HANK in Continuous Time

Adrien Bilal

Stanford University

June 2025

Introduction

- So far in workshop saw how to handle workhorse HANK model in discrete time
  - Though paper coining 'HANK' developed in continuous time! (Kaplan et al. 2018)
- Continuous time has several attractive properties for heterogeneous agent models
  - Efficient calculation of steady-state with finite differences (Achdou et al. 2022)
  - Seamlessly handles nonconvexities (Kaplan et al. 2018): kinks in interest rates, fixed portfolio costs
  - Sometimes can gain more pen-and-paper traction
- **What does a sequence-space approach to HANK in continuous time look like?**
  - Retain fast transitions and low memory requirements?
  - Additional benefits in continuous time?

This lecture

Based on Bilal and Goyal "Some Pleasant Sequence-Space Arithmetic in Continuous Time" (2025)

- Set up standard HANK economy in continuous time
  - Individual decision problem: backward PDE
  - Law of motion: forward PDE
- Develop sequence space Jacobians in continuous time
  - Main payoff of continuous time: characterize policy functions analytically w/ backward PDE
  - Rest of SSJ construction as in Auclert et al. (2021)
  - Approach applies to much more general settings
- Discuss implementation and discretization (time permitting)
- Leverage analytical characterization: 3x speed gain

Literature

- Perturbation methods for heterogeneous agent macro in discrete time: Reiter (2008), Ahn et al. (2018), **Auclert et al. (2021)**, Bhandari et al. (2024)
  - Develop and characterize SSJ in continuous time
- Perturbation methods for heterogeneous macro in continuous time: Bilal (2023), **Glawion (2023)**
  - Sequence space rather than state space (Bilal 2023)
  - Does not rely on discretizing state/time first & derive PDE for policy functions (Glawion 2023)
  - If interested in **second-order perturbations**: Bilal (2023)

Setup

Household decision problem

- Consider continuous-time version of setup seen so far
- Budget constraint: $da_t = (r_t a_t + X_t y_t^{1-\theta} - c_t)dt$, where
  - a_t denotes assets, c_t consumption,
  - post-tax income satisfies $X_t y_t^{1-\theta} = \tau_t (w_t y_t n_t)^{1-\theta}$, $\int y_{jt}^{1-\theta} dj \equiv 1$, $X_t = Y_t - T_t$
- Borrowing constraint: $a_t \geq \underline{a}$
- For exposition use income process with discrete states $\{y_1, ..., y_J\}$ and transition rates λ_{ij}
- Hamilton-Jacobi-Bellman (HJB) equation:

$$\rho V_{it}(a) = \max_{c \geq 0} u(c) - v(n_{it}(a)) + \overbrace{\boldsymbol{L_{it}(a,c)}[V_t]}^{\substack{\text{continuation value} \\ \text{from changes in} \\ \text{own assets and prod.}}} + \overbrace{\mathbb{E}_t \left[\frac{dV_{it}(a)}{dt}\right]}^{\substack{\text{continuation value} \\ \text{from changes in} \\ \text{prices } r_t, X_t}}$$

$$\boldsymbol{L_{it}(a,c)}[V] \equiv \underbrace{\underbrace{\left(r_t a + X_t y_i^{1-\theta} - c\right)}_{\text{savings rate}} \frac{\partial V_i(a)}{\partial a}}_{\substack{\text{continuation value} \\ \text{from changes in assets}}} + \underbrace{\sum_j \lambda_{ij}(V_j(a) - V_i(a))}_{\substack{\text{continuation value from} \\ \text{changes in productivity}}}$$

Occasionally binding borrowing constraints

- Handling borrowing constraints is very tractable in continuous time
  - In continuous time $a_{t+dt} = a_t + S_{it}(a_t)dt$ is close to a_t
  - So only households exactly at $a_t = \underline{a}$ behave behave as constrained
  - Different from discrete time where households in some interval $[\underline{a}, \tilde{a})$ behave as constrained
- Tractably incorporate borrowing constraint as restriction on value function
  - Turns out that **consumption FOC always holds with equality**, even at $\underline{a}$:

$$u'(c_{it}(a)) = \frac{\partial V_{it}(a)}{\partial a}, \quad \text{for all } a \geq \underline{a}$$

  - Because value function adjusts so that assets do not cross $\underline{a}$: $c_{it}(\underline{a}) \leq r_t \underline{a} + X_t y_i^{1-\theta}$, that is:

$$\frac{\partial V_{it}(\underline{a})}{\partial a} \geq u'\left(r_t \underline{a} + X_t y_i^{1-\theta}\right)$$

Evolution of distribution

- Law of motion (Kolmogorov Forward) equation:

$$\frac{dg_{it}(a)}{dt} = \underbrace{-\frac{\partial}{\partial a}\Big(S_{it}(a)g_{it}(a)\Big)}_{\text{savings}} + \underbrace{\sum_k g_{kt}(a)\lambda_{ki}}_{\substack{\text{inflow from} \\ \text{other prod. states}}} - \underbrace{g_{it}(a)\sum_j \lambda_{ij}}_{\substack{\text{outflow to} \\ \text{other prod. states}}} \equiv \boldsymbol{L^*_{it}(a, c_{it}(a))}[g_t]$$

  - $S_{it}(a) = r_t a + y_i - c_{it}(a)$: savings rate; $c_{it}(a)$: consumption
  - $L^*_i(a)[\cdot]$ denotes the adjoint (transpose) of functional operator $L_i(a)[\cdot]$
  - As such holds for $a > \underline{a}$
- To extend this formula to $a = \underline{a}$ need to introduce a **base measure** $d\nu(a)$
  - Let $g_{it}(a)$ be the density of households **with respect to base measure** $d\nu(a) = \delta_{\underline{a}}(a) \oplus da$
  - $\delta_{\underline{a}}(a)$: Dirac mass point. $g_{it}(\underline{a})$: measure of households at constraint
  - $g_{it}(a)da$: measure of households in $[a, a + da)$
  - Base measure important so that law of motion in density well-posed at constraint
  - Derivatives become 'weak', not 'classic', to be compatible with base measure and constraint
  - In practice can largely ignore this subtlety for numerical discretization

Firms, unions, government and the monetary authority

- Rest of model is exactly as seen so far
- **Firms** produce under flexible prices $Y_t = Z_t N_t$ so real wages are $w_t = W_t / P_t = Z_t$
- **Monetary policy:** $i_t = r^{ss} + \phi \pi_t + \varepsilon_t$
- **Fiscal policy** sets taxes T_t, spending G_t and debt to satisfy BC: $\frac{dB_t}{dt} = r_t B_t + G_t - T_t$
- **Unions** set nominal wages, leading to wage Phillips curve:

$$\rho \pi_t^W = \kappa_1 u'(C_t) X_t + \kappa_2 v'(N_t) N_t + \mathbb{E}_t \left[\frac{d\pi_t^W}{dt} \right]$$

  - $\kappa_1 = (1-\varepsilon)(1-\theta)/\psi$, $\kappa_2 = \varepsilon/\psi$, $\pi_t^W = \frac{dW_t/dt}{W_t}$
  - ε: ES in labor aggregation across unions; ϕ: shifter of quadratic union wage adjustment costs
- **Asset market clearing:** $\sum_i \int a g_{it}(a) d\nu(a) = B_t$

Steady-state

Zero-inflation steady-state

- **HJB:**

$$\rho V_i^{ss}(a) = \max_{c\geq 0} u(c) - v(n_i^{ss}(a)) + \underbrace{\left(r^{ss}a + X^{ss}y_i^{1-\theta} - c\right)\frac{\partial V_i^{ss}(a)}{\partial a} + \sum_j \lambda_{ij}(V_j^{ss}(a) - V_i^{ss}(a))}_{\equiv \boldsymbol{\mathcal{L}(a,c)}[V^{ss}]}$$

$$\frac{\partial V_i^{ss}(\underline{a})}{\partial a} \geq u'\left(r^{ss}\underline{a} + X^{ss}y_i^{1-\theta}\right)$$

- **KFE:**

$$0 = -\frac{\partial}{\partial a}\left(S_i^{ss}(a)g_i^{ss}(a)\right) + \sum_k g_k^{ss}(a)\lambda_{ki} - g_i^{ss}(a)\sum_j \lambda_{ij} \equiv \boldsymbol{\mathcal{L}_i^*(a, c_i^{ss}(a))}[g^{ss}]$$

- **Market clearing:** $B^{ss} = \sum_i \int a g_i^{ss}(a) d\nu(a)$, together with fiscal equations

Sequence-space Jacobians

Steady-state and perturbations

- We start from a deterministic steady-state with time-invariant
  - Value function $V_i^{ss}(a)$ and consumption function $c_i^{ss}(a)$
  - Distribution $g_i^{ss}(a)$
  - **Transition operator $\mathcal{L}_i(a)$** $\equiv L_i^{ss}(a, c_i^{ss}(a))$: corresponds to **transition matrix** in discrete time
- As with all perturbation methods we consider the limit when shocks are small
- Denote deviations from steady-state (ss) with hats

Policy functions

Individual policy functions

- Work with **marginal utility of consumption** $\boldsymbol{\omega_{it}(a) = u'(c_{it}(a)) = \frac{\partial V_{it}(a)}{\partial a}}$
  - Could also work directly with value (code uses value)

- Posit to first order

$$\hat{\omega}_{it}(a) = \omega_{it}(a) - \omega_i^{ss}(a) = \mathbb{E}_t \int_0^\infty e^{-\rho s} \Big\{ \boldsymbol{\varphi^r_{is}(a) \hat{r}_{t+s}} + \boldsymbol{\varphi^X_{is}(a) \hat{X}_{t+s}} \Big\} ds \qquad (\star)$$

- φ^r, φ^X: how marginal utility responds to a future sequence of interest rates, income and hours
  - No effect of $\hat{N}$ because union-decided hours worked are uniform and labor disutility is additive
  - Drops out of the Euler equation / HJB for $u'(c)$, but easy to add back in if change model assumptions

- In discrete time numerically differentiate nonlinear Euler equation

- In continuous time instead substitute $(\star)$ into HJB and identify coefficients
  - HJB and $(\star)$ must hold for all sequences $\mathbb{E}_t \hat{r}_{r+s}, \mathbb{E}_t \hat{X}_{t+s}$
  - Imposes restrictions on φ^r, φ^X
  - Allows to **do more analytically and ease computational burden**

Individual policy functions, continued

- Obtain PDEs for φ^r, φ^X

$$\underbrace{\varphi^r_{i0}(a)}_{\text{initial condition}} = \underbrace{\frac{\partial}{\partial a}\Big(a\ u'(c_i^{ss}(a))\Big)}_{\substack{\text{marginal value of}\\ \text{disposable income gain}\\ \text{from price change}}}, \qquad \frac{\partial \varphi^r_{it}(a)}{\partial t} = \underbrace{\left(r^{ss} - \frac{\partial c_i^{ss}(a)}{\partial a}\right)}_{\text{substitution effect}} \varphi^r_{it}(a) + \underbrace{\mathcal{L}_i(a)[\varphi^r_t]}_{\substack{\text{backward time propagation}\\ \text{through expectations}}}$$

$$\overbrace{\varphi^X_{i0}(a)} = \overbrace{\frac{\partial}{\partial a}\Big(y^{1-\theta}\ u'(c_i^{ss}(a))\Big)}, \qquad \frac{\partial \varphi^X_{it}(a)}{\partial t} = \overbrace{\left(r^{ss} - \frac{\partial c_i^{ss}(a)}{\partial a}\right)} \varphi^X_{it}(a) + \overbrace{\mathcal{L}_i(a)[\varphi^X_{it}]}$$

  - Without risk and under CRRA: $c_0(a) = \left(r^{ss} + \sigma(\rho - r^{ss})\right)\left(a + \int_0^\infty e^{-r^{ss}t} y_t\right)$

- Similar to standard HJBs
  - No discounting because already accounted for in $(\star)$
  - Opposite sign of time derivative: changed direction of time when defining φ
  - Terminal condition at t becomes initial condition at 0

Individual policy functions with binding constraints

- Constraint occasionally binds $\implies$ add boundary condition
- **If constraint binds at $(\underline{a}, y_i)$ in steady-state:**

$$\varphi^{r}_{it}(\underline{a}) = u''(c^{ss}_i(\underline{a}))\underline{a}\ \delta_0(t) \qquad \varphi^{X}_{it}(\underline{a}) = u''(c^{ss}_i(\underline{a}))y_i^{1-\theta}\ \delta_0(t)$$

where $\delta_0(t)$ denotes a Dirac mass function: at constraint only react to current changes in prices

- After solving for φ^r, φ^X, obtain consumption from FOC:

$$\hat{c}_{it}(a) = \frac{\hat{\omega}_{it}(a)}{u''(c^{ss}_i(a))} = \frac{1}{u''(c^{ss}_i(a))}\mathbb{E}_t \int_0^\infty e^{-\rho s}\Big\{\varphi^{r}_{is}(a)\hat{r}_{t+s} + \varphi^{X}_{is}(a)\hat{X}_{t+s}\Big\}ds$$

Taking stock

- Have derived simple linear PDEs
  - Counterpart to Auclert et al. (2021) but here do not need to numerically differentiate
- Fully determine how individual policy functions respond to price changes
- Solution φ_s can be calculated with a single **linear** time iteration
- Explicitly shaped by steady-state objects
  - Disposable income effect: φ_0
  - Substitution effect: $r^{ss} - \partial_a c^{ss}$
  - Expectations: $\mathcal{L}$
  - Borrowing constraint: $\varphi(\underline{a})$
- Rest of construction of SSJ will mirror Auclert et al. (2021)

Distribution

Law of motion of the distribution given savings

- Now combine policy functions w/ evolution of distribution to construct market clearing conditions
- For small changes in savings rates:

$$\frac{d\hat{\boldsymbol{g}}_t}{dt} = \underbrace{\mathcal{L}^*[\hat{\boldsymbol{g}}_t]}_{\substack{\text{PE evolution} \\ \text{at steady-state decisions}}} - \underbrace{\frac{\partial}{\partial a}\left(\boldsymbol{g}^{ss}\hat{\boldsymbol{S}}_t\right)}_{\substack{\text{GE evolution} \\ \text{of savings rate}}}$$

- Solve for the evolution of distribution $\{\hat{g}_t\}_{t\geq 0}$

$$\hat{\boldsymbol{g}}_t = \boldsymbol{T}_t^*[\hat{\boldsymbol{g}}_0] - \int_0^t \boldsymbol{T}_{t-\tau}^* \left[\frac{\partial}{\partial a}\left(\boldsymbol{g}^{ss}\hat{\boldsymbol{S}}_\tau\right)\right] d\tau$$

where T_t^* is the continuous-time analogue of the transition matrix (semigroup) and satisfies

$$T_0^* = \text{identity operator} \qquad \frac{\partial T_t^*}{\partial t} = \mathcal{L}^*[T_t^*]$$

Law of motion of the distribution

- Substitute in solution for policy function into savings

$$\hat{g}_t = T_t^*[\hat{g}_0] + \int_0^\infty \left\{ \int_0^{\min\{t,\tau\}} \left(T_{t-s}^*[\mathcal{D}_{\tau-s}^r]\hat{r}_\tau + T_{t-s}^*[\mathcal{D}_{\tau-s}^X]\hat{X}_\tau \right) ds \right\} d\tau$$

- $\mathcal{D}_{\tau-s}^r, \mathcal{D}_{\tau-s}^X$: time-0 distribution changes to 'fake news'
  - Time-s announcement of time-τ price change, retracted at $s + ds$
  - Add up to $\min\{t, \tau\}$ since distribution evolution is backward-looking
- For interest rate

$$\mathcal{D}_{is}^r(a) = -\frac{\partial}{\partial a}\left(g_i^{ss}(a) \left(\underbrace{a\ \delta_0(s)}_{\substack{\text{direct effect}\\ \text{when shock}\\ \text{hits at 0}}} - \underbrace{\frac{e^{-\rho s}\varphi_{is}^r(a)}{u''(c^{ss})}}_{\text{cons. change}} \right) \right)$$

- Similar for aggregate income X
- Once more get **analytic expression** rather than numerical derivative

Market clearing

Market clearing

- To clear markets use: $\hat{B}_t = \sum_i \int a\boldsymbol{\hat{g}_{it}(a)}d\nu(a)$
- For any function $\mathcal{E}_i(a)$, denote the expectation of $\mathcal{E}$ under g_t by: $\mathcal{E}^* g_t \equiv \sum_i \int \mathcal{E}_i(a) g_{it}(a) d\nu(a)$
- Could compute T_t^* and solve for full path of distribution $\boldsymbol{\hat{g}_{it}}$
- As in discrete time would not be computationally efficient
  - Solving N $N \times T$-dimensional PDEs if T time & N asset/income gridpoints
- Instead use distribution solution to construct market clearing conditions first
- Then recognize that need only to compute lower-dimensional objects to solve for equilibrium
  - Expectation function
  - Fake news operator

Fake news matrix and Jacobians

- Integrate solution for distribution against $\mathcal{E}^*$:

$$\mathcal{E}^* \hat{g}_t = \mathcal{E}_t^*[\hat{g}_0] + \int_0^\infty \left\{ \int_0^{\min\{t,\tau\}} \left(\mathcal{E}_{t-s}^* \mathcal{D}_{\tau-s}^r \hat{r}_\tau + \mathcal{E}_{t-s}^* \mathcal{D}_{\tau-s}^X \hat{X}_\tau \right) ds \right\} d\tau$$

where **the expectation function** $\mathcal{E}^* T_t^* = (T_t \mathcal{E})^* \equiv \mathcal{E}_t^*$ satisfies simple PDE:

$$\mathcal{E}_0 = \mathcal{E} \qquad \frac{\partial \mathcal{E}_t}{\partial t} = \mathcal{L}[\mathcal{E}_t]$$

  - Compute as a single PDE given $\mathcal{E}$
  - One expectation function per market clearing condition

- Define the **fake news operator** and the **Jacobians** similarly to discrete time:

$$\mathcal{F}_{t-s,\tau-s}^p = \mathcal{E}_{t-s}^* \mathcal{D}_{\tau-s}^p \qquad \mathcal{J}_{t,\tau}^p = \int_0^{\min\{t,\tau\}} \mathcal{F}_{t-s,\tau-s}^p ds \qquad \text{for } p = r, X$$

Equilibrium conditions

- Asset market clearing becomes

$$\hat{B}_t = \boldsymbol{\mathcal{E}_t^*} \hat{g}_0 + \sum_{p=r,X} \int_0^\infty \boldsymbol{\mathcal{J}_{t,s}^p} \boldsymbol{\hat{p}_s} ds$$

- Construct Jacobians for rest of economy
  - Here just simple scalar relationships
  - Combine with asset market clearing to link $\hat{B}_t$ to prices

- In operator and stacked notation $\hat{p} = [\hat{r}, \hat{X}, \hat{N}], \hat{\mathcal{Z}} = [\hat{Z}, \hat{\varepsilon}]$

$$\boldsymbol{\mathcal{J}} \boldsymbol{\hat{p}} = \underbrace{\hat{\mathcal{Z}} + \mathcal{E}^* \hat{g}_0}_{\substack{\text{exog. shocks and} \\ \text{initial conditions}}}$$

  - $\mathcal{J}$ is the sequence-space Jacobian

- Need to solve a linear system of size $3T \times 3T$ if discretize into T time periods

Computing the sequence-space Jacobian in practice

- Just as in discrete time, compute the SSJ through simple recursions:

$$\mathcal{J}^{p}_{t,0^-} = 0 \qquad \mathcal{J}^{p}_{0^-,s} = 0 \qquad \frac{d\mathcal{J}^{p}_{t+s,\tau+s}}{ds} = \mathcal{F}^{p}_{t+s,\tau+s}$$

  - Dirac mass point at in $\mathcal{D}$ implies that $\mathcal{J}^{p}_{t,0} \neq 0$
  - Similar recursions for other Jacobians

- Once discretize into T time periods and N gridpoints requires solving:
  1. 2 $N \times T$ linear PDEs for policy functions
  2. 2 $N \times T$ linear PDEs for expectation vectors
  3. $\sim T^2$ inner products for fake news operators
  4. $\sim 2T$ scalar ODEs for Jacobians
  5. 1 $3T \times 3T$ linear system inversion

Taking stock

- Overall computational complexity equivalent to discrete time
- Main advantage: obtain interpretable linear PDEs for policy functions instead of numerical diff.
  - Can exploit for additional speed gains
- Rest of computation similar to discrete time

Discretization and computation

Overview: steady-state

- Overall structure as in discrete time, but some details differ
- Guess a value for aggregates r, X, N
- Given this value for aggregates, solve the HJB equation
  - Typically use a **finite difference method** rather than EGM, though can also use EGM
  - **Achdou et al. (2022)**: in-depth econ-friendly description of algorithm
- Given the consumption function, obtain savings and solve the KF equation
  - Similar **finite difference method**
- Given solution to KF equation, update value for r, X, N
- Iterate on r, X, N until convergence using Newton or favorite solver
- Go to `Ben Moll's website` for great codes and tutorials

Solving the steady-state HJB equation (1/2)

- Discretize state space (a_k, y_i) and $V_{ik} = V(a_k, y_i)$, set time step $\Delta > 0$
- Given (r, X, N) solve HJB by starting from initial guess V^0 and iterating on

$$\begin{aligned} \rho V_{ik}^{n+1} &= u((u')^{-1}((\partial_a V)_{ik}^n)) - v(N) + \left[r a_k + X y_i^{1-\theta} - (u')^{-1}(\partial_a V_{ik}^n) \right] (\partial_a V)_{ik}^{n+1} \\ &+ \sum_j \lambda_{ij} (V_{jk}^{n+1} - V_{ik}^{n+1}) + \frac{V_{ij}^n - V_{ij}^{n+1}}{\Delta} \end{aligned}$$

  - n indices iterations
  - Derivatives are approximated with "upwinding"
    - ⋆ Forward vs. backward approx. depends on sign of drift, e.g.

$$(\partial_a V)_{ij}^{n+1} = \begin{cases} \frac{V_{i,k+1}^{n+1} - V_{ik}^{n+1}}{a_{k+1} - a_k} & \text{if} \quad \left[r a_k + X y_j^{1-\theta} - (u')^{-1}(\partial_a V_{ik}^n) \right] > 0 \\ \frac{V_{ik}^{n+1} - V_{i,k-1}^{n+1}}{a_k - a_{k-1}} & \text{if} \quad \left[r a_k + X y_i^{1-\theta} - (u')^{-1}(\partial_a V_{ik}^n) \right] < 0 \end{cases}$$

    - ⋆ **Intuitive:** approximate derivatives in the direction of drift
    - ⋆ **Important:** numerical stability of the scheme

Solving the steady-state HJB equation (2/2)

- Stack (i, k) into a single dimension and re-write HJB equation in matrix form

$$\boldsymbol{V}^{n+1} = \left[\left(\rho + \frac{1}{\Delta}\right)\boldsymbol{I} - \boldsymbol{\mathcal{L}}(\boldsymbol{V}^n)\right]^{-1}\left[\boldsymbol{u}(\boldsymbol{V}^n) - \boldsymbol{v}(\boldsymbol{N}) + \frac{1}{\Delta}\boldsymbol{V}^n\right]$$

  - $\mathrm{u}_{ij}(\boldsymbol{V}) = u((u')^{-1}((\partial_a \boldsymbol{V})_{ij}))$
  - $\boldsymbol{\mathcal{L}}(\boldsymbol{V})$ is a diagonal-by-block matrix
  - Has drift terms above/below diagonal depending on upwinding scheme

- Enforce boundary conditions

$$V_{i1}^{n+1} \text{ s.t} \qquad (\partial_a V)_{i1}^{n+1} = \max\left\{(\partial_a V)_{i1}^{n+1} \;,\; u'(ra_1 + Xy_i^{1-\theta})\right\} \;,\; \forall i$$

Interpretation

- Start from "terminal" condition V^0
- Iterate backwards in time (n) until value V remains unchanged
- Use the **time-dependent** HJB equation with $dt \equiv \Delta$ even for stationary case
  - n increases means time t decreases
  - Approximation of time derivative is

$$\partial_t V \equiv \frac{V_{ij}^n - V_{ij}^{n+1}}{\Delta}$$

- Similar to **policy function iteration**

Solving the steady-state KF equation

- Similar to HJB equation
- Formal duality between operators in HJB and KF makes it simple

$$0 = \boldsymbol{\mathcal{L}}(\boldsymbol{V})^* \boldsymbol{g}$$

  where now * denotes the matrix transpose

- Cannot simply invert discretized KF because KF operator is not full rank
- Replace one row/column of $\boldsymbol{\mathcal{L}}$ by identity matrix row/column $\to \tilde{\boldsymbol{\mathcal{L}}}$
- Invert (now full rank) linear system

$$\boldsymbol{g} = \left[\tilde{\boldsymbol{\mathcal{L}}}^*\right]^{-1} \mathbf{1}$$

  where
  - $\mathbf{1}$ is zero except at the replaced column of A
  - Normalize $g = g/\text{sum}(g)$

Solving for the Jacobians

- Given steady-state solving for φ^r, φ^X is super simple
- Just one time iteration using analytical formula:

$$\varphi_0^r = \boldsymbol{D_a} \cdot \left(\boldsymbol{a} \circ \boldsymbol{u}'(\boldsymbol{c^{ss}})\right), \qquad \varphi_{t+dt}^r = \boldsymbol{\Phi}\varphi_t^r \qquad \boldsymbol{\Phi} = \boldsymbol{I} + (r^{ss} - \boldsymbol{D_a}\boldsymbol{c}^{ss})\boldsymbol{I} + \boldsymbol{\mathcal{L}}$$

$$\varphi_{0,i1}^r = u''(c_{i1}^{ss})a_{i1} \text{ if cons.} \qquad \varphi_{t+dt,i1}^r = 0 \text{ if cons.}$$

where

  - $\circ$ is the element-wise product and $\boldsymbol{D_a}$ encodes the asset derivative as in steady-state
  - We apply the boundary condition iff $(y_i, \underline{a})$ is constrained in steady-state
  - Similarly for φ^X

- Then construct $\boldsymbol{\mathcal{D}}^r, \boldsymbol{\mathcal{D}}^X$ using analytical formula

$$\boldsymbol{\mathcal{D}}_s^r = -\boldsymbol{D_a} \cdot \left(\boldsymbol{g}^{ss} \circ \left(\boldsymbol{a}\ 1\{s=0\} + \frac{e^{-\rho s}}{\boldsymbol{u}''(\boldsymbol{c^{ss}})} \circ \varphi_s^r\right)\right)$$

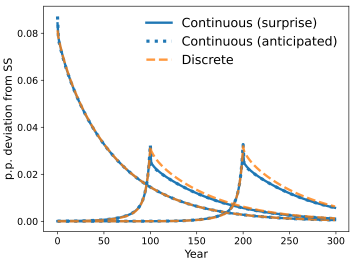
where similarly for $\boldsymbol{\mathcal{D}}^X$

- Then construct Jacobians exactly as in discrete time given $\boldsymbol{\mathcal{D}}^r, \boldsymbol{\mathcal{D}}^X$

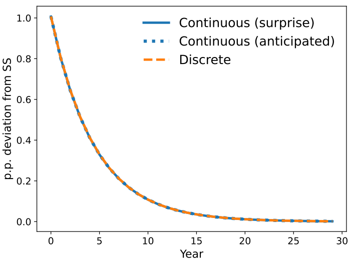
Example

Consumption Jacobian and IRFs

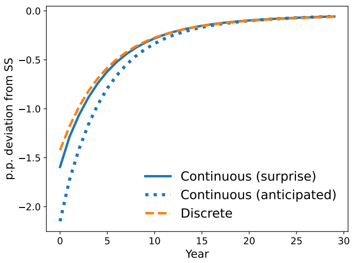
(a) Cons. Jacobian to income



(b) Output to gov. spending



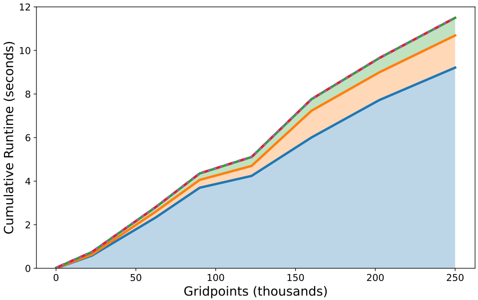
(c) Output to monetary shock



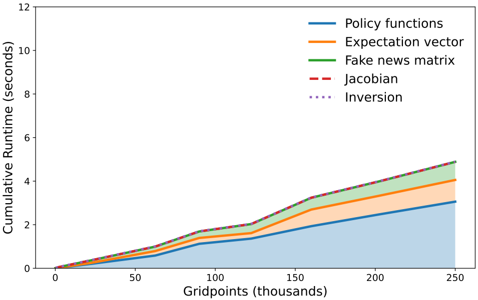
Note: Solid blue: continuous time with a discretization that imposes information aggregation as in discrete time. Dotted blue: continuous time with a discretization that does not impose information aggregation: changes in income and prices are a surprise only in instant $t = 0$ but not at any $t \in (0, 1]$ (see paper for details). Dashed orange: discrete time. 50 income states and 5,000 asset gridpoints.

HANK IRF computation times

(a) Discrete time



(b) Continuous time



Conclusion

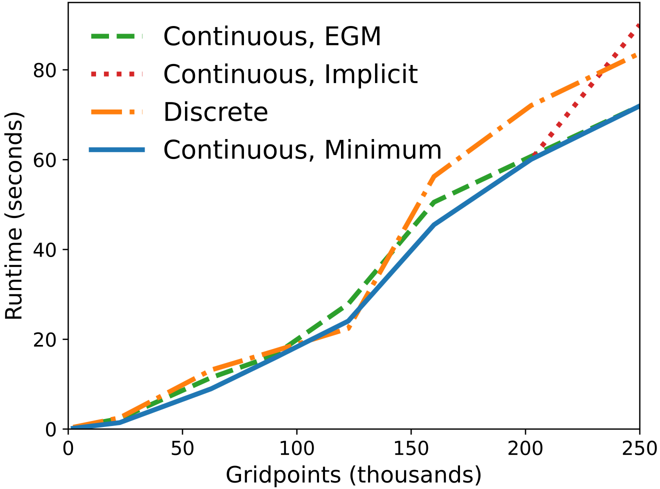
Conclusion

- Provide a sequence-space Jacobian routine for continuous-time models
- Obtain additional analytical traction, leading to speed gains
- Demonstrate how to use it to solve workhorse heterogeneous agent models
- Python and Matlab routines available at
  - `https://github.com/ShlokG/CT-SSJ/`
  - or `https://sites.google.com/site/adrienbilal/`

Thank You!

Appendix

HA steady-state computation times



Back to Jacobian