1})$
 $c_{t+1} \Rightarrow c_t$

How Do We Know SSJ?

- First, SSJs are infinite-dimensional objects, so truncate to $T \times T$ matrices
 - This is the only approximation involved in SSJ method
- How do we obtain M_r and M_Y ?
- The brute force approach:

$$\mathbf{r} := \begin{bmatrix} r + dr_0 \\ r \\ \vdots \\ r \end{bmatrix} \underbrace{\qquad\qquad\qquad}_{T \times 1 \text{ vector}}$$

Set $c_{T+1} = c_{ss}$
for $t = T : -1 : 1$
backward iteration of
the Euler:
 $u'(c_t) \geq \beta(1 + r_t)\mathbb{E}u'(c_{t+1})$
 $c_{t+1} \Rightarrow c_t$

Given $\mu_0 \equiv \mu_{ss}$ &
 $\{c_t, a_{t+1}\}_t$,
for $t = 1 : T$
forward iteration
of the law of
motion:
 $\mu_{t-1} \Rightarrow \mu_t$

How Do We Know SSJ?

- First, SSJs are infinite-dimensional objects, so truncate to $T \times T$ matrices
 - This is the only approximation involved in SSJ method
- How do we obtain M_r and M_Y ?
- The brute force approach:

$$\mathbf{r} := \begin{bmatrix} r + dr_0 \\ r \\ \vdots \\ r \end{bmatrix} \underbrace{\qquad\qquad\qquad}_{T \times 1 \text{ vector}}$$

Set $c_{T+1} = c_{ss}$
for $t = T : -1 : 1$
backward iteration of
the Euler:
 $u'(c_t) \geq \beta(1 + r_t)\mathbb{E}u'(c_{t+1})$
 $c_{t+1} \Rightarrow c_t$

Given $\mu_0 \equiv \mu_{ss}$ &
 $\{c_t, a_{t+1}\}_t$,
for $t = 1 : T$
forward iteration
of the law of
motion:
 $\mu_{t-1} \Rightarrow \mu_t$

Given $\{c_t, \mu_t\}$
 $\Rightarrow C_t = \int c_t(a, e) d\mu_t$
 $\Rightarrow dC_t = C_t - C_{ss}$

How Do We Know SSJ?

- First, SSJs are infinite-dimensional objects, so truncate to $T \times T$ matrices
 - This is the only approximation involved in SSJ method
- How do we obtain M_r and M_Y ?
- The brute force approach:

$$\mathbf{r} := \begin{bmatrix} r \\ r + dr_1 \\ \vdots \\ r \end{bmatrix} \underbrace{\qquad\qquad\qquad}_{T \times 1 \text{ vector}}$$

Set $c_{T+1} = c_{ss}$
for $t = T : -1 : 1$
backward iteration of
the Euler:
 $u'(c_t) \geq \beta(1 + r_t) \mathbb{E} u'(c_{t+1})$
 $c_{t+1} \Rightarrow c_t$

Given $\mu_0 \equiv \mu_{ss}$ &
 $\{c_t, a_{t+1}\}_t$,
for $t = 1 : T$
forward iteration
of the law of
motion:
 $\mu_{t-1} \Rightarrow \mu_t$

Given $\{c_t, \mu_t\}$
 $\Rightarrow C_t = \int c_t(a, e) d\mu_t$
 $\Rightarrow dC_t = C_t - C_{ss}$

How Do We Know SSJ?

- First, SSJs are infinite-dimensional objects, so truncate to $T \times T$ matrices
 - This is the only approximation involved in SSJ method
- How do we obtain M_r and M_Y ?
- The brute force approach:

$$\mathbf{r} := \begin{bmatrix} r \\ r \\ \vdots \\ r + dr_T \end{bmatrix} \underbrace{\qquad\qquad\qquad}_{T \times 1 \text{ vector}}$$

Set $c_{T+1} = c_{ss}$
for $t = T : -1 : 1$
backward iteration of
the Euler:
 $u'(c_t) \geq \beta(1 + r_t)\mathbb{E}u'(c_{t+1})$
 $c_{t+1} \Rightarrow c_t$

Given $\mu_0 \equiv \mu_{ss}$ &
 $\{c_t, a_{t+1}\}_t$,
for $t = 1 : T$
forward iteration
of the law of
motion:
 $\mu_{t-1} \Rightarrow \mu_t$

Given $\{c_t, \mu_t\}$
 $\Rightarrow C_t = \int c_t(a, e) d\mu_t$
 $\Rightarrow dC_t = C_t - C_{ss}$

Speeding Up the SSJ Computation

- The brute force approach is prohibitively expensive
 - We need T backward iterations & T forward iterations
- Can we do better? Yes, a lot better!

Speeding Up the Backward Iteration

- The first key insight:

$$\frac{dc_t(a, e)}{dr_s} = \begin{cases} 0 & s < t \\ \frac{dc_{T-(s-t)}(a, e)}{dr_T} & s \geq t \end{cases}$$

- Bellman equation is entirely forward-looking! This encompasses:

1. Past shocks do not change the policy **functions**
 - It does change the consumption path, but that is through changes in asset a
2. Future shocks change the policies, but only the distance to the shock matters!
 - Calendar time is irrelevant

- This implies that a **single** backward iteration is enough to compute $\frac{dc_t(a, e)}{dr_s}$ for all t, s

Matrix Notation

- With discretized μ ($N_a N_y \times 1$ vector), the law of motion is

$$\mu_{t+1} = \Lambda'_t \mu_t \quad \text{where } \Lambda_t \text{ is the Markov transition matrix}$$

- To a first-order

$$\frac{d\mu_{t+1}}{dr_s} = \left[\frac{d\Lambda_t}{dr_s} \right]' \mu_{ss} + \Lambda_{ss} \frac{d\mu_t}{dr_s}$$

- Discretize the policy (c being $N_a N_y \times 1$ vector) to write the agg. consumption as

$$\mathcal{C}_t = c'_t \mu_t$$

- To a first-order,

$$M_{t,s}^r \equiv \frac{d\mathcal{C}_t}{dr_s} = \left[\frac{dc_t}{dr_s} \right]' \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} \right]$$

Matrix Notation

- With discretized μ ($N_a N_y \times 1$ vector), the law of motion is

$$\mu_{t+1} = \Lambda'_t \mu_t \quad \text{where } \Lambda_t \text{ is the Markov transition matrix}$$

- To a first-order

$$\frac{d\mu_{t+1}}{dr_s} = \left[\frac{d\Lambda_t}{dr_s} \right]' \mu_{ss} + \Lambda_{ss} \frac{d\mu_t}{dr_s}$$

- Discretize the policy (c being $N_a N_y \times 1$ vector) to write the agg. consumption as

Impact through changes in policy
holding dist. fixed $c'_t \mu_t$

- To a first-order,

$$M_{t,s}^r \equiv \frac{d\mathcal{C}_t}{dr_s} = \left[\frac{dc_t}{dr_s} \right]' \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} \right]$$

Matrix Notation

- With discretized μ ($N_a N_y \times 1$ vector), the law of motion is

$$\mu_{t+1} = \Lambda'_t \mu_t \quad \text{where } \Lambda_t \text{ is the Markov transition matrix}$$

- To a first-order

$$\frac{d\mu_{t+1}}{dr_s} = \left[\frac{d\Lambda_t}{dr_s} \right]' \mu_{ss} + \Lambda_{ss} \frac{d\mu_t}{dr_s}$$

- Discretize the policy (c being $N_a N_y \times 1$ vector) to write the agg. consumption as

$$\mathcal{C}_t = c'_t \mu_t$$

- To a first-order,

$$M_{t,s}^r \equiv \frac{d\mathcal{C}_t}{dr_s} = \left[\frac{dc_t}{dr_s} \right]' \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} \right]$$

Matrix Notation

- With discretized μ ($N_a N_y \times 1$ vector), the law of motion is

$$\mu_{t+1} = \Lambda'_t \mu_t \quad \text{where } \Lambda_t \text{ is the Markov transition matrix}$$

- To a first-order

$$\frac{d\mu_{t+1}}{dr_s} = \left[\frac{d\Lambda_t}{dr_s} \right]' \mu_{ss} + \Lambda_{ss} \frac{d\mu_t}{dr_s}$$

- Discretize the policy (c being $N_a N_y \times 1$ vector) to write the agg. consumption as

$$C_t = c'_t \mu_t$$

Impact through changes in dist.
holding policy fixed

- To a first-order,

$$M_{t,s}^r \equiv \frac{dC_t}{dr_s} = \left[\frac{dc_t}{dr_s} \right]' \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} \right]$$

Matrix Notation

- With discretized μ ($N_a N_y \times 1$ vector), the law of motion is

$$\mu_{t+1} = \Lambda'_t \mu_t \quad \text{where } \Lambda_t \text{ is the Markov transition matrix}$$

- To a first-order

$$\frac{d\mu_{t+1}}{dr_s} = \left[\frac{d\Lambda_t}{dr_s} \right]' \mu_{ss} + \Lambda_{ss} \frac{d\mu_t}{dr_s}$$

- Discretize the policy (c being $N_a N_y \times 1$ vector) to write the agg. consumption as

$$\mathcal{C}_t = c'_t \mu_t$$

- To a first-order,

$$M_{t,s}^r \equiv \frac{d\mathcal{C}_t}{dr_s} = \left[\frac{dc_t}{dr_s} \right]' \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} \right]$$

Recursive Structure of SSJ

- The SSJs have a particular recursive structure:

$$\begin{aligned} M_{t,s}^r - M_{t-1,s-1}^r &= \left[\frac{dc_t}{dr_s} - \frac{dc_{t-1}}{dr_{s-1}} \right] \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\Lambda_{t-1}}{dr_s} \mu_{ss} + \Lambda^{ss} \frac{d\mu_{t-1}}{dr_s} - \frac{d\Lambda_{t-2}}{dr_{s-1}} \mu^{ss} - \Lambda^{ss} \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} \Lambda_{ss} \left[\frac{d\mu_{t-1}}{dr_s} - \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} (\Lambda_{ss})^t \frac{d\mu_1}{dr_s} \equiv \mathcal{F}_{t,s} \end{aligned}$$

Recursive Structure of SSJ

- The SSJs have a particular recursive structure:

$$\begin{aligned} M_{t,s}^r - M_{t-1,s-1}^r &= \left[\frac{dc_t}{dr_s} - \frac{dc_{t-1}}{dr_{s-1}} \right] \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{a\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\Lambda_{t-1}}{dr_s} \mu_{ss} + \Lambda^{ss} \frac{d\mu_{t-1}}{dr_s} - \frac{d\Lambda_{t-2}}{dr_{s-1}} \mu^{ss} - \Lambda^{ss} \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} \Lambda_{ss} \left[\frac{d\mu_{t-1}}{dr_s} - \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} (\Lambda_{ss})^t \frac{d\mu_1}{dr_s} \equiv \mathcal{F}_{t,s} \end{aligned}$$

Recursive Structure of SSJ

- The SSJs have a particular recursive structure:

$$\begin{aligned} M_{t,s}^r - M_{t-1,s-1}^r &= \left[\frac{dc_t}{dr_s} - \frac{dc_{t-1}}{dr_{s-1}} \right] \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\Lambda_{t-1}}{dr_s} \mu_{ss} + \Lambda^{ss} \frac{d\mu_{t-1}}{dr_s} - \frac{d\Lambda_{t-2}}{dr_{s-1}} \mu^{ss} - \Lambda^{ss} \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} \Lambda_{ss} \left[\frac{d\mu_{t-1}}{dr_s} - \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} (\Lambda_{ss})^t \frac{d\mu_1}{dr_s} \equiv \mathcal{F}_{t,s} \end{aligned}$$

Recursive Structure of SSJ

- The SSJs have a particular recursive structure:

$$\begin{aligned} M_{t,s}^r - M_{t-1,s-1}^r &= \left[\frac{dc_t}{dr_s} - \frac{dc_{t-1}}{dr_{s-1}} \right] \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] = 0 \\ &= c_{ss} \left[\frac{d\Lambda_{t-1}}{dr_s} \mu_{ss} + \Lambda^{ss} \frac{d\mu_{t-1}}{dr_s} - \frac{d\Lambda_{t-2}}{dr_{s-1}} \mu^{ss} - \Lambda^{ss} \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c_{ss}' \Lambda_{ss} \left[\frac{d\mu_{t-1}}{dr_s} - \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} (\Lambda_{ss})^t \frac{d\mu_1}{dr_s} \equiv \mathcal{F}_{t,s} \end{aligned}$$

Recursive Structure of SSJ

- The SSJs have a particular recursive structure:

$$\begin{aligned} M_{t,s}^r - M_{t-1,s-1}^r &= \left[\frac{dc_t}{dr_s} - \frac{dc_{t-1}}{dr_{s-1}} \right] \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right] \\ &= c'_{ss} \left[\frac{d\Lambda_{t-1}}{dr_s} \mu_{ss} + \Lambda^{ss} \frac{d\mu_{t-1}}{dr_s} - \frac{d\Lambda_{t-2}}{dr_{s-1}} \mu^{ss} - \Lambda^{ss} \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} \Lambda_{ss} \left[\frac{d\mu_{t-1}}{dr_s} - \frac{d\mu_{t-2}}{dr_{s-1}} \right] \\ &= c'_{ss} (\Lambda_{ss})^t \frac{d\mu_1}{dr_s} \equiv \mathcal{F}_{t,s} \end{aligned}$$

Recursive Structure of SSJ

- The SSJs have a particular recursive structure:

$$M_{t,s}^r - M_{t-1,s-1}^r = \left[\frac{dc_t}{dr_s} - \frac{dc_{t-1}}{dr_{s-1}} \right] \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right]$$

The difference btwn :

1. Response at t to shock at s

$$\left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right]$$

2. Response at $t - 1$ to shock at $s - 1$

$$\left[\frac{d\Lambda_{t-1}}{dr_s} \mu_{ss} + \Lambda^{ss} \frac{d\mu_{t-1}}{dr_s} - \frac{d\Lambda_{t-2}}{dr_{s-1}} \mu_{ss} - \Lambda^{ss} \frac{d\mu_{t-2}}{dr_{s-1}} \right]$$

$$= c'_{ss} \Lambda_{ss} \left[\frac{d\mu_{t-1}}{dr_s} - \frac{d\mu_{t-2}}{dr_{s-1}} \right]$$

$$= c'_{ss} (\Lambda_{ss})^t \frac{d\mu_1}{dr_s} \equiv \mathcal{F}_{t,s}$$

Recursive Structure of SSJ

- The SSJs have a particular recursive structure:

$$M_{t,s}^r - M_{t-1,s-1}^r = \left[\frac{dc_t}{dr_s} - \frac{dc_{t-1}}{dr_{s-1}} \right] \mu_{ss} + c'_{ss} \left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right]$$

The difference btwn :

1. Response at t to shock at s

$$\left[\frac{d\mu_t}{dr_s} - \frac{d\mu_{t-1}}{dr_{s-1}} \right]$$

2. Response at $t - 1$ to shock at $s - 1$

$$\left[\frac{d\Lambda_{t-1}}{dr_s} \mu_{ss} + \Lambda^{ss} \frac{d\mu_{t-1}}{dr_s} - \frac{d\Lambda_{t-2}}{dr_{s-1}} \mu^{ss} - \Lambda^{ss} \frac{d\mu_{t-2}}{dr_{s-1}} \right]$$

$$\left[\begin{matrix} d\mu_{t-1} & d\mu_{t-2} \end{matrix} \right]$$

Households had one more period in advance to prepare for the shock

$$= c'_{ss} (\Lambda_{ss})^t \frac{d\mu_1}{dr_s} \equiv \mathcal{F}_{t,s}$$

Initial Condition for SSJ

- The initial conditions are

$$M_{0,s}^r = \left[\frac{dc_0}{dr_s} \right]' \mu_{ss}$$

- Now we know

1. $M_{t,s}^r - M_{t-1,s-1}^r \equiv \mathcal{F}_{t,s}$
2. $M_{0,s}^r$

⇒ We can compute any element of M^r !

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & M_{0,1}^r & \cdots & \cdots & \cdots & \cdots & M_{0,T}^r \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & M_{0,1}^r & \cdots & \cdots & \cdots & \cdots & M_{0,T}^r \\ \vdots & \vdots & & & & & \vdots \\ M_{1,1}^r = M_{0,0}^r + \mathcal{F}_{1,1} & & & & & & \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & & M_{0,1}^r & & \cdots & & \cdots & & M_{0,T}^r \\ & \ddots & & M_{1,1}^r & & & & & \vdots \\ & & \ddots & & & & & & \\ & & & \ddots & & & & & \\ & & & & \ddots & & & & \\ & & & & & \ddots & & & \\ & & & & & & \ddots & & \\ & & & & & & & \ddots & \\ & & & & & & & & \ddots \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & M_{0,1}^r & \cdots & \cdots & \cdots & \cdots & M_{0,T}^r \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{1,1}^r & M_{1,2}^r = M_{0,1}^r + \mathcal{F}_{1,2} & & & & & \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & & M_{0,1}^r & & \cdots & & \cdots & & \cdots & & M_{0,T}^r \\ & \ddots & & & & & & & & & \vdots \\ & & M_{1,1}^r & M_{1,2}^r & & & & & & & \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & & M_{0,1}^r & & \cdots & & \cdots & & \cdots & & M_{0,T}^r \\ M_{1,0}^r = \mathcal{F}_{1,0}^r M_{1,1}^r & M_{1,2}^r & & & & & & & & & \vdots \\ \vdots & \vdots \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & & M_{0,1}^r & & \cdots & & \cdots & & M_{0,T}^r \\ M_{1,0}^r & & M_{1,1}^r & M_{1,2}^r & & & & & \vdots \\ \vdots & & \vdots & & \ddots & & \ddots & & \vdots \\ \vdots & & \vdots & & \ddots & & \ddots & & \vdots \\ \vdots & & \vdots & & \ddots & & \ddots & & \vdots \\ \vdots & & \vdots & & \ddots & & \ddots & & \vdots \\ \vdots & & \vdots & & \ddots & & \ddots & & \vdots \\ \vdots & & \vdots & & \ddots & & \ddots & & \vdots \\ \vdots & & \vdots & & \ddots & & \ddots & & \vdots \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & & M_{0,1}^r & & \cdots & \cdots & \cdots & \cdots & M_{0,T}^r \\ M_{1,0}^r & & M_{1,1}^r & & \cdots & \cdots & \cdots & \cdots & M_{1,T}^r \\ \vdots & & \vdots & & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & \vdots & & & & & & \\ \vdots & & \vdots & & & & & & \\ \vdots & & \vdots & & & & & & \\ \vdots & & \vdots & & & & & & \\ \vdots & & \vdots & & & & & & \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & M_{0,1}^r & \cdots & \cdots & \cdots & M_{0,T}^r \\ M_{1,0}^r & M_{1,1}^r & \cdots & \cdots & \cdots & M_{1,T}^r \\ M_{2,0}^r & M_{2,1}^r & \cdots & \cdots & \cdots & M_{2,T}^r \\ \vdots & & \ddots & \ddots & \ddots & \\ \vdots & & & \ddots & \ddots & \\ \vdots & & & & \ddots & \end{bmatrix}$$

Filling the SSJ

$$M^r \equiv \begin{bmatrix} M_{0,0}^r & M_{0,1}^r & \cdots & \cdots & \cdots & \cdots & M_{0,T}^r \\ M_{1,0}^r & M_{1,1}^r & \cdots & \cdots & \cdots & \cdots & M_{1,T}^r \\ M_{2,0}^r & M_{2,1}^r & \cdots & \cdots & \cdots & \cdots & M_{2,T}^r \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & & \ddots & \ddots & \ddots & \\ \vdots & & & & \ddots & \ddots & \\ M_{T,0}^r & M_{T,1}^r & \cdots & \cdots & \cdots & \cdots & M_{T,T}^r \end{bmatrix}$$

Sequence Space Jacobian Algorithm

1. Solve Euler backward in response to a shock at the terminal period, dr_T

- This gives $\left\{ \frac{dc_t}{dr_s}, \frac{da_{t+1}}{dr_s} \right\}$ for any t, s because $\frac{dx_t}{dr_s} = \frac{dx_T}{dr_{T-(t-s)}}$

2. For each $s = 0, 1, \dots, T$

- Compute $M_{0,s}^r \equiv \left[\frac{dc_0}{dr_s} \right]' \boldsymbol{\mu}_{ss}$, $\frac{d\boldsymbol{\mu}_1}{dr_s} = \frac{d\boldsymbol{\Lambda}_0}{dr_s} \times \boldsymbol{\mu}_{ss}$, $\mathcal{F}_{t,s} = \boldsymbol{c}_{ss}' (\boldsymbol{\Lambda}_{ss})^t \frac{d\boldsymbol{\mu}_1}{dr_s}$
- For each $t = 0, 1, \dots, T$, compute $M_{t,s}^r$ recursively using

$$M_{t,s}^r = M_{t-1,s-1}^r + \mathcal{F}_{t,s}$$

⚠ For $t < 0$ or $s < 0$, set $M_{t,s}^r = 0$

The Sources of Speed Gains

Brute-force algorithm

1. Backward iteration from $t = T$ to $t = 0$ for each $s = 1, \dots, T$
2. Forward iteration from $t = 0$ to $t = T$ for each $s = 1, \dots, T$

ARBS algorithm (“fake-news algorithm”)

1. A **single** backward iteration from $t = T$ to $t = 0$
2. (Non-SS) forward iteration **only at** $t = 0$ for each $s = 1, \dots, T$

This cuts the computation time by a factor of T

- It takes ≈ 0.03 seconds to compute SSJ on my laptop

Tips

- You cannot do a grid search for Bellman iteration
 - The optimal grid never moves in response to a small shock
 - Use EGM, FOC, or continuous optimization
- Use automatic differentiation whenever you can!
 - This is easy in Julia
- I encounter numerical issues whenever T is big
 - The small error in the market clearing in the SS accumulates as $T \rightarrow \infty$
 - Need to separately take care of the error coming from the SS tolerance
- ARBS' Python package: <https://github.com/shade-econ/sequence-jacobian>
 - I'd discourage using it. Don't constrain what you do with what the package can do!

Scope and Frontiers

Scope of the Methodology

- So far, MIT shock. However, to a first order, certainty equivalence implies
IRF in a stochastic economy = IRF to an unanticipated shock
under the incomplete market w.r.t. aggregate shocks
- The methodology naturally extends to a much broader classes of models
 - Other shocks (e.g., ϕ shock & β shock)
 - Many other classes of heterogeneous agent models (e.g., heterogeneous firms)
- A key restriction is that individuals do not directly care about the distribution
 - This fails in many classes of job-ladder models
 - Still, the SSJ methodology extends as long as the job ladder is rank preserving
(Fukui, 2021; Fukui & Mukoyama, 2025)

SSJs are Useful Not Only for Computation

- SSJs are, in principle, directly measurable from the data
 - Sufficient statistics that strongly discipline the model
(e.g., Auclert-Rognlie-Straub, 2024, "Intertemporal Keynesian Cross")
- SSJs serve as the basis for the policy rule counterfactuals
(McKay and Wolf, 2023; Barnichon and Mesters, 2023)
- SSJs guide the optimal macro policies
(e.g., McKay and Wolf, 2025; Auclert, Cai, Rognlie, and Straub, 2024)

Frontier

1. Allow agents to hedge against the aggregate shocks
(Aculert, Rognlie, Straub, Ţapák, 2024; Donald, Fukui, Miyauchi, 2025)
2. 2nd- and higher-order approximation
("Beyond Certainty Equivalence: Second-Order Solutions in the Sequence Space" by ARS+)
3. Other approaches:
 - Machine learning to obtain global solutions:
Fernández-Villaverde, Hurtado, Nuno (2023), Payne, Rebei, Yang (2025)
 - State-space: Bhandari-Bourany-Golosov-Sargent (2023)
 - State-space + master equation: Bilal (2024)