Continuous-time methods for heterogeneous-agent models in macroeconomics: a spectral approach

Master Thesis - Paris School of Economics

Constantin Schesch

Supervisors: Daniel Cohen, Tobias Broer

September 9th, 2022

Outline

Motivation

Literature review & motivation

1. Numerical methods (e.g. Krusell & Smith, 1998): how to solve heterogeneous-agent models (with aggregate shocks) numerically?
2. Mean Field Games (e.g. Lasry & Lions, 2007): game theory with $N \to \infty$
3. Achdou, Han, Lasry, Lions & Moll: "Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach" (REStud, 2021)
   - Recast HA model (without aggregate shocks) as MFG in continuous time
   - Coupled system of Partial Differential Equations: HJB & KFE
   - Can be solved elegantly & (very!) quickly using (upwind) finite differences

Q: Can spectral methods be used to solve continuous-time HA models?

$\Rightarrow$ Yes, to some extent $\rightarrow$ Present simple applications that play to their strengths

Prolegomenon: what is a (pseudo-)spectral method?

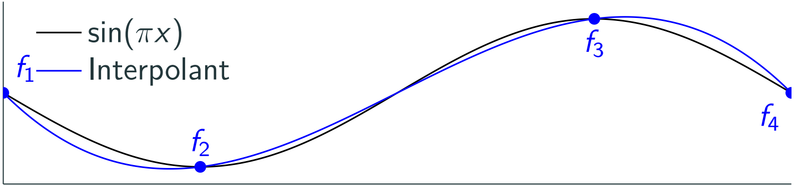
Three main approaches to solve differential equations

$$f'(x) = g(x)$$

1. Finite differences (FD): f as piece-wise linear function
2. Finite elements (FEM): f as sum of local basis functions (e.g. hat functions)
3. Spectral method: f as sum of global basis functions (e.g. polynomials)

- 2 & 3 similar in theory, but
- 1 & 3 similar in practice when using collocation ("pseudospectral method")

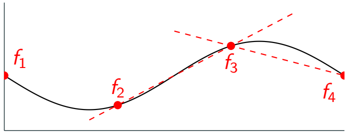
Pseudospectral methods (1/2): polynomial interpolation



- Write $f = \sum_{n=0}^{\infty} c_n T_n$ with (T_n) polynomial basis (e.g. Chebyshev)
- In practice, we can only write N first terms:
  - Polynomial of order N defined by its values at nodes $x_0, \ldots, x_N$
  - For well-chosen nodes, errors decrease very fast (order: $\frac{\max_{[-1,1]} |f^{(N)}|}{2^{N-1} N!}$)
  - Derivative at each node $P'(x_k)$ is a **linear** function of node values $f_0, \ldots, f_N$
    $\rightarrow$ can be assembled into $(N+1) \times (N+1)$ **differentiation matrix** D

Pseudospectral methods (2/2): solving differential equations

Finite Differences

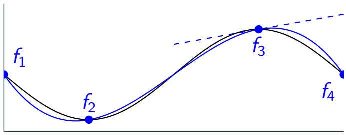


- Compute the derivative at a point as difference from previous (or next) point
- Assemble into a differentiation matrix

$$\frac{1}{\Delta x}\begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix}$$

- Solve the equation: $f = D_{FD}^{-1}\, g$

Pseudospectral Method



- Compute the derivative at a point as derivative of the polynomial
- Assemble into a differentiation matrix

$$\begin{pmatrix} \frac{19}{16} & -4 & \frac{4}{3} & -\frac{1}{2} \\ 1 & -\frac{1}{3} & -1 & \frac{1}{3} \\ -\frac{1}{3} & 1 & -\frac{1}{3} & -1 \\ \frac{1}{2} & -\frac{4}{3} & 4 & -\frac{19}{16} \end{pmatrix}\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix}$$

- Solve the equation: $f = D_{Sp}^{-1}\, g$

Solving a two-income Hugget model with the spectral method

Two-income model in continuous time: setup, HJB & KFE

- Agents' income z_t follows Markov process over $\{z_1 < z_2\}$, intensities λ_1, λ_2
- Hold assets a_t but credit-constrained by $\underline{a} \leq a_t$, r is exogeneous
- Utility is $u(c) = {}^{c^{1-\gamma}}/_{1-\gamma}$, future is discounted at rate ρ

1. **Hamilton-Jacobi-Bellman (HJB)**: characterizes value functions $v_j(a)$

$$\rho v_j(a) = \underbrace{\max_c u(c) + (z_j + ra - c)v_j'(a)}_{\text{Savings vs consumption trade-off}} + \underbrace{\lambda_j(v_{-j}(a) - v_j(a))}_{\text{Income-switching}} \qquad \forall j,\ \forall a \in [\underline{a}, +\infty]$$

2. **Kolmogorov Forward Equation (KFE)**: characterizes stationary densities $g_j(a)$ via savings function $s_j(a)$ (depends on v_j)

$$0 = \underbrace{-\frac{d}{da}[s_j(a)g_j(a)]}_{\text{Savings}} - \underbrace{\lambda_j g_j(a) + \lambda_{-j} g_{-j}(a)}_{\text{Income-switching}} \qquad \forall j \in \{1,2\}, \forall a \in [\underline{a}, +\infty],$$

Solving the HJB equation spectrally is fairly straightforward

- Choose N collocation nodes over $[\underline{a}, \bar{a}]$, assemble differentiation matrix D
- *Max* in HJB expensive to solve directly $\rightarrow$ use (implicit) iterative scheme, start from an initial guess v_j^0 and update using time steps Δt:

$$\underbrace{\frac{v_{j,n}^{k+1} - v_{j,n}^{k}}{\Delta t}}_{\text{Update of v}} + \rho v_{j,n}^{k+1} = \underbrace{u(c_{j,n}^{k}) + \left(z_j + r a_n - c_{j,n}^{k}\right) v_{j,n}^{k+1\prime}}_{\text{No max because c pinned down by FOC}} + \lambda_j \left(v_{-j,n}^{k+1} - v_{j,n}^{k+1}\right)$$

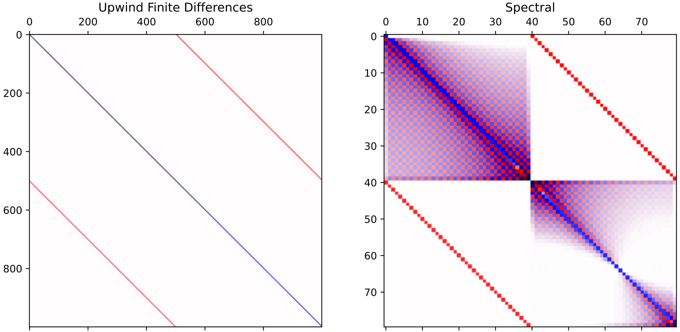
- With $\mathrm{v}_j^k := (v_j(a_1), \ldots, v_j(a_N))$ and $D_n := D$'s n-th row, collocation writes:

$$\frac{v_{j,n}^{k+1} - v_{j,n}^{k}}{\Delta t} + \rho v_{j,n}^{k+1} = u(c_{j,n}^{k}) + (z_j + r a_n - c_{j,n}^{k}) D_n \mathrm{v}_j^{k+1} + \lambda_j \left(v_{-j,n}^{k+1} - v_{j,n}^{k+1}\right)$$

- Stack into $2N \times 2N$ matrix equation, using $\mathbf{D} := I_2 \otimes D$ and $\mathbf{G} := G \otimes I_N$:

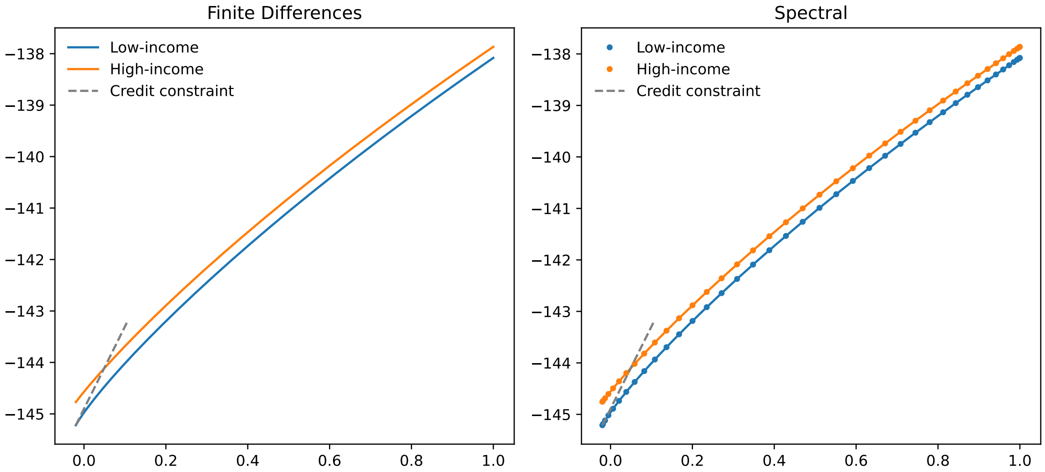
$$\frac{1}{\Delta}(\mathbf{v}^{k+1} - \mathbf{v}^k) + \rho \mathbf{v}^{k+1} = \mathbf{u}^k + \mathbf{s}^k \mathbf{D}\mathbf{v}^{k+1} + \mathbf{G}\mathbf{v}^{k+1}$$

Matrix $\mathbf{A} := \mathbf{s}\mathbf{D} + \mathbf{G}$ in different approaches: sparsity vs global differentiation



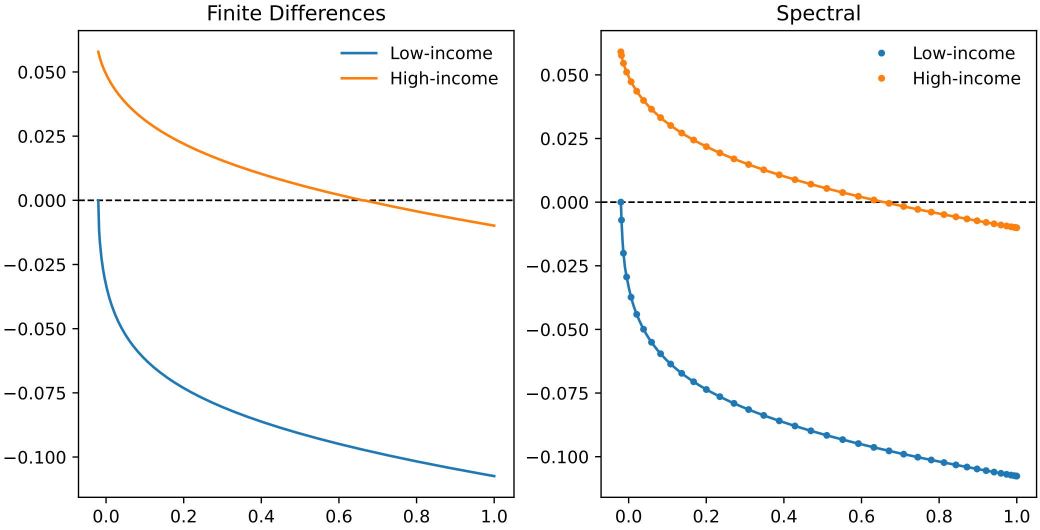
- Credit constraint $\Rightarrow$ boundary condition on v_j $\rightarrow$ **easy** boundary-bordering

Two-income model: value functions



FD: 500 asset nodes / Spectral: 40 asset nodes

Two-income model: savings functions



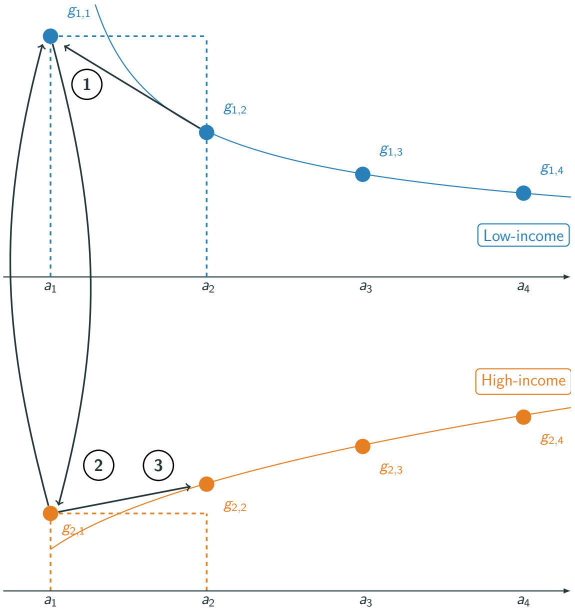
- Also: error analysis, checks on boundary conditions, consistency checks etc.

Solving the KFE spectrally impossible without refinements

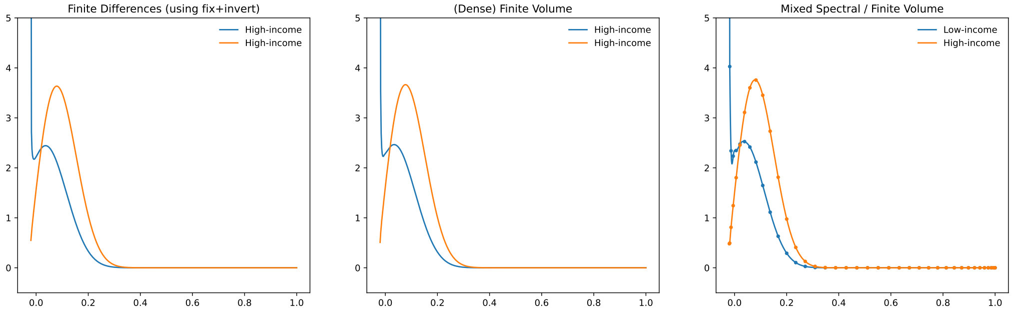
- KFE plays to FD's strength and against spectral weaknesses:
  - Operators in KFE & HJB adjoint: for FD transpose A, **doesn't work here**
  - Dirac mass of agents at credit constraint (cf. Corr. 1): **no density!**

1. Solve spectrally nonetheless → works when $\underline{a} \sim \underline{a}^\natural = -z_1/r$
2. Solve using finite volumes → easy to program, fast & can use high-def grid
3. Solve using a hybrid scheme → best of both worlds?
   - Spectral part: most of the distribution admits very smooth density
   - Finite volume part: model in- and out-flows of cell around Dirac mass

Illustration of mixed spectral / finite volume approach to KFE



Stationary distribution

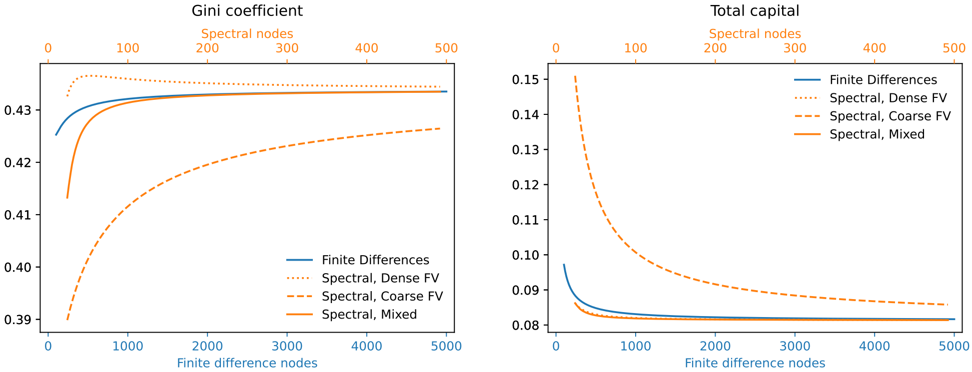


- Finite volumes works well; mixed method too, remarkable given # of nodes

Benchmark

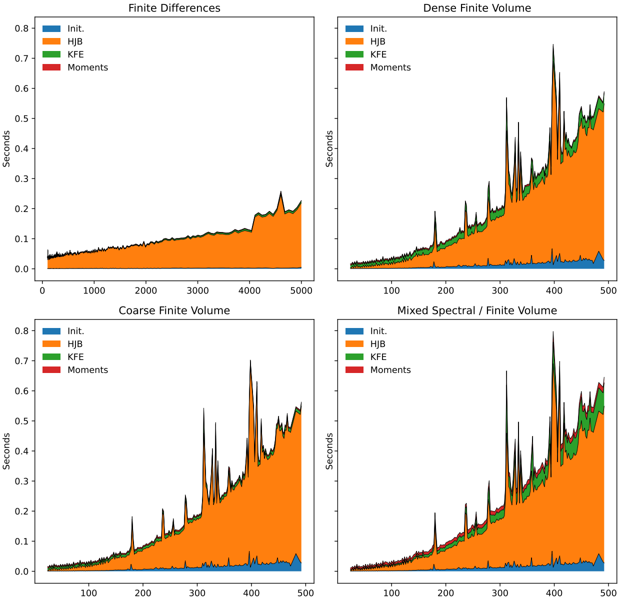
Given # of nodes, spectral outperforms FD...

Distributional moments by number of nodes



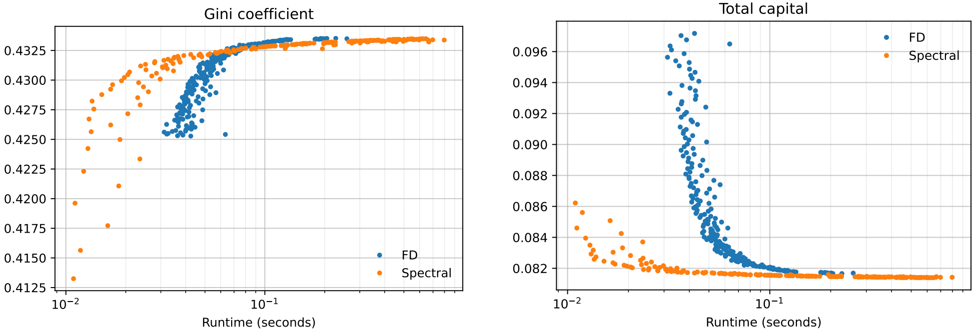
NB: Axes are not the same!

...but is also is much slower...



...though the balance tilts in favor of spectral

Distributional moments by execution time



Diffusive income

Diffusive income model: setup, HJB & KFE

- Now, agents' income z_t follows an **Ornstein-Uhlenbeck** ($\sim$AR1) process $dz_t = \theta(\mathring{z} - z_t)\,dt + \sigma\,dB_t$, reflected between $\underline{z} < \bar{z}$

1. Hamilton-Jacobi-Bellman (HJB): characterizes value $v(a, z)$

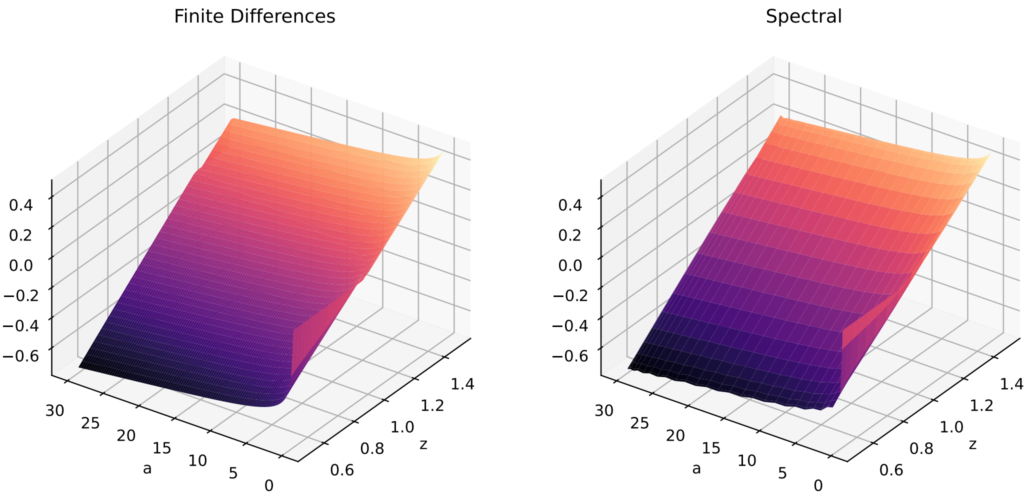
$$\rho v(a,z) = \underbrace{\max_c u(c) + [z + ra - c]\partial_a v(a,z)}_{\text{Savings vs consumption trade-off}} + \underbrace{\mu(z)\partial_z v(a,z)}_{\text{Income drift}} + \underbrace{\frac{\sigma^2}{2}\partial_{zz} v(a,z)}_{\text{Income diffusion}}$$

2. Kolmogorov Forward Equation (KFE): characterizes density $g(a, z)$

$$0 = -\underbrace{\partial_a[s(a,z)g(a,z)]}_{\text{Savings drift}} - \underbrace{\partial_z[\mu(z)g(a,z)]}_{\text{Income drift}} + \underbrace{\frac{1}{2}\partial_{zz}\left[\sigma^2(z)g(a,z)\right]}_{\text{Income diffusion}}$$

- Solution is **very smooth along diffusive dimension** (income)
- Approximation now **tensor of polynomials** $\rightarrow$ broadcast matrices

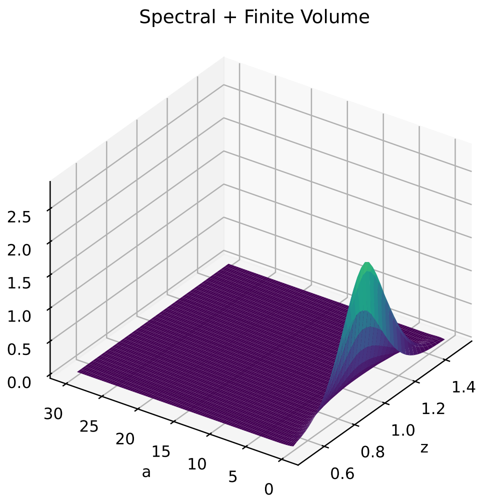
Diffusive income model: Savings functions

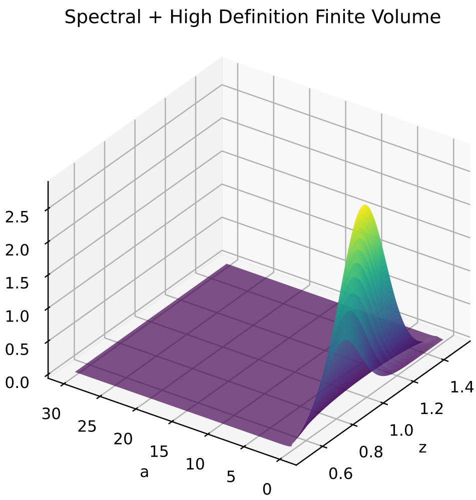


FD: 100 asset, 40 income nodes / Spectral: 30 asset, 20 income nodes

Diffusive income model: stationary distribution

- Hybrid approach too complex, have to use finite volumes $\rightarrow$ back to FD?
- Reliable interpolation to high-def. grid: $<$10s to solve KFE on 1000×400 grid





Life-cycle

Life-cycle model: setup, HJB & KFE

- To yield log-normal distribution, income follows exponential OU, $d\ln z_t = \theta(\ln \mathring{z} - \ln z_t)\,dt + \sigma\,dB_t + \psi\,dt$, where ψ is the age-income gradient
- Agents live from $\underline{t}$ to $\bar{t}$; at death, they spawn a descendant with same wealth but deflated income $\frac{z_{\bar{t}}}{(1+\psi)^{\bar{t}-\underline{t}}}$; discount offspring's value by $\alpha \in [0,1]$

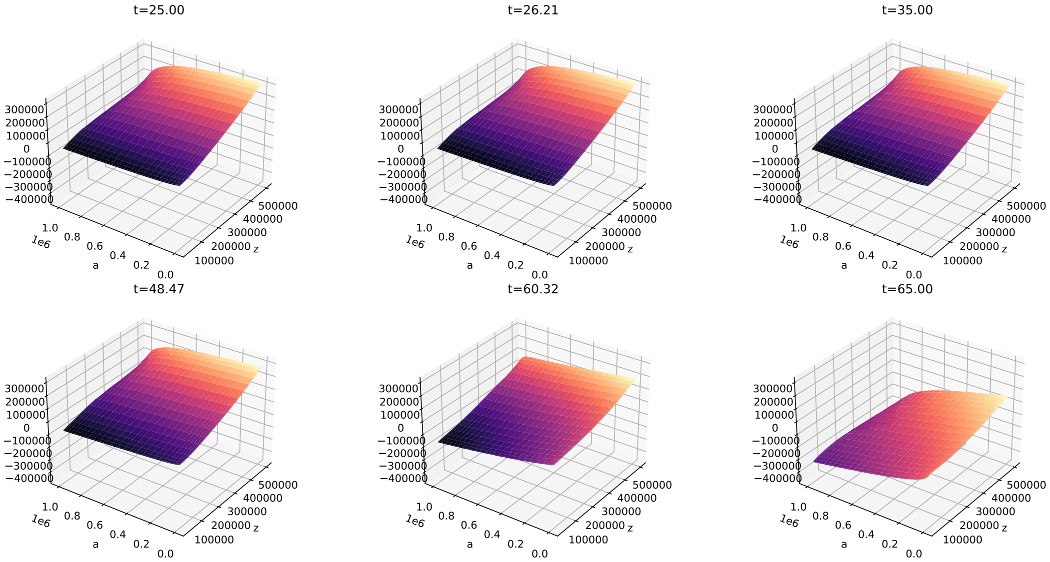
1. **Hamilton-Jacobi-Bellman (HJB)**: characterizes value $v(a,z)$

$$\rho v(a,z,t) = \underbrace{\max_c u(c) + [z + ra - c]\partial_a v(a,z,t)}_{\text{Savings vs consumption trade-off}} + \underbrace{\mu(z)\partial_z v(a,z,t)}_{\text{Income drift}} + \underbrace{\frac{\sigma^2}{2}\partial_{zz} v(a,z,t)}_{\text{Income diffusion}} + \underbrace{\partial_t v(a,z,t)}_{\text{Time drift}}$$

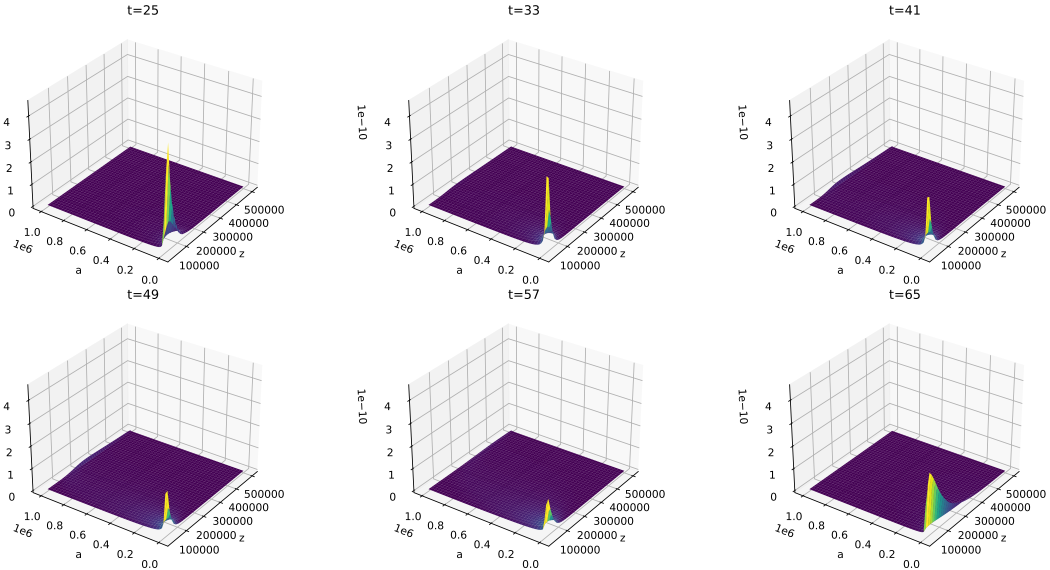
2. **Kolmogorov Forward Equation (KFE)**: characterizes density $g(a,z)$

$$0 = -\underbrace{\partial_a[s(a,z,t)g(a,z,t)]}_{\text{Savings drift}} - \underbrace{\partial_z[\mu(z)g(a,z,t)]}_{\text{Income drift}} + \underbrace{\frac{1}{2}\partial_{zz}\left[\sigma^2(z)g(a,z,t)\right]}_{\text{Income diffusion}} - \underbrace{\partial_t g(a,z,t)}_{\text{Time drift}}$$

Life-cycle model: savings function



Life-cycle model: stationary distribution

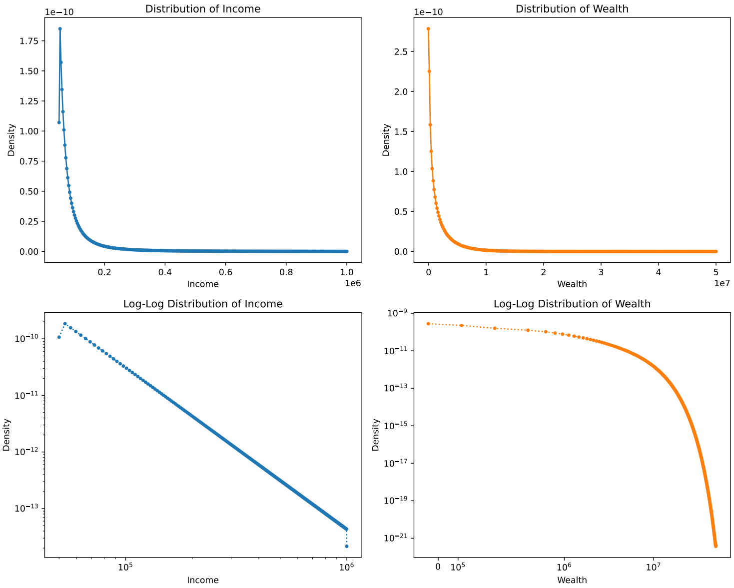


Power laws, eigenvalues and "slow transitions"

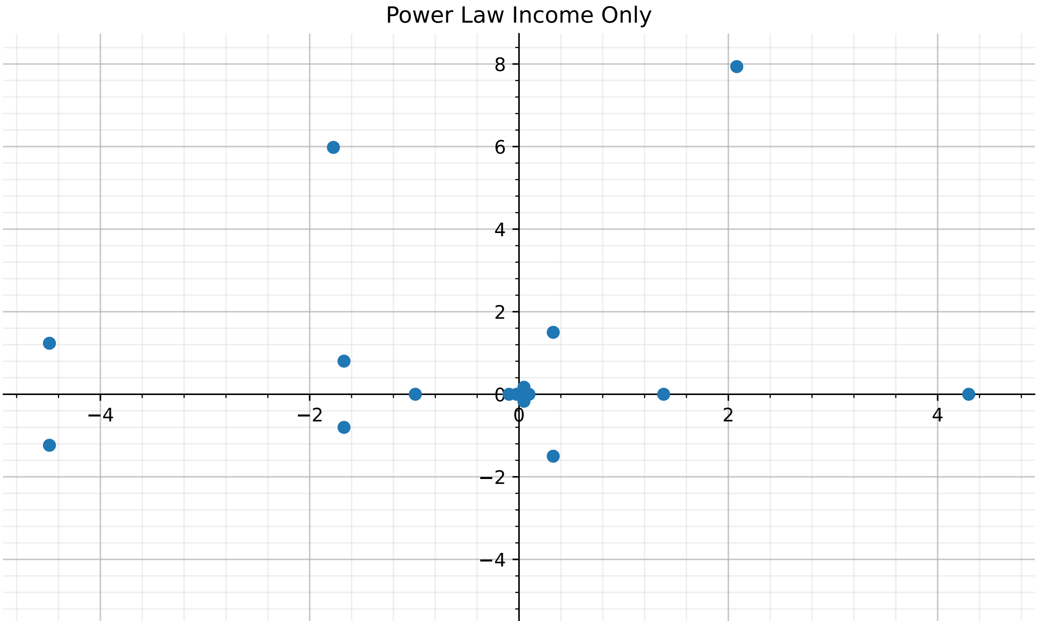
"Slow transitions" under Power-distributed income

- Gabaix, Lasry, Lions & Moll, "The Dynamics of Inequality" (2016):
  - Reasonable models of income, i.e. which yield Power law, also yield...
  - ... "slow transitions", i.e. change in inequality far slower than seen since 70s
  - Based on analytical characterization of eigenvalues of generator $\mathcal{A}$
- Finite volume discretization of KFE yields A, transition rate matrix of discrete-state, continuous-time process: discrete analog of $\mathcal{A}$
- $\Rightarrow$ **Compute eigenvalues *numerically*** for income *and* wealth inequality
- Now, income follows "random growth": $d \ln z_t = g dt + s\, dB_t$
- Added value:
  - Solve HJB spectrally on large space: 50k-1m income & -50k-50m assets
  - $\rightarrow$ interpolate to 300×300 grid for KFE: yields A of size $90{,}000 \times 90{,}000$

Power law income: distribution cross-sections



Power law income: even slower transitions with wealth!



Conclusion

Kolmogorov Forward Equation: where the trouble starts...

- Pseudospectral approach to heterogeneous-agent models has advantages:
  - Solutions to HJB are usually smooth, leverage to save on nodes
  - Easy to explain, much easier to code up than than FD
- ... but also a number of shortcomings
  - KFE can only be solved using tricks or resorting to FV (but high-def.)
- Avenues for future research:
  - Make spectral approach to KFE work: via weak form?
  - Adapt to high-dimensional problems ($d \geq 4$):
    - Garcke & Ruttscheidt (2017), Schaab & Zhang (2022): sparse grids for FD
    - Develop Smolyak-based sparse integration / collocation methods
    - Draw on active work in optimal control, e.g. Dolgov, Kalise & Kunisch (2022)

Thank you for your attention!

Appendix

Calibrations and numerical parameters

	Two-state income	Diffusion income	Life-cycle	"Slow transitions"
Macroeconomy				
r	0.035	0.04	0.04	0.04
Utility				
Type	CRRA	CRRA	CRRA	CRRA
γ	1.2	2	2	2
ρ	0.05	0.05	0.05	0.05
Income				
Type	Two-state Markov	Ornstein-Uhlenbeck	Exponential O.-U.	Exponential O.-U.
$\hat{z}$		1	\$70,000	\$70,000
$\underline{z}$	0.1	0.5	\$30,000	\$50,000
$\bar{z}$	0.2	1.5	\$500,000	\$1,000,000
λ_1, λ_2	1.5, 1.0			
σ^2		0.05	0.0974	0.0974
θ		1	0.0682	0.0682
Assets				
$\underline{a}$	-0.02	-0.1	\$0	-\$50,000
$\bar{a}$	1.5	30	\$1,000,000	\$50,000,000
Life cycle				
$\underline{t}, \bar{t}$			25, 65	
α			0.1	
ψ			1.58%	
Spectral parameters				
N	40	30	30	30
M		20	15	20
O			10	
FD parameters				
I	500	100	80	100
J		40	40	80
K			41	
Source:	*huggett_partialeq.m*	*huggett_diffusion_partialeq.m*	*See Section 6.1*	*See Section 7.1*