Using the Master Equation to Solve HA Models in Discrete-Time

Based on Bilal (2024)

741 Macroeconomics

Topic 10

Masao Fukui

Fall 2025

State Space Approach with Perturbation

- State-space approach: keep track of the distribution as state variables
- A non-linear solution is infeasible or very expensive (Krusell and Smith, 1998)
- Perturbation approach (typically 1st-order, can go higher)

1. Reiter (2010)
    - Discretize all the equilibrium conditions, and then Taylor expand
    - Takes minutes
2. Bilal (2010)
    - Taylor expand first and then discretize
    - Takes less than a second

Hugget Model (from 704)

- Households solve

$$V_t(a,e) = \max_{c,\,a' \geq -\phi} u(c) + \beta \mathbb{E}_t V_{t+1}(a',e')$$

$$\text{s.t.} \quad c + q_t a' = a + eY$$

where $q \equiv 1/(1+r_t)$ is the inverse interest rate

  - No aggregate uncertainty for now

- Distribution evolves

$$\mu_{t+1}(a',e') = \text{Prob}(e' \mid e) \int_{a'(a,e) \in \mathscr{A}} \mu_t(a,e)da \equiv (\mathscr{T}_t^* \mu_t)(a',e')$$

- Bonds market clears (goods market clears by Warlas' law):

$$\int \int a_t'(a,e)\mu_t(a,e)dade = 0$$

Preliminaries

- Throughtout $x \equiv (a, e)$ and use them interchangeably
- Define the following transition operator

$$(\mathcal{T}_t v)(x) = \text{Pr}(e' \,|\, e) v(a_t'(x), e') = \mathbb{E}[v(x') \,|\, x]$$

  - In a discretized state space, $\mathcal{T}_t$ corresponds to the transition matrix
  - Expected value of $v(x')$ conditional on x today

- Define the adjoint operator of $\mathcal{T}_t$ as

$$(\mathcal{T}_t^* \mu)(x') = \int_{a: a' \in a_t'(a,e)} \text{Pr}(e' \,|\, e) \mu(a, e) da$$

  - In a discretized state space, $\mathcal{T}_t^*$ corresponds to the tranpose of $\mathcal{T}_t$
  - Collects all the origin x that flows into x'

Master Equation in Discrete Time

- Include all the state variables (a, e, μ) and drop t:

$$V(a, e, \mu) = \max_{c, a' \geq -\phi} u(c) + \beta \mathbb{E} V(a', e', \mu')$$

$$\text{s.t.} \quad c + q(\mu)a' = a + eY$$

$$\mu' = \mathcal{T}^*(\,\cdot\,, \mu)[\mu]$$

which is called the master equation. Note μ is infinite-dimensional.

- The pricing function $q(\mu)$ is implicitly defined as

$$\int \hat{a}'(x, q(\mu), \mu) d\mu = 0$$

where $\hat{a}'(x, q, \mu) \equiv \arg\max_{a'} \{u(a + eY - qa') + \beta \mathbb{E} V(a', e', \mathcal{T}^* \mu)\}$

Step 1: Approximate

FAME

- First-order approximation of $V(a, e, \mu)$ around the deterministic steady state:

$$V(x, \mu^{ss} + h) \approx V^{ss}(x) + \int v(x, \xi)h(\xi)d\xi$$

where $x \equiv (a, e)$ and $v(x, \xi)$ captures the extra value of adding households with ξ

- The change in the value of HH from adding HH itself solves the Bellman equation:

$$v(x, \xi) = u'(c^{ss}(x))D(x, \xi) + \beta(\mathcal{T}_x^{ss}\mathcal{T}_\xi^{ss}v)(x, \xi) + \beta\mathcal{T}_x^{ss}\int v(x, z)G(z, \xi)dz$$

where

$$D(x, \xi) \equiv -\frac{\partial q}{\partial \mu(\xi)}a'^{ss}(x)$$

$$G(z, \xi) = \partial_{\tilde{\mu}_\xi}(\mathcal{T}^*(\cdot, \tilde{\mu})[\mu])(z)$$

$$= \partial_{a'}(\mathcal{T}^*(\cdot, \tilde{\mu})[\mu])(z)\frac{\partial a'}{\partial \mu(\xi)}$$

- Note $v(x, \xi)$ is 4 $(= 2 \times 2)$ dimensional! Big dimensional reduction

Unpacking Derivatives

- By the implicit function theorem,

$$\frac{\partial q(\mu)}{\partial \mu(\xi)} = \frac{a^{'ss}(\xi) + \int \frac{\partial \hat{a}'(z, q^{ss}, \mu^{ss})}{\partial \mu(\xi)} \mu^{ss}(z) dz}{\int \frac{\partial \hat{a}'(z, q^{ss}, \mu^{ss})}{\partial q} \mu^{ss}(x) dx}$$

- For unconstrained HHs with $qu'(c) = \beta \mathbb{E} V(z', \mu')$, the saving responses are

$$\frac{\partial \hat{a}'(z, q^{ss}, \mu^{ss})}{\partial q} = \frac{u'(c^{ss}(z)) - qu''(c^{ss}(z)) a^{'ss}(z)}{(q^{ss})^2 u''(c^{ss}(z)) + \beta \partial_{a'a'} \mathbb{E} V(z')}$$

$$\frac{\partial \hat{a}'(z, q^{ss}, \mu^{ss})}{\partial \mu(\xi)} = \frac{\frac{\beta \mathbb{E} V(z', \mu')}{\partial \mu(\xi)}}{(q^{ss})^2 u'(c(z)) + \beta \partial_{a'a'} \mathbb{E} V(z')}$$

- For constrained HHs, $\frac{\partial \hat{a}'}{\partial q} = 0$ and $\frac{\partial \hat{a}'}{\partial \mu(\xi)} = 0$

Step 2: Discretize and Solve

Big Picture Algorithm

- The goal is to solve the following Bellman equation:

$$v(x,\xi) = u'(c^{ss}(x))D(x,\xi) + \beta(\mathcal{T}_x^{ss}\mathcal{T}_\xi^{ss}v)(x,\xi) + \beta\mathcal{T}_x^{ss}\int v(x,z)G(z,\xi)dz$$

- Both D and G depend on v, so this is a non-linear system in v
- We can solve using a simple iterative procedure

1. Guess v^i ($n \times n$ matrix, where $n = n_a \times n_e$)
2. Construct D^i and G^i based on v^i (both D and G are $n \times n$ matrices)
3. Update:

$$v^{i+1} = u'(c) \odot D^i + \beta T \cdot v^i \cdot T' + \beta T \cdot v^i \cdot G^i$$

4. Repeat until $|v^{i+1} - v^i| < tol$

Preliminaries

- Numerically compute $\frac{\partial a'}{\partial q}$ to obtain

$$denom_i \equiv -qu''(c) - \partial_{aa}\mathbb{E}V = -\frac{u'(c) - qu''(c)a'}{\partial a'/\partial q}.$$

- We do this because numerical 2nd derivative can be inaccurate

```julia
p_local = PARAMS[]
agrid = ss.agrid
na, ny = p_local.na, p_local.ny
n = na * ny
β, γ, q = p_local.β, p_local.γ, ss.q

cflat  = vec(ss.c)
apflat = vec(ss.ap)
g      = ss.g
T      = ss.T

# --- policy sensitivity to the CURRENT price, continuation fixed: dap_dq = ∂a'/∂q
EWa_ss = continuation_EWa(ss.c, ss.Π, p_local)
dq = 1e-5
_, ap_p = egm_policy(EWa_ss, q + dq, p_local, agrid)
_, ap_m = egm_policy(EWa_ss, q - dq, p_local, agrid)
dap_dq = vec((ap_p .- ap_m) ./ (2dq))                  # dap_dq_i < 0 (demand slopes down)

# constrained households: policy pinned at the borrowing limit
constrained = vec(ss.ap) .<= (agrid[1] + 1e-8)
dap_dq[constrained] .= 0.0

# curvature denominator  denom_i = -(u'(c)-q u''(c)a') / dap_dq_i  (>0, unconstrained)
uc  = cflat .^ (-γ)
ucc = -γ .* cflat .^ (-γ - 1)
num = uc .- q .* ucc .* apflat                     # = u'(c) - q u''(c) a'
denom = fill(Inf, n)
@inbounds for i in 1:n
    if !constrained[i] && dap_dq[i] != 0.0
        denom[i] = -num[i] / dap_dq[i]
    end
end
```

Preliminaries

```
    M  = make_M(ss.ap, g, ss.Π, p_local, agrid)
    bD = -(uc .* apflat)                              # b_i = -u'(c_i) a'_i   (for Dtilde)
    g_dap_dq = dot(g, dap_dq)                                   # aggregate demand slope (<0)

    v   = zeros(n, n)
    G  = zeros(n, n)
    Qp  = -(apflat) ./ g_dap_dq                            # initial price impact (B_future=0)
```

- M corresponds to $\partial_{a'}(\mathcal{T}^*(\cdot, \tilde{\mu})[\mu])$ (defined below)
- bD is preliminary for computing $u'(c) \odot D$
- g_dap_dq is $\int \frac{\partial \hat{a}'}{\partial q} d\mu^{ss}$
- v and G are all initial guesses for v and G; Qp is the initial guess for $dq/d\mu$

```
function make_M(ap, g, Π, p::Huggett, agrid)
    na, ny = p.na, p.ny
    n = na * ny
    M = zeros(n, n)
    for iy in 1:ny, ia in 1:na
        i = idx(ia, iy, p)
        k, _ = bracket(agrid, ap[ia, iy])
        Δ = agrid[k+1] - agrid[k]
        gi = g[i]
        for jy in 1:ny
            pr = Π[iy, jy] * gi / Δ
            M[idx(k,   jy, p), i] -= pr
            M[idx(k+1, jy, p), i] += pr
        end
    end
    return M
end
```

Iteration

```
for it in 1:maxit
    va = dv_da(v, p_local, agrid)

    # response of policy to the distribution through FUTURE prices, q fixed
    Bfut = β .* (T * va * (Matrix(T') + G))
    @inbounds for i in 1:n
        if isfinite(denom[i])
            @views Bfut[i, :] ./= denom[i]
        else
            @views Bfut[i, :] .= 0.0
        end
    end

    # price impact of mass at ξ:  Q'_j = −(a'_j + Σ_i g_i Bfut[i,j]) / (g·dap_dq)
    Qp = −(apflat .+ (Bfut' * g)) ./ g_dap_dq

    # total policy response to the impulse and the GE kernel
    dadg = dap_dq * Qp' .+ Bfut                         # da'_i/dg_j  (n×n)
    G    = M * dadg

    # direct price impact (rank one) and the FAME update
    Dtil = bD * Qp'                                 # Dtilde[i,j] = −u'(c_i)a'_i Q'_j
    vnew = Dtil .+ β .* (T * v * (Matrix(T') + G))

    err = maximum(abs.(vnew .− v))
    v = vnew
    if verbose && (it % 25 == 0 || it == 1)
        @printf("  FAME iter %4d   ‖Δv‖=%.3e\n", it, err)
    end
    err < tol && break
end
```

FAME vs. SSJ

Experiment

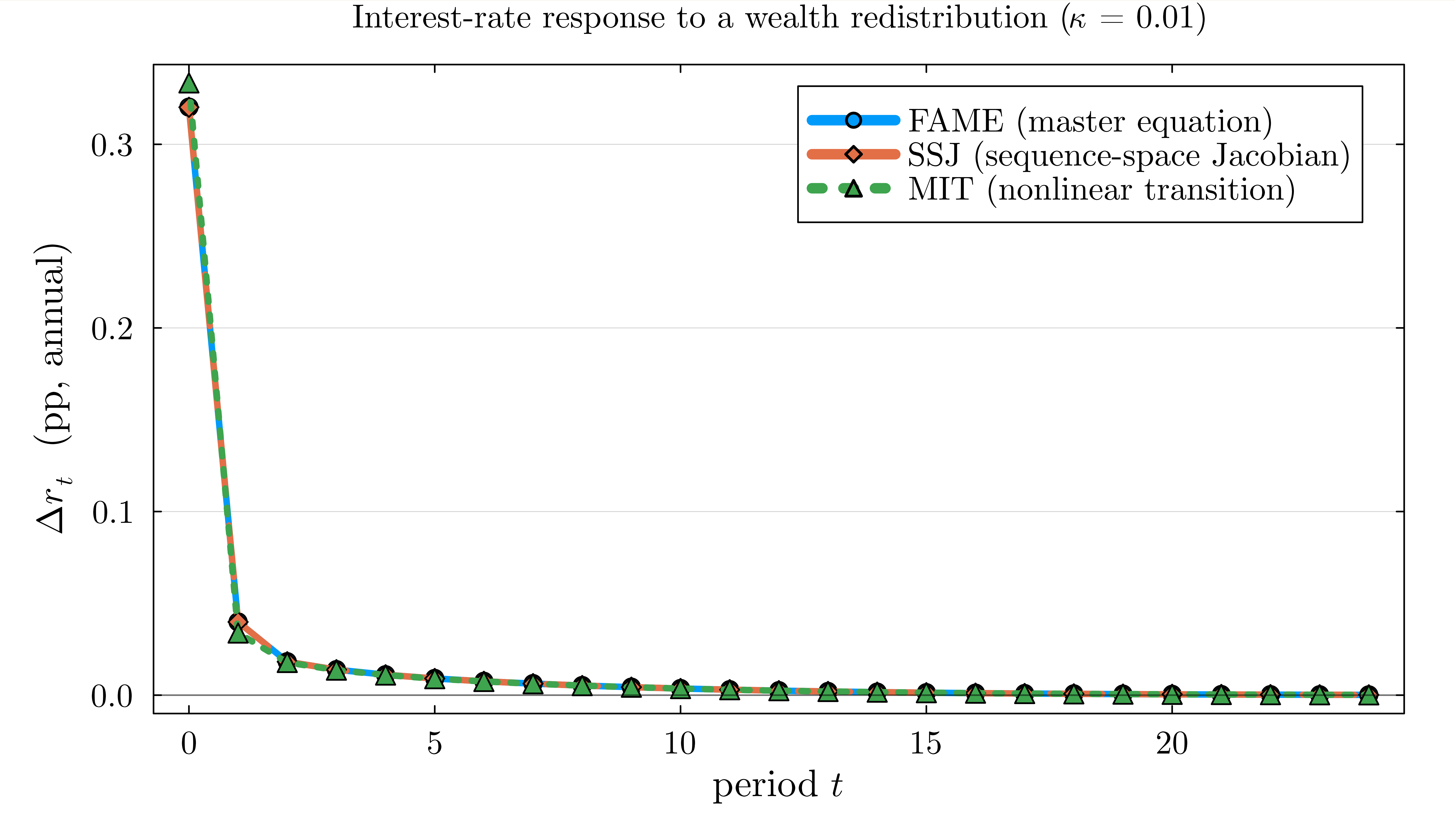
- Consider a redistribution shock
- At time 0, the government redistributes wealth so that

$$a' = a(1 - \kappa)$$

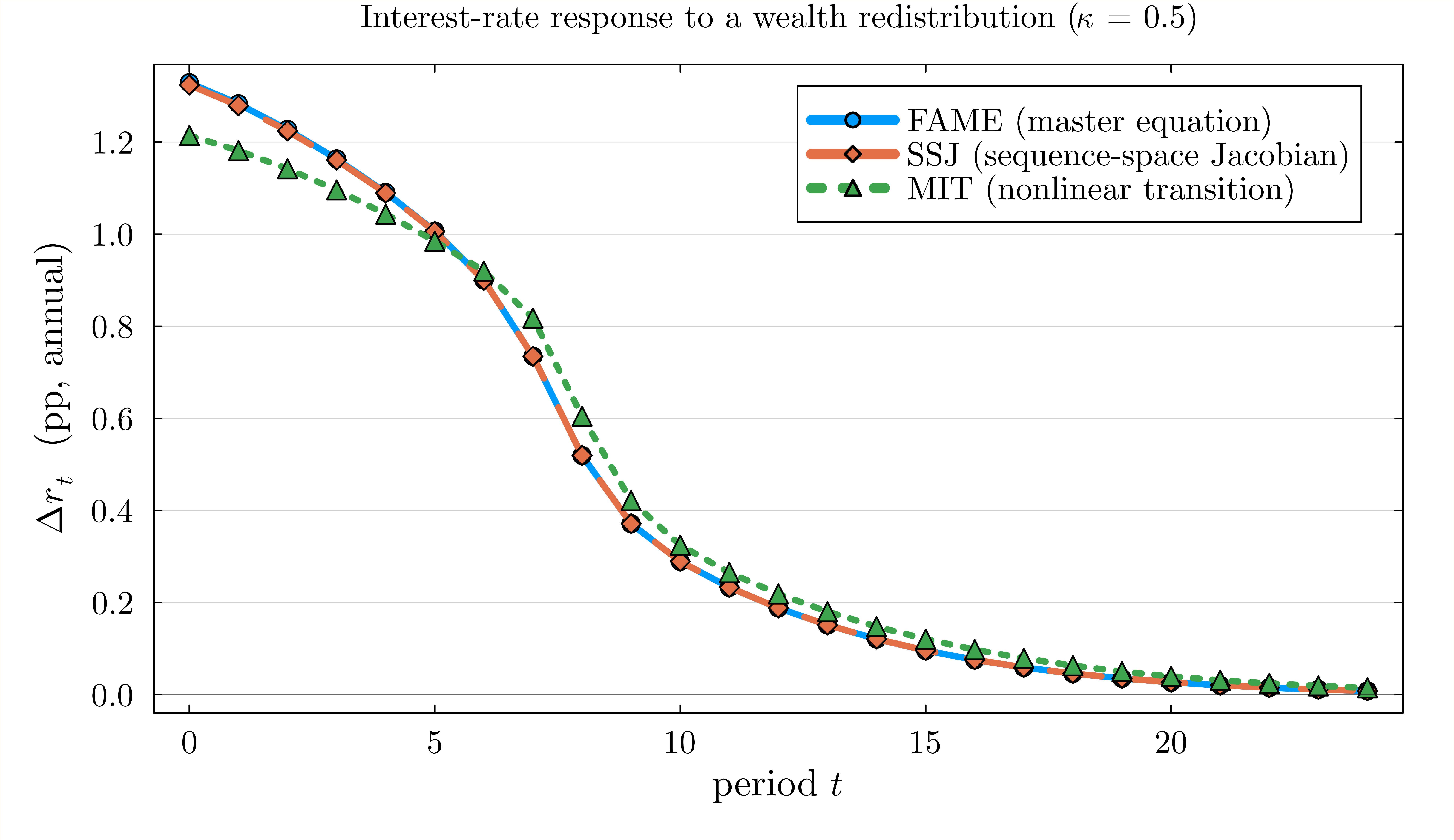
  where $\kappa > 0$ is the degree of redistribution

- Compare three approaches:
  1. FAME
  2. SSJ
  3. Non-linear transition (iterate the sequence of $\{q_t\}$)

Impulse Response to Small Shock



Impulse Response to Large Shock



Performance Comparison

Huggett interest-rate IRF: solver performance

method	min (ms)	median (ms)	memory (MiB)	allocs	speedup
FAME (master equation)	346.0	350.3	1390	7,659	78×
SSJ (sequence-space J)	14.7	15.9	20	24,028	1712×
MIT (nonlinear trans.)	27244.1	27244.1	57243	46,117,996	1×

- SSJ is a clear winner in this particular case

Spatial Huggett Model

Spatial Huggett Model

- In Huggett model, there is only one price in each period, q_t
- This might explain the huge advantage of SSJ over FAME
- What if there are many prices in each period?
- We will consider a spatial extension of Huggett model
  - Households consume and save with bonds, as in Huggett
  - Dynamic migration and trade across J regions, as in Caliendo-Dvorkin-Parro (2019)
  - Based on Donald-Fukui-Miyauchi (2025)

Huggett + CDP

- Households solve

$$V_{it}(a) = \max_{c,\, a' \geq -\phi} u(c) + \beta \left[\sum_j \mu_{ij} V_{jt+1}(a') - \psi_i(\{\mu_{ij}\}_j) \right]$$

$$\text{s.t.} \quad P_{it} c + q_t a' = a + w_{it}$$

  - μ_{ij} is the migration probability from i to j
  - ψ_{it} is the migration cost function; can be micro-founded with discrete choice

- Firms solve

$$\max_{l_{ij}} P_{jt} f_{jt}(\{l_{ij}\}_i) - \sum_i w_{it} l_{ij}$$

- Markets clear

$$\sum_j l_{ijt} = \int \mu_{it}(a) da, \quad f_{jt}(\{l_{ijt}\}_i) = \int c_{it}(a) \mu_{it}(a) da$$

Parameteriztaon

- Assume CES (gravity) in trade:

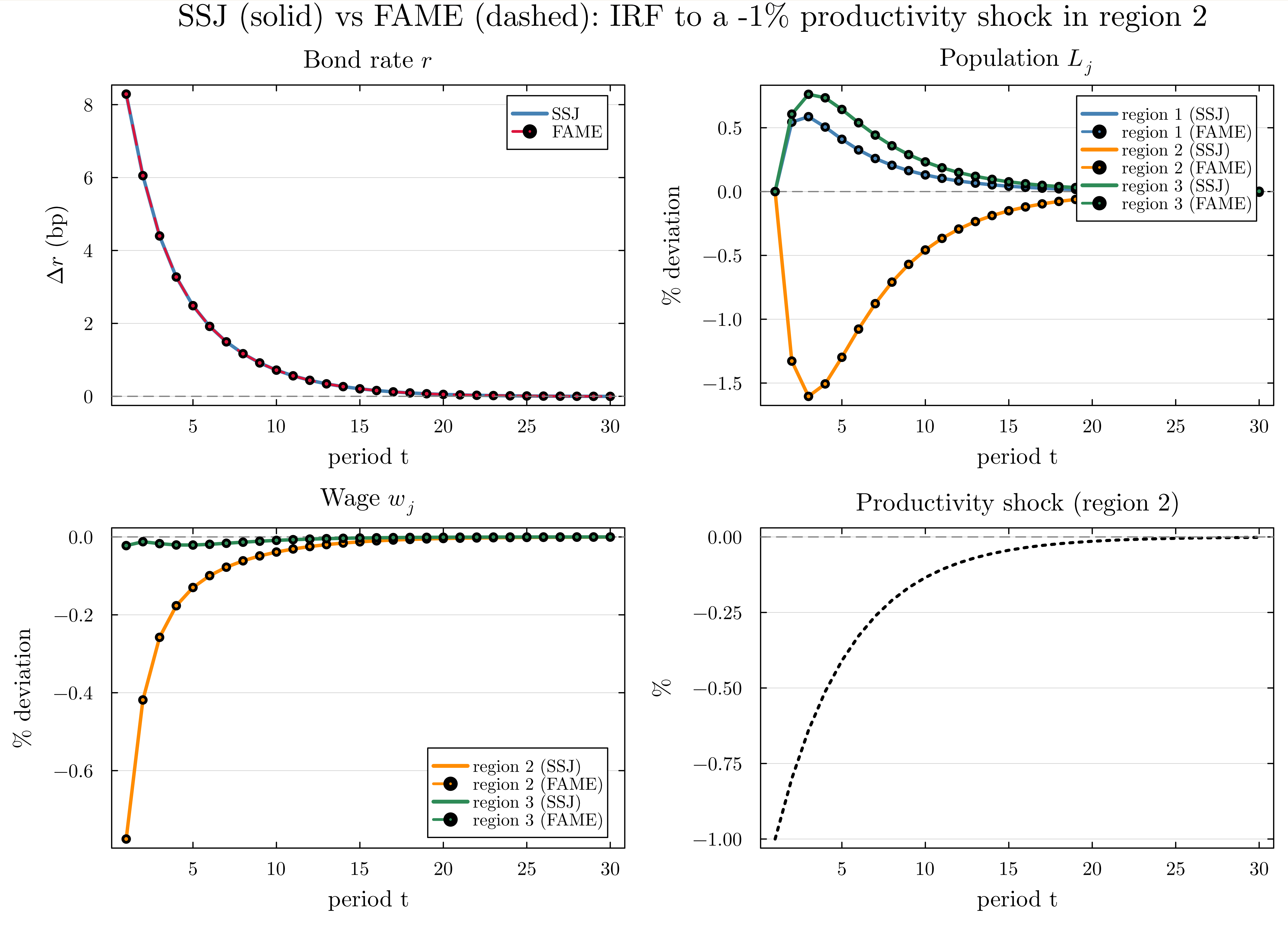
$$f_{jt}(\{l_{ij}\}_i) = \left(\sum_i (A_{ijt} l_{ijt})^{\frac{\sigma-1}{\sigma}} \right)^{\frac{\sigma}{\sigma-1}}$$

- Assume a logit migration system:

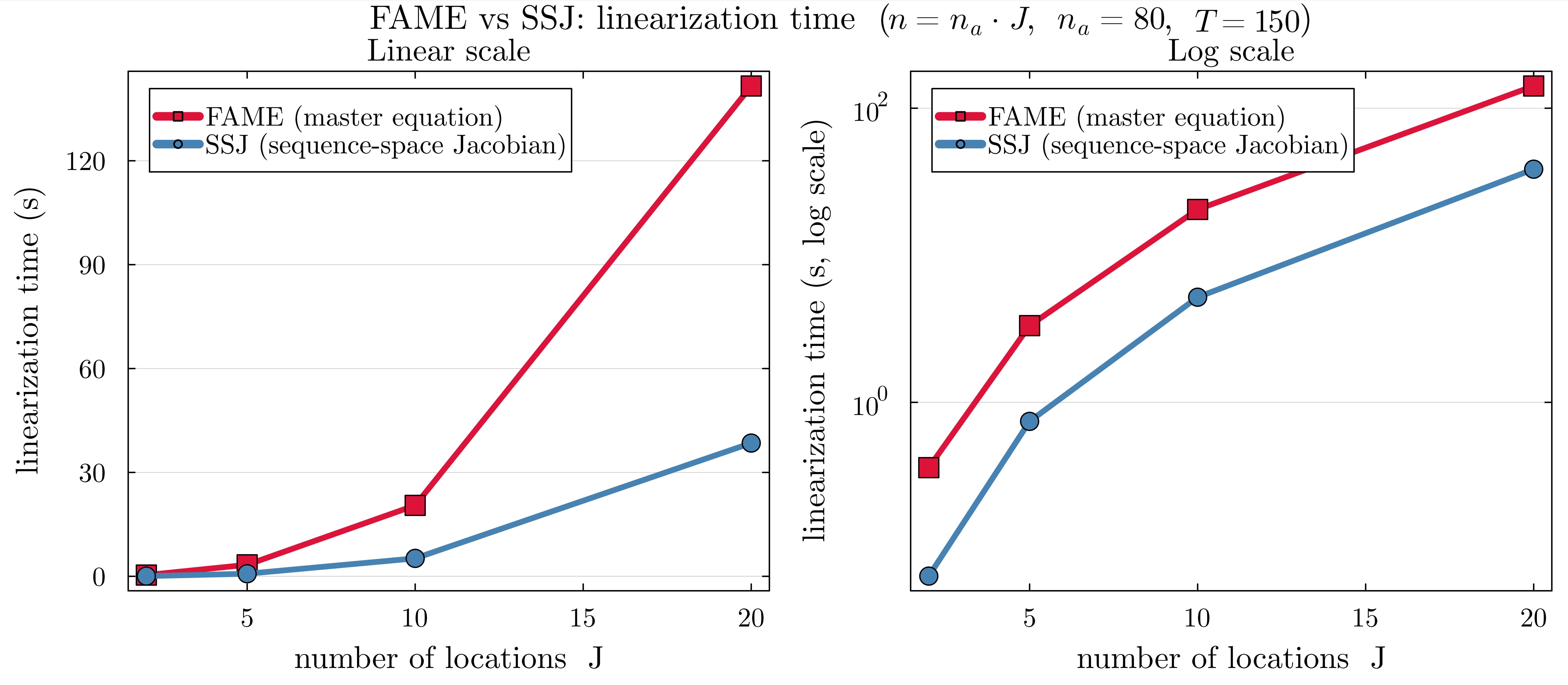
$$\psi_{it}(\{\mu_{ij}\}_j) = \frac{1}{\theta} \sum_j \mu_{ij} \ln \mu_{ij}$$

- Consider a productivity shock to location 2
- Wage of location 1 is numeraire

Impulse Response



Performance Comparison



- SSJ keeps winning with no indication of striking the advantage as we increase J

Open Question

- When does FAME have an advantage over SSJ?
- Many prices within a location?
- Search model?