
Continuous-Time Methods for Heterogeneous-Agent Models in Macroeconomics: A Spectral Approach

Constantin Schesch

August 2022

Supervisors: Prof. Tobias Broer, Prof. Daniel Cohen

Master's Thesis

Master in Analysis and Policy in Economics



PARIS SCHOOL OF ECONOMICS
ECOLE D'ECONOMIE DE PARIS

JEL Codes: C61, C63, C88, D31, G12, G14

Keywords: *Heterogeneous agents, continuous time, spectral, pseudospectral, collocation*

Abstract: Over the last decades, the attention received by heterogeneous-agent models has only been rivalled by the considerable difficulties macroeconomists encountered in solving them numerically. Achdou et al. (2021)'s hallmark contribution recast these models in continuous-time, describing them as coupled Hamilton-Jacobi-Bellman and Kolmogorov Forward partial differential equations, which can then be solved very quickly using (upwind) finite differences. In their footsteps, we propose to solve continuous-time heterogeneous-agent models using (pseudo-)spectral methods, which represent the solution as a sum of global basis functions, e.g. Chebyshev polynomials.

In a Hugget model with two incomes and credit-constrained savings, the smoothness of the value function means that a good approximation can be obtained using few collocation nodes. Solving for the stationary distribution is more involved because the Dirac mass of agents at the borrowing constraint cannot be represented spectrally, so that a hybrid scheme combining a smooth spectral domain and a Dirac-like finite volume domain has to be employed. Benchmarks show that this outperforms a pure finite difference approach, with precise estimations taking less than 1/10th of a second.

To illustrate the added value of the pseudospectral approach in higher-dimensional problems, we extend it to diffusive income and then to a life-cycle. Each time, the smoothness of the spectral solution to the HJB equation means that relatively few nodes in each dimension yield a precise solution. The resulting savings function can then be interpolated to a high definition grid to quickly & easily solve for the stationary distribution using finite volumes.

Finally, we show that the pseudospectral approach can be useful to solve savings problems over a large value range by applying it to a model with power law income. The finite volume discretization of the Kolmogorov Forward Equation yields a discretized transition rate matrix whose eigenvalues can be computed to study the puzzle of "slow transitions", formulated in Gabaix et al. (2016) for incomes, in a model incorporating both income and wealth inequality.

Contents

1	Introduction	4
2	Literature review	5
2.1	Heterogeneous-agent models & Mean Field Games	5
2.2	Numerical methods for Mean Field Games	6
2.3	Spectral methods for Hamilton-Jacobi-Bellman equations	7
2.4	High-dimensional methods for Hamilton-Jacobi-Bellman equations	9
3	Prolegomenon: spectral & pseudospectral methods	10
4	Spectral method for a Huggett model with two incomes	13
4.1	Description of the two-income Huggett model	13
4.2	Solving for optimal savings: the Hamilton-Jacobi-Bellman equation	14
4.2.1	Numerical procedure	14
4.2.2	Numerical results	18
4.3	Solving for the stationary distribution: Kolmogorov Forward Equation	22

4.3.1	"Not because they are easy, ..."	22
4.3.2	Pure spectral approach	23
4.3.3	Upwind finite difference / finite volume approach	24
4.3.4	Mixed spectral / finite volume approach	30
5	Spectral method for a Huggett model with diffusive income	32
5.1	A simple heterogeneous-agent model with Ornstein-Uhlenbeck income	32
5.2	Solving the diffusive Hamilton-Jacobi-Bellman equation	33
5.2.1	Numerical procedure	33
5.2.2	Numerical results	36
5.2.3	Boundary condition and consistency checks	39
5.3	Solving the diffusive Kolmogorov Forward equation	40
5.3.1	Spectral approach	40
5.3.2	Finite volume approach	41
6	Spectral method for a life-cycle model	44
6.1	A simple life-cycle model	44
6.2	Solving the life-cycle Hamilton-Jacobi-Bellman equation	46
6.3	Solving the life-cycle Kolmogorov Forward equation	47
7	Application: "slow transitions" with wealth?	51
7.1	A simple model of Power-law income with credit-constrained savings	51
7.2	Characterizing transition speeds using the spectral gap	53
8	Benchmark: are spectral methods actually better?	58
8.1	Errors for a given number of nodes	58
8.2	Runtime for a given number of nodes	59
8.3	Errors for a given runtime	63
9	Conclusion	63
Appendices		70
A	Codes	70
B	Calibrations and numerical parameters	71
C	Additional plots for the two-income Hugget model	72
D	Additional plots for the diffusive income Hugget model	74
E	Additional plots for the life-cycle model	78
F	Additional plots for the "slow transitions" application	80
G	Additional plots for the two-income benchmark	82

"Il n'y a de vraiment beau que ce qui ne peut servir à rien."

Théophile Gautier, *Mademoiselle de Maupin* (1835)

1. Introduction

Rising inequality across advanced economies since the end of the 20th century has prompted growing interest for macroeconomic models that account for household's heterogeneous income and wealth. However, heterogeneous-agent models are complex and thus often impossible to solve analytically. Solving these models numerically has also proven to be a challenge, largely because the distribution of agents along the dimension of heterogeneity is an infinite-dimensional object that is difficult to represent and manipulate on a computer.

Although most numerical endeavors on this topic since the early 1990s have focused on discrete-time methods, recent contributions by Achdou et al. (2021) have stressed the analogy between heterogeneous-agent models and Mean Field Games (MFGs) in continuous time. Such models can then be recast as a coupled system of a Hamilton-Jacobi-Bellman equation describing the optimal behavior of agents and a Kolmogorov Forward equation describing their distribution. These coupled PDEs can then be solved via upwind finite differences, which is made particularly elegant (and efficient) because of their adjoint structure.

Whereas finite differences approximate the solution to a PDE by a piecewise linear function, many other numerical approaches to solving PDEs exist. In particular, we propose to use spectral methods, which represent the solution as a sum of global basis functions (e.g. Fourier series, Chebyshev polynomials), and can then be used to solve the PDE via a Galerkin approach. In practice, we will use so-called pseudospectral methods, which characterize the combination of basis functions by its values on certain collocation nodes (e.g. Chebyshev nodes). As these methods typically perform very well for smooth problems with relatively few nodes, it is natural to use them to solve common heterogeneous-agent models.

Section 2 presents a review of the economic and mathematical literature on this topic, focusing on heterogeneous-agent models and mean field games, numerical methods for mean field games, spectral methods for HJB equations and high-dimensional methods for HJB equations.

Section 3 presents a brief introduction to spectral and pseudospectral methods, though the reader is mainly encouraged to read one of the many fascinating textbooks on the topic.

Section 4 applies the spectral approach to the two-income heterogeneous-agent model which forms the benchmark in Achdou et al. (2021): the solution of the HJB equation is very straightforward and works remarkably well, while the solution of the Kolmogorov Forward equation requires some adaptation to account for the Dirac mass of agents at the borrowing constraint but then works similarly well.

Section 5 solves a heterogeneous-agent model with diffusive Ornstein-Uhlenbeck income. As the income dimension is now also described by a partial differential operator, it can be solved for spectrally. This works extremely well for the HJB equation given the smoothness of the optimal value function. However, solving spectrally for the stationary distribution again does not work, and hybrid approaches become too complex. The simplest approach is thus to solve for the distribution via upwind finite volumes ¹, which is fast and easy to implement and can thus be done on a much finer grid.

Section 6 showcases the power of the pseudospectral approach by solving a smooth three-dimensional problem, namely a life-cycle model with exponential Ornstein-Uhlenbeck income, borrowing con-

¹Which is essentially the upwind finite differences method from Achdou et al. (2021).

straints and bequests. The spectral approach again leverages the smoothness of the solution across the time dimension to reduce the number of nodes required for an accurate solution.

Section 7 presents a final application to a problem that is in and of itself simple, a constrained savings problem with Power law income, but where the solution has to cover a very large range (as Power laws admit extreme outliers, i.e. very high earners). The smoothness of the spectral approach to the HJB equation overcomes this problem, while the transition matrix interpretation of the finite volume discretization of the KFE equation allows us to study the puzzle of "slow transitions" between inequality regimes identified in Gabaix et al. (2016), but in a more general setting with income *and* wealth inequality.

Finally, Section 8 presents a benchmark comparison with upwind finite differences in the two-income case, showing that the spectral method is much faster for a given precision. As all of these methods are extremely fast (typically taking less than a second to a few seconds depending on the problem at hand), the main added value of spectral methods remains in their ability to treat smooth dimension using very few nodes.

Section 9 concludes.

Appendix A outlines the codes accompanying this Master Thesis, which are available on GitHub. Other Appendices provide additional tables and plots for the main Sections.

2. Literature review

This Master Thesis lives at the intersection of four strands of scientific literature: (i) the economic literature on heterogeneous-agent models, and specifically their treatment as a class of Mean Field Games, (ii) numerical methods for Mean Field Games, which typically revolve around finite differences, (iii) spectral methods for Hamilton-Jacobi-Bellman equations, which have been successfully applied across a large array of situations, and, aspirationally, (iv) to numerical methods for high-dimensional HJB equations.

2.1. Heterogeneous-agent models & Mean Field Games

Although economic inequalities have been discussed at least since Aristotle, models featuring agents with heterogeneous incomes have made their debut in macroeconomics only fairly recently, with the hallmark contributions of Aiyagari (1994), Huggett (1993) and Bewley (1986). These models typically feature agents with different, stochastic incomes who accumulate precautionary savings, but are credit constrained as they cannot borrow below a certain threshold. Heterogeneous incomes combined with a credit constraint can help explain empirically high marginal propensities to consume (MPCs), which are critical for understanding aggregate dynamics, and have thus seen hundreds of publications over the last three decades. See Cherrier (2021) for an extensive historical review on this topic.

Heterogeneous-agent models have proven near-impossible to solve analytically and very challenging to simulate. Hallmark contributions have been Den Haan (1997), who proposed to track the distribution of agents using a low-dimensionally parametrized exponentials, and especially Krusell and Smith (1998), who showed that only next-period capital, i.e. the average of the asset distribution, mattered for agents' decisions ("quasi-aggregation") and that simple forecasting rules could thus be used to simulate a heterogeneous-agent economy. These approaches (reviewed e.g. in Den Haan (2010)) typically feature idiosyncratic and aggregate shocks, and operate in discrete time using a combination of projection and perturbation methods. It remains a highly active area of research, with recent contributions like Auclert et al. (2021) significantly improving the performance of discrete-time simulations.

More recently, heterogeneous-agent models have seen increased attention from the mathematics community as they have been described as instances of Mean Field Games (MFGs), as theorized

by Lasry and Lions (2007). Mean-field game theory describes decisions made by interacting agents when the number of agents becomes infinite and their relative importance infinitesimal; each agent is thus said so interact with the "mean field" of other agents, a terminology imported from physics. A MFG is then typically composed of a backward Hamilton-Jacobi-Bellman equation that describes agents' optimal behavior (e.g. savings) coupled with a forward Kolmogorov Forward equation that characterizes the distribution of agents.

This analogy has allowed to adapt numerical methods for MFGs (reviewed below) to simulate heterogeneous-agent models in continuous-time, as is masterfully done in Achdou et al. (2021). Authors first develop the analogy between MFGs and standard HA models, also stressing some important differences, recast the standard Ayagari-Hugget-Bewley model as a continuous-time MFG and derive a number of analytical properties in a two-income benchmark model. They then show how an upwind finite difference discretization of the HJB equation can be used in combination with an iterative scheme to solve the non-linear equation. Finally, they show how to solve for the stationary distribution by exploiting the adjointness of the HJB and KF equations, by simply transposing the discretization operator of the HJB equation (which admits a natural interpretation as the generator of a discrete-space, continuous-time Markov process) and finding its stationary distribution. All this works on HA models with an exogenous interest rate, but Achdou et al.'s solution method can be nested in capital-supply-and-demand iterations to add a productive sector, or combined with classical time-marching schemes to solve for transition dynamics.

This seminal contribution has since then been deepened and applied to a number of areas. In Ahn et al. (2018), many of the same authors outline how the MFG approach, which only accounts for idiosyncratic shocks, can be applied to an economy with aggregate shocks: put briefly, the idea is to solve for the stationary distribution of the economy without aggregate shocks, to compute derivatives (using automatic differentiation) with respect to aggregate state variables, and to simulate the aggregate economy using this linearization.

The continuous-time approach to heterogeneous-agent models has also been applied to the computation of social optima, e.g. Nuño and Moll (2018) and Nuño and Thomas (2019), to economies with heterogeneous - Recherche Google firms à la Hopenhayn (1992) as in Shaker Akhtekhane (2017) and Moll (2018). Fernández-Villaverde et al. (2019) complement this approach with a neural network to forecast aggregate variables, which allows them to study the interaction of idiosyncratic and aggregate shocks in financial friction cycles.

Achdou et al. (2021) is by far the main inspiration for this Master Thesis. As it stands on the shoulders of a large body of work on numerical methods for Mean Field Games, we synthesize this literature in the following Subsection.

2.2. Numerical methods for Mean Field Games

Numerical methods for Mean Field Games have mostly relied on finite difference approximations combined with a Newton or Newton-like iteration, e.g. Achdou and Capuzzo-Dolcetta (2010) and Achdou (2013). Key early insights were that the forward-backward structure of the coupled HJB & KF equations meant that standard time-marching methods could not be used, while their adjoint nature could be exploited. These approaches were then also extended to games where agents interact through their controls (Achdou and Kobeissi, 2020).

Finite difference approximations are powerful because they rely on a number of proven convergence theorems thanks to their monotonic nature, in turn relying on the results of Barles and Souganidis (1991), Barles and Jakobsen (2002) and Barles and Jakobsen (2007). The generality of these results rest on the theory of viscosity solutions (Crandall and Lions, 1983), which is especially applicable to non-smooth (e.g. discontinuous or kinked) problems, which we will not attempt to study using the spectral approach.

Many more numerical approaches to Mean Field Games have been developed over the years and we

cite only the most influential. Carlini and Silva (2014) propose a fully discrete, semi-Lagrangian scheme to solve the equation along characteristics, and provide convergence results under suitable assumptions for $d = 1$. Lauriere and Pironneau (2016) describe a method based on Gradient Descent for the HJB equation. Almulla et al. (2017) devise an approach for monotone MFGs based on monotone operators. Chassagneux et al. (2019) build a probabilistic, recursive scheme for certain classes of McKean-Vlasov-type MFGs.

To mitigate the curse of dimensionality, recent contributions have focused on importing mesh-free methods from machine learning, e.g. Carmona and Laurière (2021) and Ruthotto et al. (2020) who teach neural networks to approximate the theoretical solution, or Guo et al. (2019) who even train them to learn the (*a priori* unknown) dynamics of the system. See Lauriere (2021) for an extensive and detailed review on this topic.

2.3. Spectral methods for Hamilton-Jacobi-Bellman equations

Spectral methods have already extensively been used to solve Hamilton-Jacobi-Bellman equations. The earliest contributions in this direction, unsurprisingly stemming from the field of optimal control, are Vlassenbroeck and Van Dooren (1988) and Elnagar et al. (1995). They have since then been successfully applied in a number of practical applications. E.g. Ross and Karpenko (2012) retell the application of pseudospectral optimal control to trajectory-optimization at NASA, and show that between 2006 to 2020, it helped save $\sim 5,000\text{kg}$ of fuel, representing savings of about \$100M. More esoteric recent applications include Mashadi and Majidi (2014) about driving autonomous cars, Chen et al. (2021) about parking them, and Dastyar et al. (2022) who characterize optimal heat transfer between blood vessels.

Spectral and pseudospectral methods were introduced to economics by Judd (1992), who used the more general term "projection methods" (which include finite-element, spectral and pseudospectral approaches) which has since then at times been used as a synonym for collocation in economics. Judd presents an application to a discrete-time optimal growth problem, with or without stochasticity, where the Euler equation is discretized using Galerkin or collocation methods. Moreover, he performs fairly extensive stability analysis and provides an interpretation of approximation errors reflecting the behavior of boundedly rational agents.

Judd (1998) contains a chapter dedicated to spectral and pseudospectral ("projection") methods. It applies them mostly to discrete-time problems, but also to a continuous-time optimal growth problem using collocation both on the Euler and HJB characterization. However, they are not applied to problems with random income processes nor with borrowing constraints, and there is no tight link with a solution for the stationary distribution.

Another textbook on numerical methods for economics, Miranda and Fackler (2004) contains a whole chapter on continuous-time HJB equations. It stresses the advantages of continuous-time (no numerical integration for the expectation, first order conditions are easy to analytically) and presents three approaches to solving collocated HJB equations (value iteration, policy iteration, Newton / quasi-Newton iteration). Finally, it provides a large number of examples: optimal growth, renewable resource management, asset replacement, investment timing, optimal fish harvest, american options etc. However, these examples only collocate on a single dimension, do not feature a credit constraint

Among useful pedagogical contributions, we should also cite David R. Pugh's PyCollocation (Pugh, 2016), which builds a collocation toolbox and applies them to an optimal growth model, a Ramsey-Cass-Koopmans model, a Spence signaling problem and a symmetric auction, all of which are set in continuous-time (though they only collocate along a single dimension).

Since the late 1990s, collocation methods have been successfully applied to dynamic programming problems in economics in a number of areas. For simulating one-country representative agent mod-

els, they have been shown to be highly accurate and fast (Gaspar and Judd, 1997; Aruoba et al., 2006; Caldara et al., 2012). Brumm and Grill (2014) apply them to models with occasionally binding constraints by dynamically adding nodes around non-differentiabilities. Krueger and Kubler (2004) show that they can be used to simulate OLG models where agents live for 20–30 periods. Malin et al. (2011) apply a Smolyak method to solve a multi-country model with up to 10 countries and two state variables per country. However, most of these are set in discrete time, and none of them build a tight link with the solution for the stationary distribution.

This is somewhat surprising, as numerical procedures to solve for the distribution of agents in economies with heterogeneity have also seen methods with great similarity to spectral and pseudospectral approaches. These are largely inspired by earlier approaches which parametrized the cross-sectional distribution of agents, typically writing it as a sum of exponential distributions, as in Den Haan (1997), Reiter (2004). These methods usually compute and characterize moments of the distribution, which means one then has to solve non-linearly for the coefficients of the linear combination. Numerical tests suggest than relatively few moments can give a reasonably precise solution. Algan et al. (2008) develops an optimized simulation procedure to obtain these moments.

More recently, Winberry (2018) applies Algan et al.'s parametrization approach over two dimensions by using a tensorised basis over incomes and assets, and simulates the dynamics of the economy by perturbation on the moments of the cross-sectional distribution. Gordon (2020) presents a very similar approach, but complements it with a Smolyak method to interpolate using relatively fewer nodes. Note that, though similar in spirit, none of these approaches are strictly speaking spectral or pseudospectral methods, as they do not involve the solution of a PDE (in our case the Kolmogorov Forward equation, which describes the motion of agents in the economy) via a Galerkin or collocation approach. Relatedly, all of these examples are set in discrete time. To the best of our knowledge, no paper so far has tried to use pseudospectral methods to solve for the distribution of agents in a heterogeneous-agent economy.²

Finally, spectral and pseudospectral methods have seen some applications to more esoteric, non-HJB PDEs in economics: Nikooeinejad et al. (2016) use them to solve for open-loop Nash equilibria in nonlinear differential games, Tarkiainen and Tuomala (1999) characterize optimal, nonlinear two-dimensional income tax schedules and (Greenberg, 2003) uses them to price American options by casting the exercise boundary as a Stefan-like moving boundary problem.

The mathematical theory behind spectral methods applied to HJB equations is far more limited than for, e.g. finite differences. This largely reflects the fact that spectral and pseudospectral methods are designed for smooth problems, as the Gibbs phenomenon appears when there are discontinuities or non-differentiabilities in the solution, which means that the machinery of viscosity solutions cannot be used. In an important contribution, Tadmor (1990) describes a "spectral viscosity method", which means adding a high frequency vanishing regularization, and shows that this allows convergence to the true results while preserving spectral accuracy.

Lepsky (2000) and Mai (2022) put forward additional regularization approaches, and provide some results showing that, under appropriate conditions, they converge to the unique viscosity solution. See also Boyd (2010) for an interesting counterexample of a Burgers-like equation in which adding Legendre viscosity at singularity points fails.

²Probably because it doesn't work very well.

2.4. High-dimensional methods for Hamilton-Jacobi-Bellman equations

Although it doesn't use any of these methods directly, this Thesis tries to build a bridge with the literature on high-dimensional methods for HJB equations. A first part of this literature revolves around sparse grids (SGs) and adaptive sparse grids (ASGs), which are useful to implementing finite differences more economically, and have already been applied to economics. A second part concerns high-dimensional spectral & pseudospectral methods for HJB equations, mostly coming from the field of optimal control, which could be used to enrich the pseudospectral approach presented here to a higher-dimensional setting ($d \geq 4$). Finally, new developments coming notably from Machine Learning have tried to build mesh-free methods to solve high-dimensional dynamic optimization problems.

The mathematical literature on using sparse grids to solve Hamilton-Jacobi-Bellman equation is growing but already fairly extensive. Sparse grids save points compared to full (cartesian) grids by representing meshes as hierarchies and only retaining levels up to a certain influence level. Adaptive sparse grids do this anisotropically, i.e. they place different resolutions on different dimensions. This approach saves nodes and is especially powerful in higher dimensions, as it tames the curse of dimensionality by making the number of nodes needed to achieve a certain precision subexponential (the exact growth rate depending on the smoothness of the function).

These methods have been applied to HJB equations in the field of optimal control. For example, Bokanowski et al. (2013) use this method to implement a semi-Lagrangian scheme to solve non-linear HJB equations. Kang and Wilcox (2017) propose a method that discretizes the HJB on a sparse grid, and characterize the optimum as a boundary value problem, or characteristics equation, on each node, which can then efficiently be solved with a high degree of parallelization.

These methods have also found applications in economics starting with Brumm and Scheidegger (2017), who apply them to an international real business cycle model and to a multiproduct menu-cost problem, both of which are set in discrete time. Schaab and Zhang (2021) extend this contribution to continuous-time, allow for a finer treatment of boundary conditions and of differential operators, and build a useful code library to solve these models.

In a more focused contribution that is highly relevant to our work, Ahn (2019) builds a finite volume to solve for the distribution of agents in models with heterogeneity, show how they ensure mass conservation and positivity. As we will discuss extensively below, finite volume discretizations of the KF equation are closely related to finite difference discretizations of the coupled HJB-KFE system.

In a more mathematical treatment of the topic, Ruttscheidt (2018) notes that, as already pointed out by Hemker (2000), interpolation on sparse grids is generally not monotone. He shows that, even in the case of monotone, concave functions, sparse grid interpolation can be non-monotonic. On the numerical front, this means it could be useful to increase the precision of the grid wherever there are non-monotonicities. On the theoretical front, this means that monotonicity-based convergence results à la Barles and Souganidis (1991) cannot be applied. Brumm et al. (2021) admit "General results on the convergence of sparse finite-difference schemes in the context of viscosity solutions have remained elusive". Brumm et al. (2021) offer a pedagogical review of this fascinating body of work.

Although *vanilla* spectral methods already partially stave off the curse of dimensionality by requiring relatively few nodes for smooth problems, more advanced methods have been developed to apply these to high-dimensional settings. Kalise and Kunisch (2018) solve a continuous-time HJB equation using collocation and an iterative scheme, which makes the problem linear but involves computing high-dimensional integrals; by assuming that system dynamics are separable across dimensions, these reduce to products of one-dimensional integrals, which are much faster to compute. They then apply it to controls of systems with up to 14 dimensions. Dolgov et al. (2021) presents a more general method to reduce the effect of dimensionality by using a tensor-train polynomial approximation of the

solution, which allows for polynomial scaling with the dimension at a reduced precision cost. Authors apply it to non-linear systems with hundreds of variables.

High-dimensional collocation methods have also been introduced in economics. Krueger and Kubler (2004) and Malin et al. (2011) are two early contributions using this technique. Judd et al. (2014) adapt and extend the Smolyak sparse grid interpolation formula, and devise a derivative-free fixed-point iteration method to solve recursive problems. They apply it to a one-country and multi-country optimal growth problem, though they note that solving a 10-country model up to the second-level Smolyak approximation takes about 45 minutes in Matlab, but a 45 hours for the third level... Fernández-Villaverde and Levintal (2018) enrich this approach to models with "rare disasters", which require high-precision solution methods given the large non-linearities they entail, and present a Smolyak collocation method combined with a Newton method using analytic Jacobians, though their conclusion is that Taylor projections methods are most reliable. As most applications of collocation to HJB equations in economics, their approach is set in discrete time.

Recently, more orthogonal approaches based e.g. on mesh-free methods from machine learning have also been used to great effect to solve high-dimensional HJB equations. Sirignano and Spiliopoulos (2018) put forward a Deep Galerkin Method (DGM), in which a neural network is trained to satisfy the differential operator, initial condition and boundary condition; instead of using a mesh, this is done by randomly sampling time and space points, which allows them to solve problems with up to 200 dimensions. Han et al. (2017) similarly outline an analogy between reinforcement learning and backward stochastic differential equations (BSDEs), which they solve using a neural network and TensorFlow's highly optimized routines.

More specifically focussing on HJBs, Han et al. (2017) build a tight connection between the architecture of certain neural networks and HJB equations, which can then be used to solve them. Nakamura-Zimmerer et al. (2021) implement a similar strategy, notably by using the trained neural networks for adaptive data generation. They also present an application in which neural networks are used to solve a pseudospectrally discretized Burgers-type PDE.

It should be noted that, though they are very interesting, the convergence properties of most of these methods remain largely unknown. Even though there are now more and more applications of ML to economics-related optimization problems, it seems only Duarte (2018) does this in continuous time using PDE solution methods. Mesh-free methods are undoubtedly a promising area for future research in numerical methods for heterogeneous-agent models.

3. Prolegomenon: spectral & pseudospectral methods

When faced with a generic differential equation of the form

$$\mathcal{L}u = f , \quad (1)$$

defined over a domain Ω , where \mathcal{L} is some differential operator and f some right-hand-side function, it is often of great numerical interest to decompose the solution into a series of basis functions:

$$u = \sum_{n=0}^{\infty} a_n \phi_n \quad (2)$$

Finite element methods chose low-order polynomials defined over disjoint subdomains, so that they are zero over most of the domain, e.g. hat functions,. On the other hand, spectral methods choose global basis functions, e.g. Fourier series for periodic problems or Chebyshev polynomials for compactly-supported ones.

The idea is then to approximate the solution by computing a truncated series,

$$u_N := \sum_{n=0}^N \alpha_n \phi_n , \quad (3)$$

where the coefficients α_n are chosen to minimize, in some appropriate sense, the residual

$$\mathcal{R}(\alpha_1, \dots, \alpha_N) := \mathcal{L}u - f . \quad (4)$$

A natural avenue to do so is to define a set of test functions $\{\psi_n\}_{n \leq N}$ and to impose

$$\langle \mathcal{R}(\alpha_1, \dots, \alpha_N), \psi_n \rangle_\omega = 0 \quad \forall n \leq N \quad (5)$$

for some inner product weighted by the function ω :

$$\langle v, w \rangle_\omega := \int_{\Omega} v(x) w(x) \omega(x) dx \quad (6)$$

Submethods of the spectral family are then distinguished by their choice of test functions ψ_n and weighting function ω .

The Galerkin approach, which yields what is usually meant by "spectral method" in the strict sense, is to take $\psi_n = \phi_n$. When the initial basis functions were well-chosen, typically because they form an orthogonal basis of the considered function space, this can work remarkably well. The implementation then requires computing the interaction terms $\langle \mathcal{A}\phi_i, \psi_j \rangle_\omega$, which can involve fairly complicated integrals which are hard to compute efficiently on a computer, though it is then very precise.

The "pseudospectral" or "collocation" approach is to select a number of points x_1, \dots, x_N and to choose Dirac delta functions $\psi_n = \delta_{x_n}$. In effect, (5) then reduces to imposing (1) at each point, i.e.

$$\mathcal{R}(\alpha_1, \dots, \alpha_N)(x_n) = 0 \quad \forall n \leq N \quad \iff \quad \mathcal{L}u(x_n) = f(x_n) \quad n \leq N \quad (7)$$

This can be considerably easier to implement numerically, especially for some of the iterative schemes we will be working on, which is why the rest of this Master Thesis focusses on pseudospectral solution methods, even though the word "spectral" is used at times as a loose synonym.

The key insight that makes the pseudospectral approach much easier to implement is that, instead of working with the coefficients α_n in the series expansion of u_N , one can work with the values of u_N at $N+1$ nodes, as any N -th order polynomial is uniquely defined by its values at $N+1$ distinct points. One can then obtain the coefficients from these points, or, what is often more relevant in practice, values of the solution at off-nodal points by (exact!) polynomial interpolation.

Although the $N+1$ nodes could in theory be arbitrary, we seek to minimize errors in the polynomial interpolation, and the Runge phenomenon means that these have to be chosen very carefully. The Chebyshev nodes over $[-1, 1]$, given by the roots of the Chebyshev polynomials of the first kind, are the best choice:

$$x_n = \cos\left(\frac{2n-1}{2N}\pi\right) \quad n = 1, \dots, N \quad (8)$$

To these, we will always add the end-points $\{-1, 1\}$, so that we are, strictly speaking, working with Chebyshev-Lobatto nodes. As we will be working on intervals other than $[-1, 1]$, we simply have to linearly rescale all nodes. Although the barycentric Lagrange interpolation formula shows that polynomial interpolation from a given set of nodes can always be reduced to a weighted sum of nodal

values (see [Berrut and Trefethen \(2004\)](#)), in this case there are analytical formulae for the weights which will make the whole procedure faster and less subject to round-off error.

The barycentric Lagrange formula also shows that the derivative at some given point of a polynomial of given values at given nodes can be expressed as a weighted sum of these nodal values; again, in the case of Chebyshev nodes, there are analytical formulae to do this, and especially to compute the derivative of the interpolating polynomial *at the collocation nodes*.

These linear operations can then be stacked into a Chebyshev differentiation matrix, which operates on the nodal values of a polynomial and returns the derivatives at these nodes. In notation, if $u_N = P$ for some polynomial P such that $P(x_n) = u_n$, one can assemble a matrix D such that:

$$D \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} u'_1 \\ \vdots \\ u'_N \end{bmatrix} \quad (9)$$

where $u'_n = P'(x_n)$ are the nodal derivatives. Higher-order derivatives can be obtained using the powers of this differentiation matrix, or via additional analytic results. The precise formulae are fairly involved and usually expressed recursively, but luckily the package `DmSuite`, available in Matlab and Python, does this for us.

Solving a differential equation using a pseudospectral method thus reduces to choosing N Chebyshev collocation nodes, discretizing the differential operator \mathcal{L} into a polynomial equivalent L by an appropriate combination of differentiation matrices of different orders, computing the right-hand side $f_n = f(x_n)$ and then solving an $N \times N$ linear system of the form:

$$L \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} \quad (10)$$

The polynomial interpolating the values u_1, \dots, u_N at the nodes x_1, \dots, x_N will then be the pseudospectral solution.

Boundary conditions, which we have set aside so far, are generally imposed using boundary-bordering, i.e. replacing the appropriate columns of L with a linear operation corresponding to the condition (e.g. an identity matrix for Dirichlet conditions, a differentiation matrix for Neuman conditions, a combination of both for Robin conditions etc.) Extending this method to PDEs means tensorizing the implicit polynomial basis, which translates into relatively straightforward Kronecker-broadcasting operations on matrices.

We have tried to keep the exposition brief, but the curious reader is referred to [Kopriva \(2009\)](#) and especially [Boyd \(2001\)](#) for an excellent introduction to spectral and pseudospectral methods, and to [Trefethen \(2000\)](#) for an elementary guide on the numerical implementation of pseudospectral methods.

4. Spectral method for a Huggett model with two incomes

4.1. Description of the two-income Huggett model

We start our application of spectral methods to a standard Huggett economy by outlining the main features of the model and the core partial differential equations it generates. See Achdou et al. (2021) for a far more extensive exposition.

In this economy, agents have an income $z_t \in \{z_1, z_2\}$, with $z_1 < z_2$, which follows a continuous-time Markov process with transition rates λ_1, λ_2 . They can save or borrow to hold assets a_t , but cannot borrow below a credit constraint $a > \underline{a}$. The interest rate on savings and borrowings is exogenous and fixed at r . This means that their wealth follows the law of motion:

$$\dot{a}_t = z_t + a_t r - c_t \quad (11)$$

where c_t denotes flow consumption. Agents optimize over the discounted utility from these future consumption flows. Instantaneous utility is given by a CRRA function $u(c) = c^{1-\gamma}/1-\gamma$, with $\gamma > 0$. Finally, agents use a fixed discount rate ρ , with $r < \rho$, and thus maximize over:

$$\mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t) dt \quad (12)$$

Two key components characterize equilibrium in this economy: the optimal behavior of each individual agent and the stationary distribution of all agents in the economy. Each component gives rise to a partial differential equation, which form the basis of any numerical approach, and to which we now turn.

The value function of each agent solves a Hamilton-Jacobi-Bellman (HJB) equation of the form:

$$\rho v_j(a) = \max_c u(c) + v'_j(a)(z_j + ra - c) + \lambda_j(v_{-j}(a) - v_j(a)) \quad \forall a \in [\underline{a}, +\infty], \forall j \in \{1, 2\} \quad (13)$$

See e.g. Stokey (2008) for a derivation. Note that here, c is defined implicitly as it is a function of v : it is given by the first order condition $u'(c_j(a)) = v'_j(a) \iff c_j(a) = u'^{-1}(v'_j(a))$.

The borrowing constraint $\underline{a} < a$ adds a boundary condition to the HJB equation. As the first-order condition still holds at \underline{a} , it can be rewritten as:

$$\begin{aligned} s_j(\underline{a}) \geq 0 &\iff z_j + ra - c_j(a) \geq 0 \\ &\iff z_j + ra \geq u'^{-1}(v'_j(a)) \\ &\iff v'_j(a) \geq u(z_j + ra) \quad j \in \{1, 2\} \end{aligned} \quad (14)$$

In turn, given individual savings function, the stationary distribution of agents g_1, g_2 solves the following Kolmogorov Forward Equation (KFE):

$$0 = -\frac{d}{da} [s_j(a)g_j(a)] - \lambda_j g_j(a) + \lambda_{-j} g_{-j}(a) \quad \forall a \in [\underline{a}, +\infty], \forall j \in \{1, 2\} \quad (15)$$

The first term represents the asset drift, which is given by the mass of agents at each point times their savings, while the second term represent the flow of agents switching from their current income j to the other state, and the third term the flow of agents switching from the other state to income j .

The KFE does not require any additional boundary conditions, but still some issues will arise at

the boundaries. Indeed, as is shown analytically by Achdou et al. (2021) (see Corollary 1), low-income agents will hit the borrowing in finite time with non-zero probability (i.e. given a sufficiently long, but finite, realizations of the bad income state). This means that a Dirac mass of agents will accumulate at (z_1, \underline{a}) , and therefore also at (z_2, \underline{a}) .

The distribution of agents thus does not admit a density at those two points, which means the KFE equation has to be understood in a weak, measure-valued sense. More precisely, solutions to the KF equation should be thought of as a pair of measures μ_1, μ_2 that are linear combinations of a density g_j over a Lebesgue measure \mathcal{L} and a Dirac function $\delta_{\underline{a}}$: $d\mu_j(a) = g_j(a)d\mathcal{L}(a) + m_i\delta_{\underline{a}}$. The weak formulation of the KF equation then states that, for any $\phi \in C^\infty([\underline{a}, \infty]) \subset L^2([\underline{a}, \infty])$, the measures should verify, for $j \in \{1, 2\}$:

$$0 = \int_{\underline{a}}^{\infty} [\varphi'_j(a)s_j(a) + \lambda_j(\varphi_{-i}(a) - \varphi_j(a))] g_j(a)d\mathcal{L}(a) + [\varphi'_j(\underline{a})s_j(\underline{a}) + \lambda_j(\varphi_{-j}(\underline{a}) - \varphi_j(\underline{a}))] m_j \quad (16)$$

Although the Dirac source mass will be a source of complications afterwards, the weak form is of limited interest for our collocation-based approach, so we will keep discussing the primary formulation.

For greater comparability, we take all economic and numerical parameters over from Achdou et al. (2021), which are synthesized in Table 1.

4.2. Solving for optimal savings: the Hamilton-Jacobi-Bellman equation

4.2.1. Numerical procedure

Solving the Hamilton-Jacobi-Bellman equation using a pseudospectral method is reasonably straightforward and works remarkably well.

We start by fixing N nodes over $[\underline{a}, \bar{a}]$, for some high enough upper bound \bar{a} , which we will denote a_1, \dots, a_N . As explained in Section 3, nodes have to be chosen carefully to avoid the Runge phenomenon: we thus choose rescaled Chebyshev nodes. We can then approximate the value function v_j by its value at each node $v_{j,n} := v_j(a_n)$. We will write the vector of values for a given income as $\mathbf{v}_j = \{v_{j,1}, \dots, v_{j,N}\}$ and the stacked vector for both incomes as $\mathbf{v} = [\mathbf{v}_1 | \mathbf{v}_2] = \{v_{1,1}, \dots, v_{1,N}, v_{2,1}, \dots, v_{2,N}\}$. We are thus looking for a $(N - 1)$ -th degree polynomial such that the HJB equation holds at each node:

$$\rho v_{j,n} = \max_{c_n} u(c_n) + v'_{j,n}(z_j + ra_n - c_n) + \lambda_j(v_{-j,n} - v_{j,n}) \quad \forall n \in \llbracket 1, N \rrbracket, \forall j \in \{1, 2\} \quad (17)$$

The derivative of the value function at each node appears as a key quantity to assess whether the HJB equation holds locally. Recall that the derivative at each node n is given by a linear combination of the values at all nodes $v_{j,1}, \dots, v_{j,N}$ and that these linear operators can be stacked into a $N \times N$ differentiation matrix, which we will denote D . Denoting D 's n -th row D_n , this writes:

$$\rho v_{j,n} = \max_{c_n} u(c_n) + (D_n \mathbf{v}_j)(z_j + ra_n - c_n) + \lambda_j(v_{-j,n} - v_{j,n}) \quad \forall n \in \llbracket 1, N \rrbracket, \forall j \in \{1, 2\} \quad (18)$$

Even though the derivative has now been turned into a linear operation, the maximum operator makes this equation highly non-linear and difficult to solve. We therefore obtain a solution via an iterative scheme, following the approach adopted by Achdou et al. (2021). Other works have called this solution method "successive approximation", because each update solves the generalized Hamilton-Jacobi-Belman (GHJB) equation, where one solves for the value function while keeping the control law fixed (which makes the equation linear) and then updates the control law. However, one could envisage an explicit, implicit or semi-implicit approach to these iterations, a distinction which we will dwell on briefly.

An explicit scheme would start from an "initial guess" $\mathbf{v}^0 = [v_1^0 | v_2^0]$ and "update" each $v_{j,n}^k$ via:

$$\frac{v_{j,n}^{k+1} - v_{j,n}^k}{\Delta} + \rho v_{j,n}^k = u(c_{j,n}^k) + v_{j,n}^k' (z_j + ra_n - c_{j,n}^k) + \lambda_j (v_{-j,n}^k - v_{j,n}^k) \quad \forall n, \forall j \quad (19)$$

with $c_{j,n}^k = u'^{-1}(v_{j,n}^k')$ and $\Delta > 0$. Recall that j indexes income states, n indexes asset nodes and k indexes iterations. This explicit scheme has a natural interpretation: it reflects the value of an agent who experiences utility \mathbf{v}^0 in an initial period and then moves back in time in steps of size Δ , setting his consumption according to the first-order-condition stemming from the *current* value function.

To use larger time steps and thus speed up convergence, we can use an implicit scheme, which moves backward in time by solving for a value function consistent with the steps back in time. This means updating v_j^k via:

$$\frac{v_{j,n}^{k+1} - v_{j,n}^k}{\Delta} + \rho v_{j,n}^{k+1} = u(c_{j,n}^k) + v_{j,n}^{k+1'} (z_j + ra_n - c_{j,n}^k) + \lambda_j (v_{-j,n}^{k+1} - v_{j,n}^{k+1}) \quad \forall n, \forall j \quad (20)$$

where, as before, $c_{j,n}^k = u'^{-1}(v_{j,n}^k')$, but with $\Delta \gg 0$. Note that strictly speaking this scheme is semi-implicit, as we are still setting consumption according to the first-order-condition stemming from the *current* value function. A fully implicit scheme, which would essentially be a Newton iteration, should converge in less steps, but each step is much harder to solve for so that it does not bring any material benefits.

As we have to solve jointly for the high and low income, we have to stack these N coupled equations into a $(2N) \times (2N)$ matrix equation. In matrix notation, the implicit update then writes:

$$\frac{1}{\Delta} (\mathbf{v}^{k+1} - \mathbf{v}^k) + \rho \mathbf{v}^{k+1} = \mathbf{u}^k + \mathbf{s}^k \mathbf{D} \mathbf{v}^{k+1} + \mathbf{G} \mathbf{v}^{k+1} \quad (21)$$

where, using the Kronecker product \otimes , we have defined:

$$\mathbf{u}^k := [u(c_{1,1}^k), \dots, u(c_{1,N}^k), u(c_{2,1}^k), \dots, u(c_{2,N}^k)]' \quad (22)$$

$$\mathbf{s}^k := [z_1 + ra_1 - c_{1,1}^k, \dots, z_1 + ra_N - c_{1,N}^k, z_2 + ra_1 - c_{2,1}^k, \dots, z_2 + ra_N - c_{2,N}^k]' \quad (23)$$

$$\mathbf{D} := I_2 \otimes D = \left[\begin{array}{c|c} D & 0 \\ \hline 0 & D \end{array} \right] \quad (24)$$

$$\mathbf{G} := G \otimes I_N = \left[\begin{array}{cccccc} -\lambda_1 & \cdots & 0 & \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -\lambda_1 & 0 & \cdots & \lambda_1 \\ \lambda_2 & \cdots & 0 & -\lambda_2 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_2 & 0 & \cdots & -\lambda_2 \end{array} \right] \quad (25)$$

As it will be useful to discuss similarities with the upwind finite differences approach, we further

define $\mathbf{A}^k := \mathbf{s}^k \mathbf{D} + \mathbf{G}$. Finally, (21) can be rearranged into:

$$\mathbf{B}^k := \left(\frac{1}{\Delta} + \rho \right) \mathbf{I} - \mathbf{A}^k, \quad \mathbf{b}^k := \mathbf{u}^k + \frac{1}{\Delta} \mathbf{v}^k \quad (26)$$

$$\mathbf{B}\mathbf{v}^{k+1} = \mathbf{b}^k \quad (27)$$

where $\mathbf{I} := I_2 \otimes I_N = I_{2N}$. Note that (27), which is the updating step in the implicit scheme, now becomes a simple matrix inversion problem.

Readers familiar with Achdou et al. (2021)'s upwind finite difference approach will note the great apparent similarity with the spectral approach, with a few notable exceptions.

When using upwind finite differences, the direction in which the difference is computed, i.e. either *backward* or *forward*, depends on the drift of the state variable (in our case savings, s_t , which are the drift of assets a_t). This means that, at each iteration, the differentiation matrix \mathbf{A} has to be reassembled depending on each $s_{j,n}^k = z_j + ra_n - c_{j,n}^k$. In the pseudospectral approach, the differentiation matrix D is *exact* for $(N - 1)$ -th degree polynomials, which means it does not depend on savings and thus doesn't have to be reassembled at each round.

That's why the matrix \mathbf{D} doesn't feature a k superscript. We can therefore assemble it outside of the implicit scheme loop, and only have to compute \mathbf{s}^k (a matrix-vector product followed by a factorized algebraic function) and \mathbf{B}^k (a vector-matrix product) at each round. Even though spectral matrices are not sparse, the spectral method is thus not much slower than the finite difference method for the same number of nodes (see Figure 33), even though for a given number of nodes it has far higher precision (see Figure 32). E.g., it takes about $N \approx 200$ spectral nodes to be as slow as the finite difference method with $I \approx 500$ nodes.

On the other hand, the pseudospectral matrix \mathbf{A} does not have a natural "Markov process generator" interpretation. Indeed, as Achdou et al. note, a correctly assembled matrix upwind finite differences matrix \mathbf{A} has rows that sum to zero, non-positive diagonals and non-negative off-diagonals: this means that it can be viewed as the transition matrix of a continuous-time, discrete-space Markov process, which approximates the true continuous-time, continuous-space Markov process of income and wealth. This feature can then be exploited very elegantly when solving for the stationary distribution, as the stationary distribution is simply the left-eigenvector of the generator associated with the eigenvalue zero. One thus only has to transpose the generator and find its (right-)eigenvector associated with eigenvalue zero.³

In the pseudospectral approach, $\mathbf{A}^k := \mathbf{s}^k \mathbf{D} + \mathbf{G}$ does not have any of these properties, even though \mathbf{G} clearly does. This reflects the nature of the polynomial differentiation matrix D , which is not tridiagonal nor even sparse (in fact, all elements are strictly non-zero) and has a fairly complex sign structure. Figure 1, where blue indicates negative and red positive numbers, shows the different matrix structures quite well. As polynomial differentiation is a global operation, unlike a finite difference which is inherently local, it does not allow for a Markov process transition rate matrix interpretation.

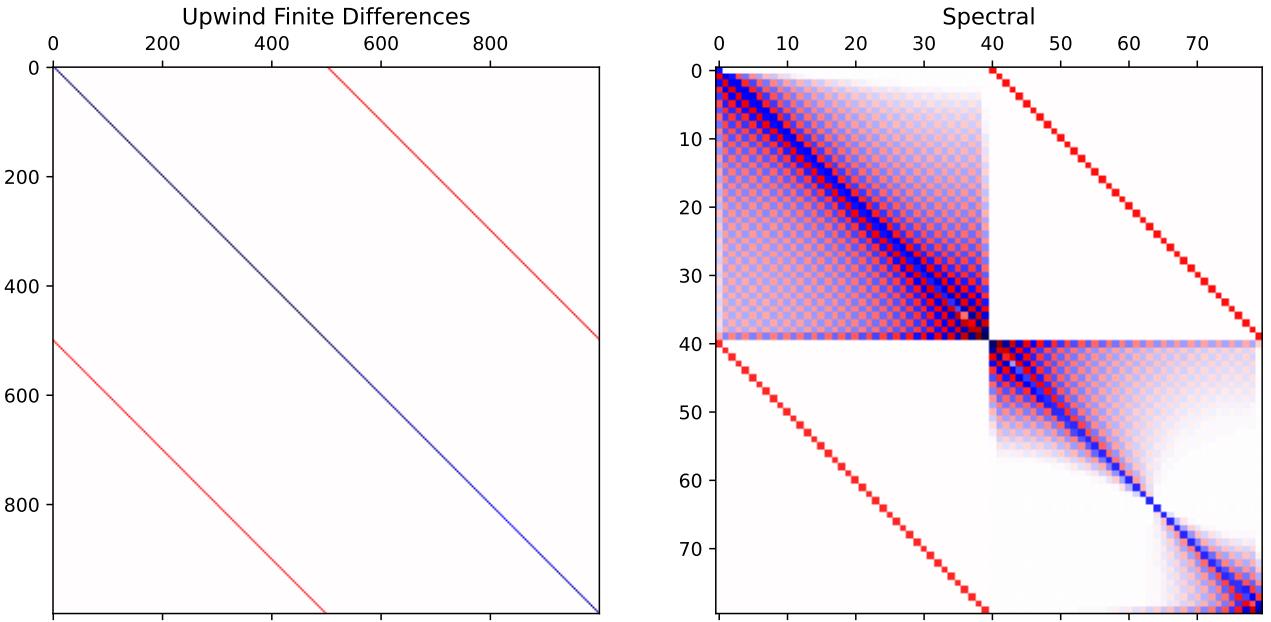
So far, we have not discussed the treatment of boundary conditions in the HJB equation. In the upwind finite difference world, this involves differences with virtual "ghost nodes", which works well but is somewhat involved. In the spectral world, imposing boundary conditions is very straightforward.

We know from Proposition 1 in Achdou et al. (2021)⁴ that the constraint only binds low-income agents at \underline{a} , i.e. at (z_1, \underline{a}) . Then, (14) simply reduces to $v'_1(a) = u'(z_1 + r\underline{a})$. As the derivative of v_1

³Although Achdou et al. don't do it, this can be further sped up by using sparse matrix routines that compute the lowest-magnitude eigenvalue for large matrices.

⁴And also from common sense.

Figure 1: Hamilton Jacobi Bellman Equation - Matrix \mathbf{A} in Different Approaches



at node 1 is given by $v'_{1,1} = D_1 v_1$, we simply have to set:

$$D_1 v_1 = u'(z_1 + r\underline{a}) \quad (28)$$

This is achieved via boundary-bordering, i.e. simply replacing \mathbf{B}^k 's first row by D_1 and \mathbf{b}^k 's first element by $u'(z_1 + r\underline{a})$. We do not have to do any cumbersome modification on differentiation matrices, savings etc.

As an initial guess, the simplest approach is to use the value of standing still, i.e. never saving anything and consuming current income plus (or minus) interest on assets forever:

$$v_{j,n}^0 = \int_0^\infty e^{-\rho t} u(z_j + r a_n) dx = \frac{u(z_j + r a_n)}{\rho} \quad (29)$$

Finally, we have to choose an updating step Δ . As noted above, the implicit scheme, in theory, allows for arbitrarily large time steps. In practice, the boundary constraint sometimes imposes sharp corrections towards the edges, which can lead to chaotic polynomial derivatives and thus break convergence. A similar phenomenon happens when using finite differences with very large time steps. These excessive corrections are larger (as they are more localized) when the number of nodes N is large; and they are smaller when the time step Δ is smaller.

Trial & error has shown that the best approach is to use time steps that are initially small and that increase at each iteration. Once the boundary constraint has been "smoothed out" of the initial guess after a few small-step iterations, the spectral method allows for very large time steps. We have thus found that using time steps of $\Delta_k = 0.01 \times e^k$ produces remarkably fast and robust convergences, usually requiring only 10 – 16 iterations.

To synthesize, our algorithm to find a spectral solution to the HJB equation takes the form:

1. Start with an initial guess $\mathbf{v}^0 = (v_{1,1}, \dots, v_{1,N}, v_{2,1}, \dots, v_{2,N})$ given by (29).

2. Compute \mathbf{c}^k using $c_{j,n}^k = u'^{-1}(v_{j,n}^{k'})$, where the $v_{j,n}^{k'}$ are obtained as $\mathbf{v}^{k'} = \mathbf{D}\mathbf{v}^k$.
3. Solve for \mathbf{v}^{k+1} using (27), while imposing (28) via boundary-bordering.
4. If $\|\mathbf{v}^{k+1} - \mathbf{v}^k\|_\infty < \varepsilon$, exit the iteration; otherwise, return to step 2.

4.2.2. Numerical results

We now turn to results from this numerical approach. To showcase the power of the spectral method, we use an intentionally low number of nodes of $N = 30$. Throughout, we compare results to a finite difference solution using $I = 500$ nodes, while all other parameters are kept identical (see Table 1). Solution of the HJB equation takes 38.2 ms for finite differences and 6.7 ms for the spectral method.

Figure 2: Value Functions (and Credit Constraint)

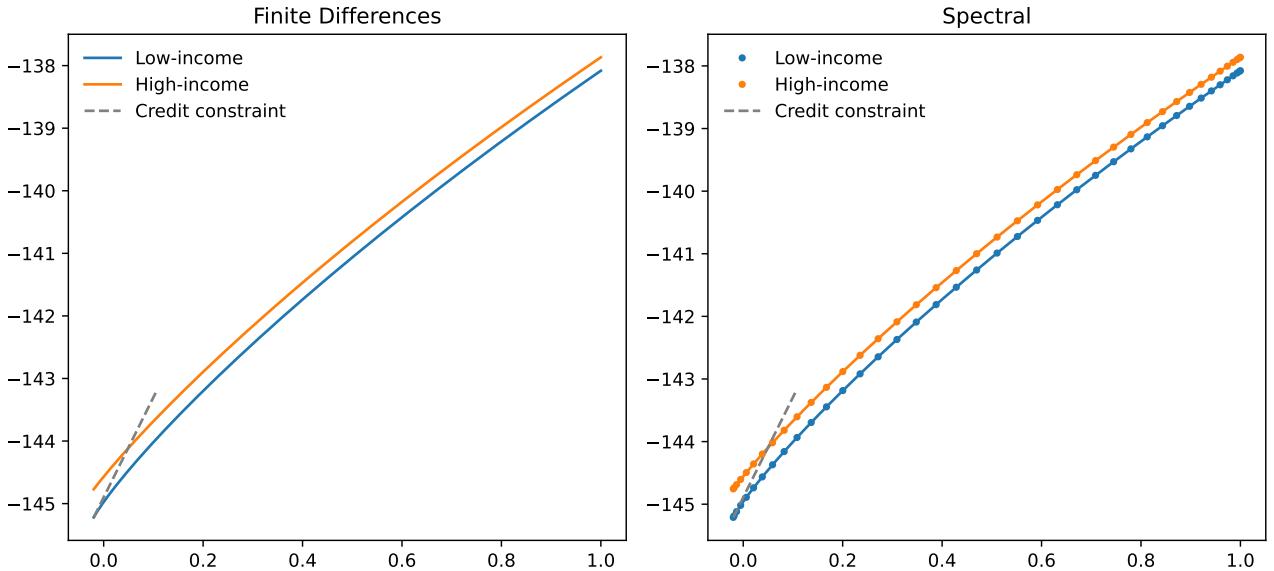


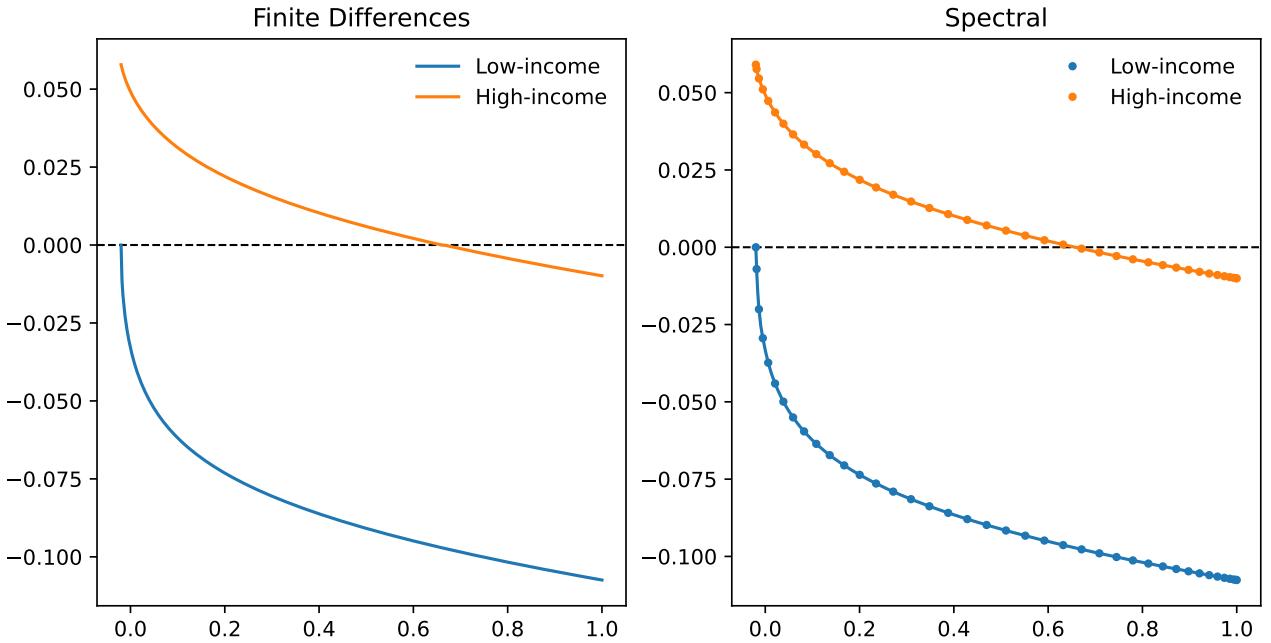
Figure 2 shows the value function obtained via finite differences (left) and via the spectral method (right). On the right panel, dots represent the value at the asset nodes, while the line shows the corresponding interpolating polynomial. Again, it should be stressed that the values at the nodes, although they are the output of the numerical procedure, are only a *representation* of the (tentative) solution, which *is the interpolating polynomial*. In plain English, *the line is the spectral solution*.

Figure 2 additionally shows the credit constraint, via a grey dashed line of slope $u'(z_1 + r\underline{a})$. The boundary constraint imposes that this slope be lower than or equal to the slope of the value functions at \underline{a} . We can see that it is clearly lower for the high-income value function, which means that high income types are not constrained (indeed, they have positive savings at \underline{a}). It is however equal to the slope of the low-income value function, which means that low income types are constrained. To the naked eye, this constraint seems to be respected in either approach, and numerical computations confirm this is the case. Thus our boundary-bordering approach to imposing the boundary constraint produces the desired results.

Savings functions associated with these value functions are shown in Figure 3. In the right panel, lines show the interpolating polynomial between nodal savings. Strictly speaking, the correct way of computing off-node savings would be to (i) compute derivatives of v_j at all nodes using D , (ii) compute the derivative of v_j at the relevant off-node point by interpolating the polynomial of nodal

derivatives (which yields the exact off-node derivative as the derivative is also a polynomial represented by the nodal derivatives) and (iii) compute savings at this point using the first-order condition. As the difference between both approaches is visually indiscernible, we choose the easy route.

Figure 3: Savings Functions



Readers that still had any doubts can see that savings obtained via the spectral method do indeed respect the credit constraint, as low-income savings converge to zero as $a \rightarrow \underline{a}$. We can also note that savings become negative for both incomes beyond a certain threshold (around 0.45) and that for high values of a both savings functions appear near-linear, as suggested by Achdou et al. (2021)'s Proposition 2.

More interesting than comparing results with the naked eye, we can also inspect the difference between the finite difference solution and the spectral solution (interpolated to finite difference nodes). Results are presented in Figure 4. The value function show a relatively small but largely monotonic difference, which is larger towards the asset boundary: it reflects the different way in which the boundary condition is imposed, as with finite difference methods the boundary condition involves differences with ghost nodes which are not nodal values, while in the spectral approach all derivatives are made using actual nodal values.

The savings functions also show an interesting superposition of errors. The medium-frequency oscillations come from the spectral method, and reflect the interpolation error we make when approximating a smooth function with a polynomial. The jump in the high-income savings around 0.45 comes from upwind finite differences, as it corresponds to the moment where savings become negative and we switch from forward to backward differences in computing savings. Finally, the large error at \underline{a} comes from both: given the high local curvature, the spectral method produces a curvature that is too low, while the finite difference method features an excessively sharp cutoff in savings at the credit constraint (see Figure 5).

All of these errors become smaller and smaller with increasing N and increasing I . In particular, for the spectral method, we purposely show an implementation with relatively few nodes ($N = 30$) to discuss its strengths and weaknesses. For a number of nodes that has about the same execution time as the finite difference method, i.e. $N \approx 200$, all of the clearly spectral errors become imperceptibly small.

Figure 4: Differences in Spectral vs Finite Differences Functions

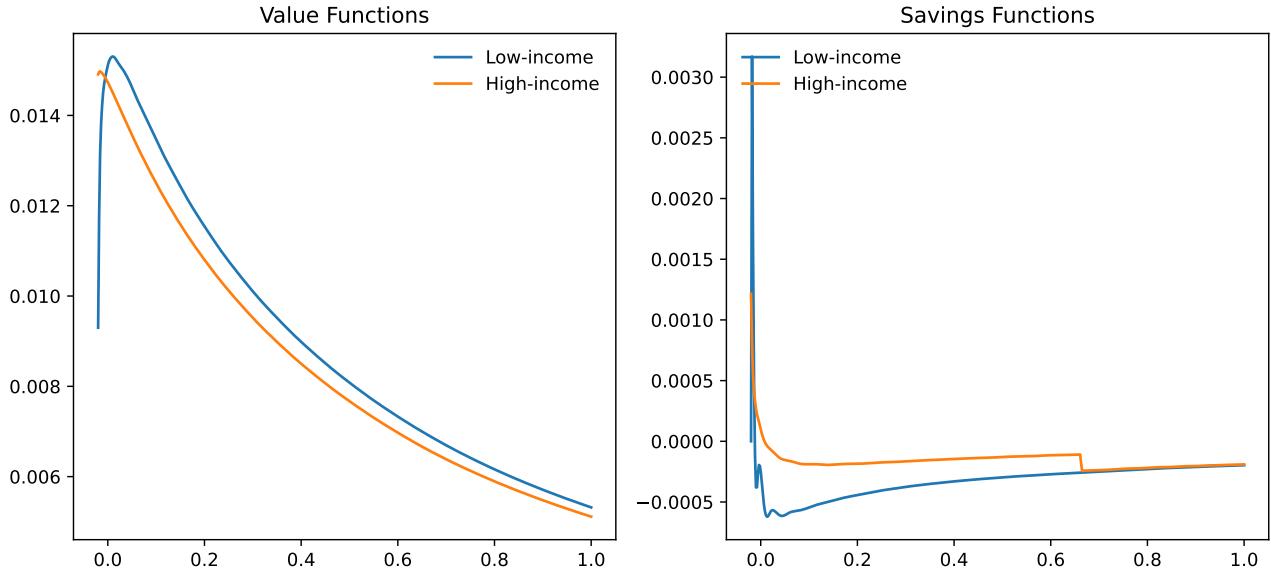


Figure 5: Value Functions & Saving Functions - Zoom on the Credit Constraint

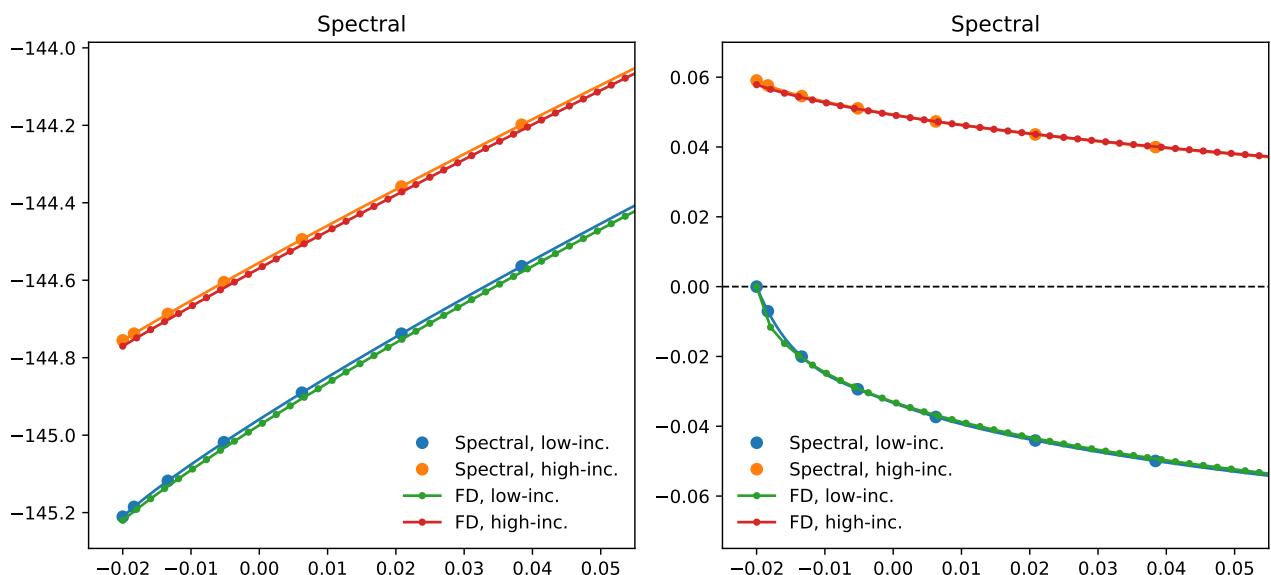
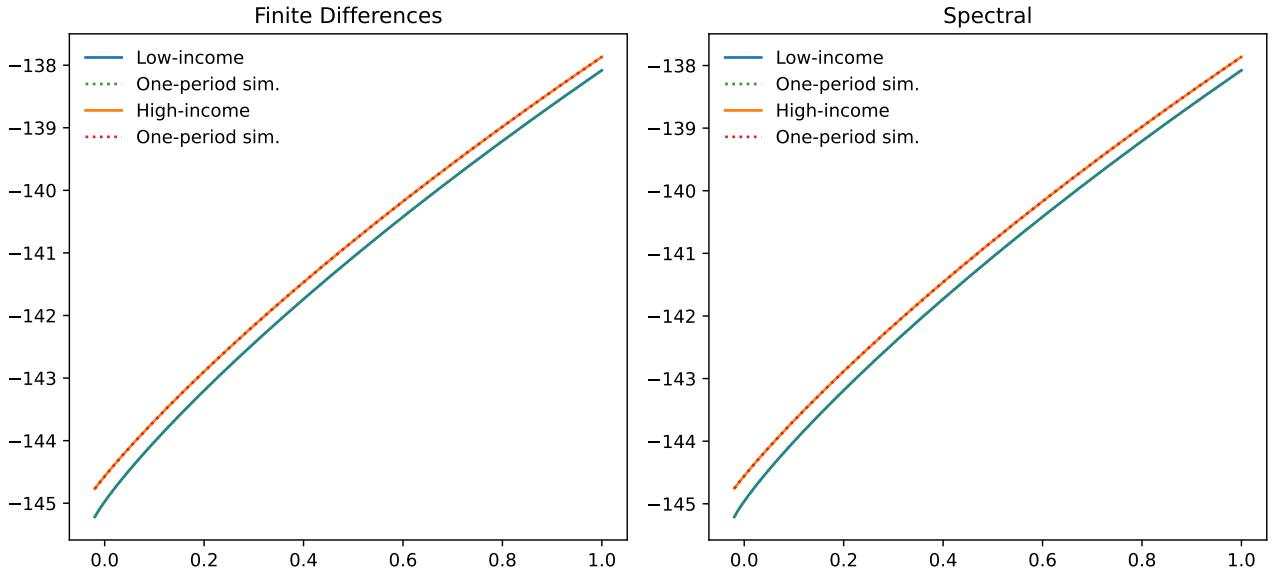


Figure 6: Value Function Consistency Check - One-period Simulation



As a last test of the quality of our numerical results for the HJB equation, we can run consistency checks on the value function. The idea is to simulate the behavior of an agent for a small time period Δ_t , and to see whether his expected utility is consistent with the value functions we have obtained. More precisely, for all initial positions $(z_0, a_0) \in \left\{ (z_j, a_n) \right\}_{j,n \in 1,2 \times [1,I]}$, in the (far denser) finite difference grid, we:

1. Compute savings s_0 at (a_0, z_0) via polynomial interpolation of the savings function.
2. Compute next-period assets as $a_1 = a_0 + \Delta_t * s_0$.
3. Draw randomly the next-period income z_1 with probabilities $[1 - \Delta_t \lambda_1, \Delta_t \lambda_1]$ or $[\Delta_t \lambda_2, 1 - \Delta_t \lambda_2]$ (recall that Δ_t is small, so we can use the first-order approximation of transition probabilities).
4. Compute the next-period value v_1 at (a_1, z_1) via polynomial interpolation of the value function.
5. Obtain the corresponding starting-period value $v_0 = \Delta_t u(c_0) + e^{-\rho \Delta_t} v_1$.

If the value function is consistent, the expectation of v_0 should be equal to the value function interpolated at (a_0, z_0) . We can thus repeat this simulation 100 times and compute to check this empirically. Repeating this for many different starting conditions, e.g. all values on the finite difference grid, gives us an idea of the consistency errors in our value functions. To compare results, we implement the same procedure for the finite differences solution, albeit using linear interpolation for savings and value functions.

Of course, an easy way to improve the precision of this check is, instead of simulating 100 draws of the income process, to compute the next-period value conditional on each realization and then the expected value using the transition probabilities. Results for the simulation, which are essentially the same save for the simulation noise (which is intentional, as using 10.000 draws takes virtually no additional time), are shown in Figure 6. Results for the expectation-based consistency checks are shown in Figure C.1.

As the difference is extremely small, errors are shown in Figure C.2. Clearly, errors in both cases are dominated by the simulation and expectation error, reflecting the error in the first-order

approximation embedded in the continuous-time transition kernel. It is then reassuring to see that the difference between the spectral and the finite difference solution is extremely small, which shows that they must both be reasonably close to the correct solution (or, more pessimistically, that they are both equally far).

4.3. Solving for the stationary distribution: Kolmogorov Forward Equation

While the spectral method works remarkably well to solve the Hamilton-Jacobi-Bellman equation, it is not very well-suited to solving the Kolmogorov Forward Equation, and will require some adaptations. The overriding issue is the Dirac mass of low-income agents at the bottom credit constraint, which means that the stationary distribution does not admit a density at \underline{a} .

Whereas the finite difference approach approximates the stationary density via a continuous-time, discrete-state Markov process, the spectral method works only on (nodal representations of) global functions. If the solution features a Dirac mass at some point, its mass will go to the nearest node in a finite difference method, while it will show up as a singularity in any spectral approach.

4.3.1. "Not because they are easy, ..."

One of the major strengths of upwind finite differences is its ability to easily solve the Kolmogorov Forward Equation. Recall that the Kolmogorov Forward Equation writes:

$$0 = -\frac{d}{da} [s_j(a)g_j(a)] - \lambda_j g_j(a) + \lambda_{-j} g_{-j}(a) \quad \forall a \in [\underline{a}, +\infty], \forall j \in \{1, 2\} \quad (30)$$

The first term represents the asset drift, which is given by the mass of agents at each point times their savings, while the second term represents the mass of agents switching from income to another. The similarity with the HJB equation is not fortuitous: the linear operator that appears in the KFE equation, \mathcal{A}^* , is the adjoint of the linear operator HJB equation, \mathcal{A} , which in turn is simply the generator of the two-dimensional income-assets Markov process.

In the finite difference approach, this can be exploited very elegantly because, if the differentiation matrix \mathbf{A}_{FD} , which is the discretization of the generator operator \mathcal{A} , is deftly assembled, its transpose \mathbf{A}_{FD}^T will also be a correct discretization of the adjoint operator \mathcal{A}^* .⁵ A correctly assembled finite differentiation matrix, notably paying close attention to the boundary conditions, has precisely these properties.

To compute a solution to the Kolmogorov Forward Equation via upwind finite differences, one simply has to find a solution to $\mathbf{A}_{FD}^T \mathbf{g}_{FD} = 0$, i.e. transpose a matrix and find its eigenvector associated with the eigenvalue 0 (which exists in virtue of the Perron–Frobenius theorem). This can be done very efficiently via sparse matrix routines (although Achdou et al. adopt a slightly more complex approach). The eigenvector then has to be rescaled so that it integrates to 1 and indeed represents a distribution.

This extremely elegant and efficient approach does not work under the spectral approach, and the reasons for this are instructive. Setting aside the more obvious problem stemming from the Dirac mass at \underline{a} , the transposition of the spectral differentiation matrix \mathbf{A} will not yield a usable discretization of the adjoint of \mathcal{A} . Recall that $\mathbf{A} = \mathbf{sD} + \mathbf{D}$: \mathbf{D}^T is obviously a correct discretization of the income-switching component of \mathcal{A}^* (in fact, the same \mathbf{D}^T gets added to the finite differentiation matrix \mathbf{A}_{FD}), so the problem must come from \mathbf{D} .

Indeed, we cannot use a simple transposition because spectral differentiation matrices are not skew-symmetric, i.e. $D^T \neq D$, which shows that polynomial differentiation is not a skew-adjoint

⁵Recall that adjointness is a generalization of transposition to infinite-dimensional linear operators.

operation. Under the appropriate boundary conditions, differentiation is skew-adjoint, i.e. its adjoint equals its opposite. Taking the simplest case of $f \in H = \{f \in L^2[0, 1] \text{ with } f(0) = f(1) = 0\}$, the integration by parts formula yields: $\forall g \in H, \int_0^1 f'g = -\int_0^1 fg'$.

Although upwind finite differentiation matrices are not skew-symmetric either, both because they are a first-order approximation and because the differentiation direction will change depending on savings, their transpose still operates correctly on the transpose of the drift. As polynomial differentiation matrices are exact and thus do not change based on the vector on which they operate, there is no hope of making this approach work here.

Although its failure is interesting, the inability to use the "transposition approach" to solve the KFE equation is not *per se* a dramatic problem. Since spectral differentiation matrices don't have to be large and are very easy to assemble, the performance loss is minimal. As it will be useful later on, we present below how, setting aside the problem of the Dirac mass at \underline{a} , a fully spectral approach to the KFE equation would look like.

4.3.2. Pure spectral approach

Even though it only vaguely works in very specific cases, we now present the "fully spectral" approach to the Kolmogorov Forward Equation, as it is very straightforward, quite instructive and will be useful later on when discussing hybrid approaches.

Recall that we are trying to discretize (15) and that we are not allowed to "cheat" by simply transposing A^k , but have to assemble the operator directly. Asset nodes can remain the same (and hence D remains valid), and we denote the collocated stationary density $\mathbf{g} = [g_1 | g_2] = \{g_{1,1}, \dots, g_{1,N}, g_{2,1}, \dots, g_{2,N}\}$. At income j and asset node n , the KFE will write:

$$0 = -D_n(\mathbf{s}_j \circ \mathbf{g}_j) - \lambda_j g_{j,n} + \lambda_{-j} g_{-j,n}, \quad (31)$$

where \circ denotes the Hadamard (element-wise) product. Stacking this system of $2N$ equations into a $2N \times 2N$ matrix, we get:

$$0 = -\mathbf{D}(\mathbf{s} \circ \mathbf{g}) + \mathbf{G}^T \mathbf{g} \quad (32)$$

where we have only used predefined matrices and exploited $G^T \otimes I_2 = (G \otimes I_2)^T = \mathbf{G}^T$ to save us one Kronecker product. Finally, we can get rid of the Hadamard product by writing:

$$0 = -(\mathbf{D}\mathbf{s} + \mathbf{G}^T) \mathbf{g} \quad (33)$$

Assembling a discretized Kolmogorov Forward Equation under the spectral method thus only takes one matrix-vector product, one transposition and a matrix-matrix sum. Writing $\mathbf{C} = -(\mathbf{D}\mathbf{s} + \mathbf{G}^T)$, we now simply have to find the eigenvector \mathbf{g} associated with its eigenvalue 0. (As \mathbf{C} is not the generator of any Markov process, there is no obvious theoretical guarantee that it admits 0 as eigenvalue, but in practice it always does.)

Of course, all of this only makes sense to the extent to which the income-savings process admits a stationary density, and we know it doesn't. In practice, this naïve approach is still vaguely usable when the boundary constraint is "not too binding", in an intentionally loose sense, i.e. when the mass of agents at (z_1, \underline{a}) is "not too high". This typically happens when \underline{a} is close to the natural borrowing constraint $\underline{a}^\natural = -z_1/r$. However, as the value function is highly curved around the borrowing constraint when it is close to the natural constraint, the spectral methods needs comparatively many points to converge in the HJB iteration, which largely nullifies the added value of the approach.

It can, however, still be used. Figure C.3 presents an illustration of this possibility, where we have

kept all parameters of the main two-income example but for the credit constraint, which we have set at $\underline{a} = -\underline{z}/r + 0.05 \approx -2.81$. Moreover, we have used $N = 200$ to account for the higher curvature of the value function. Despite using $I = 500$ points, the finite difference approximation misses the (low but non-zero) mass of agents at the credit constraint; conversely, the spectral solution features some agents near the credit constraint, though these are only agents *close to it* and not *at it* as we are looking at a density. This example shows that the fully spectral approach to the KFE can work quite well in specific contexts, but alas that it cannot work as a general method.⁶

4.3.3. Upwind finite difference / finite volume approach

To deal with the problem of the Dirac mass at the credit constraint, the most obvious solution is to use upwind finite differences. This means interpolating the spectral solution for the savings function to a much finer grid, assembling the upwind difference matrix and then solving for its zero eigenvector. However, to be interpretable as the generator of a finite-state Markov process, the finite difference matrix has to be assembled carefully. In fact, it cannot be done as the upwind discretization of the KF equation; it has to be an upwind discretization of the HJB equation which is then transposed.

Although the end result is the same, we present a slightly different motivation for this discretization, which allows to work directly on the KF equation and will thus be useful later on to discuss hybrid methods: an upwind finite volume scheme. Finite volume methods (FVM) are a class of numerical methods that represent the solution as a mesh of cells (of *finite volume*) and then approximate in- and out- fluxes from these cells via surface integrals. As the flow out of one cell is equal to the flow into its neighbor, FVM form a class of conservative methods, which make them particularly well-suited for convection-diffusion problems. See e.g. [Eymard et al. \(2000\)](#) for an extensive exposition.

Here, we will use upwind finite differences, which approximate in- and out- flows from each cell depending on the direction of flows. The literature on upwind finite differences is also extensive. [Lazarov et al. \(1996\)](#) provide H^1 and L^2 convergence results for cell-centered finite volume methods for a large class of convection-diffusion equations. As the end result is exactly the same, it could be interesting to see (but far beyond the modest scope of this Master Thesis) whether the theory of upwind finite volume methods can offer additional theoretical convergence and stability results for upwind finite differences methods applied to continuous-time heterogeneous-agent models.

[Ahn \(2019\)](#) also offers a brief but useful discussion on upwind finite volume methods specifically applied to computing distributions in heterogeneous-agent models.

To outline the "finite volume view" of the upwind finite differences scheme, let us start with a node of interest (z_1, a_n) . The node is part of a (fine) discretization grid for assets, a_1, \dots, a_I (e.g. with $I = 500$) with constant steps of size Δ_a . Its value $g_{1,n}$ represents the average density over the surrounding cell $[a_n - \frac{\Delta_a}{2}, a_n + \frac{\Delta_a}{2}]$:

$$g_{1,n} = \frac{1}{\Delta_a} \int_{a_n - \frac{\Delta_a}{2}}^{a_n + \frac{\Delta_a}{2}} g_1(a) da \quad (34)$$

In the mesh, the cell surrounding (z_1, a_n) has three adjacent cells: the cell with the same income but fewer assets, (z_1, a_{n-1}) , the cell with the same income but more assets, (z_1, a_{n+1}) , and the cell with the same assets but high income, (z_2, a_n) . These will generate the following in- or outflows:

1. Income-switching: there will be an outflow from the cell towards the corresponding high-income cell, as well as an inflow from that cell into the cell of interest.
2. Savings: as our node of interest has low income and thus negative savings, there will be an inflow

⁶Or, more modestly, that it will take keener minds than me to make it work.

of agents from the cell with more assets as well as an outflow of agents towards the cell with fewer assets.

The question is then how to approximate these various flows, especially as they will all depend on the stationary density (and its cell averages) which we are trying to solve for.

An upwind difference scheme will approximate flows depending on their direction, and take the source as an approximation. In particular, we will consider:

1. Income-switching outflows: outflows towards the "adjacent" high-income cell are simply given by $\lambda_1 g_{1,n}$. As the intensity λ_1 is constant along $[a, \infty]$, this approximation is in fact exact.
2. Income-switching inflows: analogously, inflows from the high-income cell are given by $\lambda_2 g_{2,n}$.
3. Savings outflows: as savings are negative at (z_1, a_n) , we approximate flows from (z_1, a_n) to (z_1, a_{n-1}) using flows leaving (z_1, a_n) , i.e. $s_{1,n} g_{1,n} / \Delta_a$. The Δ_a term represents the distance between the nodes: as agents go from one to the other at speed $s_{1,n}$, a flow of $s_{1,n} / \Delta_a$ will do so per unit of time.
4. Savings inflows: analogously, we approximate flows from (z_1, a_{n+1}) to (z_1, a_n) as flows leaving (z_1, a_{n+1}) , i.e. $s_{1,n+1} g_{1,n+1} / \Delta_a$.

As we are solving for the stationary equilibrium, we want inflows to be equal to outflows, or alternatively net inflows into the cell surrounding (z_1, a_n) to be zero, which will write:

$$s_{1,n+1} g_{1,n+1} / \Delta_a - s_{1,n} g_{1,n} / \Delta_a + \lambda_2 g_{2,n} - \lambda_1 g_{1,n} = 0 \quad (35)$$

A trivial rewriting shows that this is strictly equivalent to an upwind (in this case, forward) finite difference discretisation of the KFE:

$$\frac{s_{1,n+1} g_{1,n+1} - s_{1,n} g_{1,n}}{\Delta_a} + \lambda_2 g_{2,n} - \lambda_1 g_{1,n} = 0 \quad (36)$$

For a cell where savings will be negative, say (z_2, a_n) , the upwind finite volume discretization will symmetrically write:

$$s_{2,n-1} g_{2,n-1} / \Delta_a - s_{2,n} g_{2,n} / \Delta_a + \lambda_1 g_{1,n} - \lambda_2 g_{2,n} = 0 \quad (37)$$

which is in turn strictly equivalent to a backward difference discretization.

Under the general case, using the standard notations $\cdot^+ = \max(\cdot, 0)$ and $\cdot^- = \min(\cdot, 0)$, this writes:

$$c_{j,n-1}^b g_{j,n-1} + (c_{j,n-1}^c - \lambda_{-j}) g_{j,n-1} + c_{j,n+1}^f g_{j,n+1} + \lambda_{-j} g_{-j,n} = 0 \quad (38)$$

$$c_{j,n+1}^b = -\frac{(s_{j,n+1})^+}{\Delta_a} \quad c_{j,n+1}^f = -\frac{(s_{j,n+1})^-}{\Delta_a} \quad (39)$$

$$c_{i,j}^c = \text{sign}\left(\frac{s_{j,n}}{\Delta_a}\right) \quad (40)$$

This is exactly the discretization suggested in Achdou et al. (2021) (e.g. p. 8 of the Numerical Appendix), except we have used c^b, c^c, c^f for the backward, central and forward coefficients, instead of

x, y, z , which could have led to some confusion. We can then collect these terms into a $2I \times 2I$ matrix equation:

$$0 = -\mathbf{D}_F s^+ \mathbf{g} + -\mathbf{D}_B s^- \mathbf{g} + \mathbf{G}^T \mathbf{g} \quad (41)$$

where we have defined:

$$\mathbf{D}_F := I_2 \otimes D_F = \frac{1}{\Delta_a} \begin{bmatrix} -1 & 1 \\ \ddots & \ddots \\ & -1 & 1 \\ & 0 & 0 & -1 & 1 \\ & & & \ddots & \ddots \\ & & & -1 & 1 \\ & & & 0 & 0 \end{bmatrix} \quad (42)$$

$$\mathbf{D}_B := I_2 \otimes D_B = \frac{1}{\Delta_a} \begin{bmatrix} 0 & 0 \\ -1 & 1 \\ \ddots & \ddots \\ & -1 & 1 & -1 & 1 \\ & & & \ddots & \ddots \\ & & & -1 & 1 \\ & & & 0 & 0 \end{bmatrix} \quad (43)$$

Defining $\mathbf{C} := \mathbf{D}_F s^+ - \mathbf{D}_B s^- + \mathbf{G}^T$, we need to solve $\mathbf{C}\mathbf{g} = 0$. This can be done very efficiently by solving for \mathbf{C} 's 0 eigenvalue and its associated eigenvector, and is especially fast because SciPy's `eigs` is tuned to search for (specific) eigenvalues of *sparse* matrices.

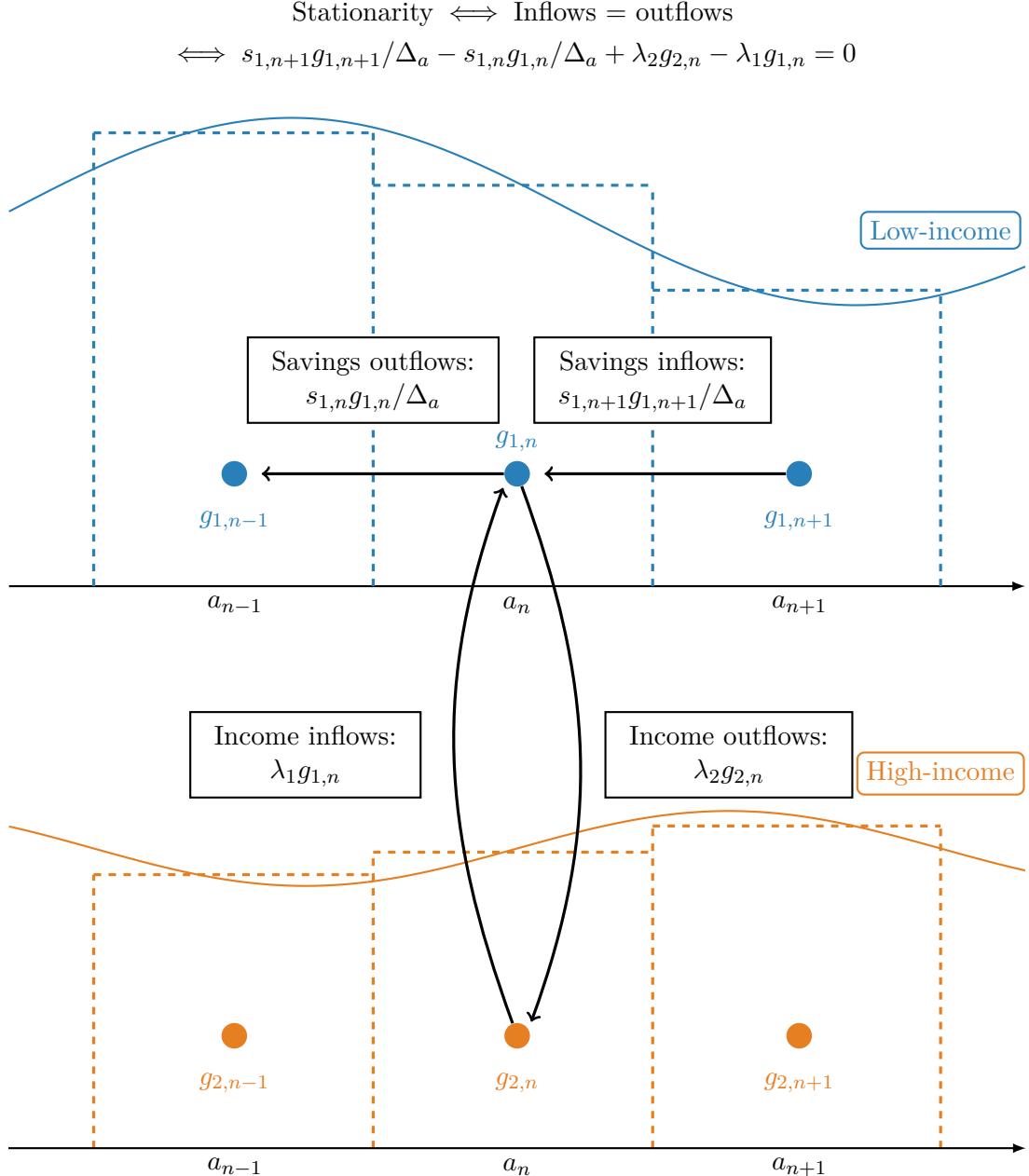
Indeed, it should be noted that as finite difference and finite volume methods offer much lower precision for a given number of nodes, we have to work with a denser mesh and thus larger matrices (in practice, we use the same $I = 500$ from the finite difference comparator). On the other hand, a cursory look at the formulae above shows that C is very sparse, as it only has entries on the diagonals $0, 1, -1, I$ and $-I$. This reflects the local nature of the finite volume / finite difference approximation.

Resorting to finite differences for the KFE thus allows us to exploit the smoothness of the HJB equation to solve it more efficiently using the pseudospectral method, and then to interpolate it to a potentially extremely dense grid to efficiently solve the not-so-smooth KFE. Of course, this means forfeiting much of the elegance of the spectral approach, and it's therefore only natural to look for ways of salvaging at least some of the benefits of the pseudospectral method.

A simple and natural first avenue is to simply apply the finite difference / finite volume approach to the spectral grid, the only complication is that the spectral grid is not uniform. Given the usually far lower number of nodes in a satisfactory spectral solution to the HJB equation, we call this approach the *coarse* finite difference approach. In a sense, the strengths and weaknesses of the spectral grid play against each other here: the far lower number of nodes will make the solution less precise, but the placement of nodes, which is denser at the edges and thus also at the high-density, high-curvature points of the stationary distribution, could mean that it makes up for it.

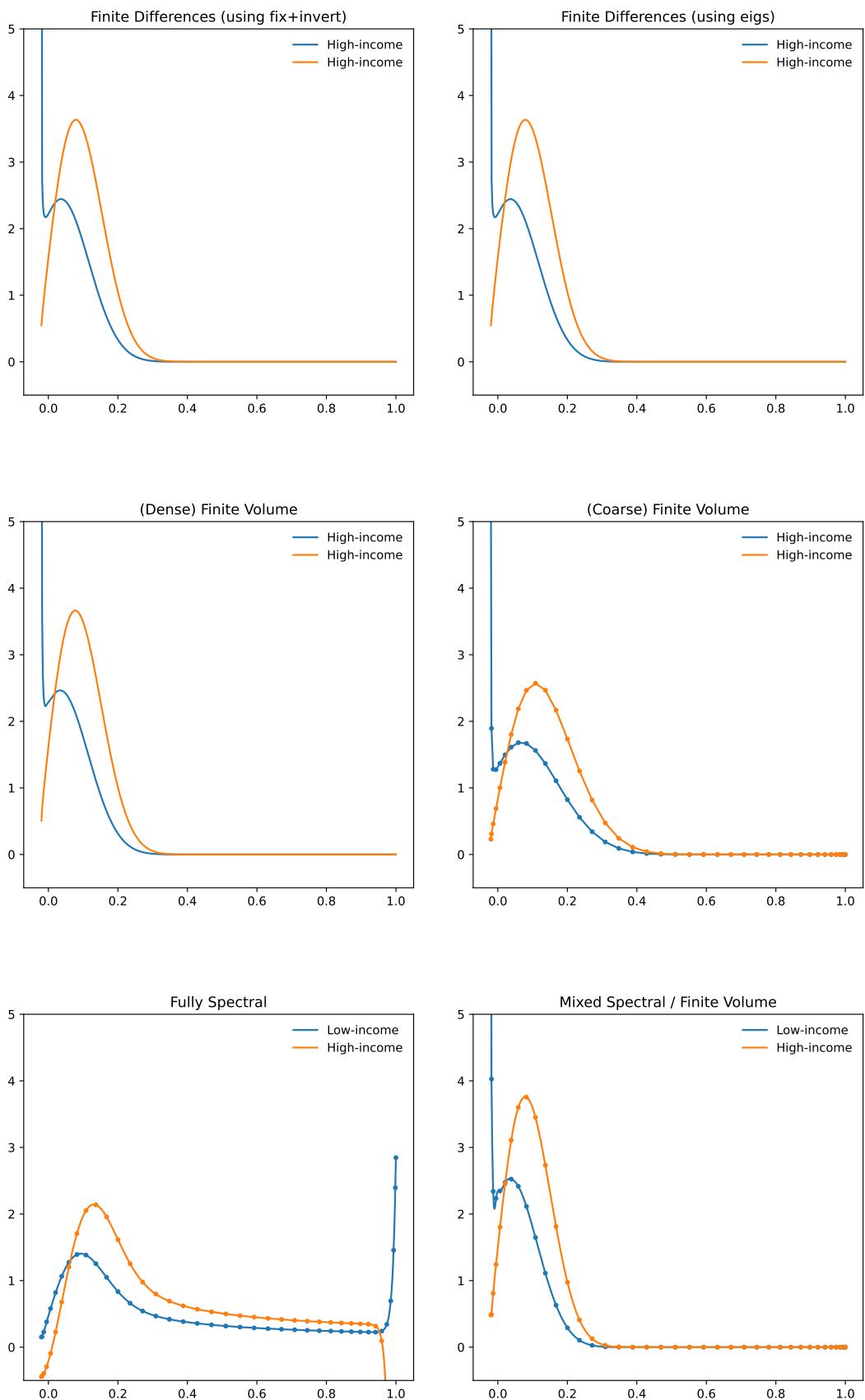
One first has to adapt the upwind finite differences / finite volume discretization to account for the non-uniform grid. This is fairly straightforward to do, especially when thinking in terms of the finite volume rationalization: inflows to and outflows from "the left", i.e. between a_{n-1} and a_n , travel

Figure 7: Illustration of finite volume Approach to Kolmogorov-Forward Equation



Note: This figure illustrates the upwind finite volume approach to the Kolmogorov Forward Equation. The upper part represents low-income agents and the lower part high-income agents, while the axis shows (increasing) assets. The blue line shows the (true) stationary density of low-income agents, while the blue nodes represent discretization points. Nodes are separated by a distance Δ_a , assumed constant for expositional simplicity. Their masses $g_{1,n-1}, g_{1,n}, g_{1,n+1}$ are represented by the dashed blue rectangles, which approximate the density. Analogous objects in orange have the same interpretation, but for high-income agents. Arrows represent an "upwind" discretization of inflows into and outflows from the node of interest $(1, n)$. Stationary implies that outflows equal inflows. The end result is strictly equivalent to an upwind finite differences discretization of the Kolmogorov Forward Equation.

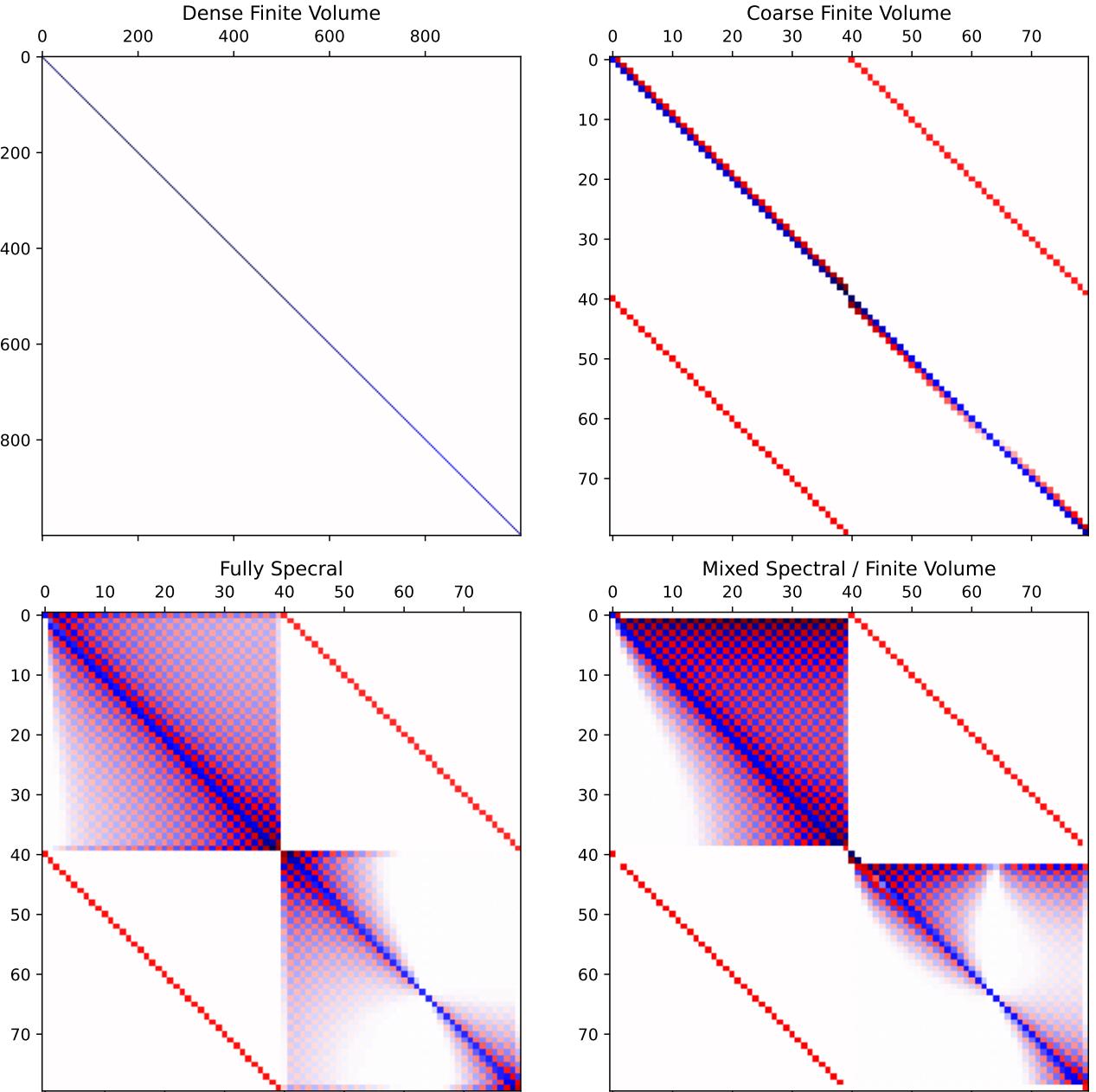
Figure 8: Stationary Distribution using Different Numeric Methods



over a distance $\Delta_a^b = a_n - a_{n-1}$; inflows to and outflows from "the right", i.e. between a_n and a_{n+1} , travel over a distance $\Delta_a^f = a_{n+1} - a_n$. The Δ_a terms thus need to be replaced appropriately in (35), (37) and (38). The assembly into $2N \times 2N$ matrices is also slightly more involved than in (41)

In the end, results show that this approach is *pretty bad, but not too bad*. The distribution shown in Figure 8 is clearly a rough approximation of the better-working methods, but still a reasonable approximation. In the end, this approach can be useful as a rough sanity check on other methods, or when one already operates on a needlessly dense spectral grid.

Figure 9: Kolmogorov Forward Equation - Matrix \mathbf{C} in Different Approaches



4.3.4. Mixed spectral / finite volume approach

A far more powerful approach to retain some of the benefits of the spectral method is to *combine it* with upwind finite differences. The finite volume rationalization of finite differences is especially useful here. In short, the idea is to consider the stationary distribution as a Dirac part at \underline{a} and a continuous part over $[\underline{a}, \infty[$. This naturally leads us to consider a numerical solution as a finite volume part surrounding \underline{a} and a spectral part over all other nodes. We will start by detailing the "cut" spectral component and then explain how we add an upwind finite volume discretization for only two boundary nodes.

The simplest way to make this approach work is to keep the spectral grid from the HJB equation, $a_1 = \underline{a}, \dots, a_N = \bar{a}$. We will consider the finite volume over $[\underline{a}, a_2]$ and the pseudospectral solution over $[a_2, a_N]$. Note that, strictly speaking, Chebyshev nodes do not include the end points, though they are usually included for numerical convenience (e.g., they are useful here): removing one of the endpoints thus doesn't alter the Runge-phenomenon-minimizing properties of the nodes.

However, we need to compute a new $N - 1 \times N - 1$ differentiation matrix for the nodes a_2, \dots, a_N . This is most easily done by simply going back to the derivative of the barycentric Lagrange formula (see [Berrut and Trefethen \(2004\)](#)), which DmSuite's **poldif** implements. We denote this new matrix D^{cut} . Similarly, we take over all notation from the spectral part and label with superscript *cut* their values over a_2, \dots, a_N .

Analogous to the fully spectral approach in (33), the $2(N - 1) \times 2(N - 1)$ matrix equation describing the KFE over a_2, \dots, a_N writes:

$$0 = -\left(\mathbf{D}^{\text{cut}} \mathbf{s}^{\text{cut}} + \mathbf{G}^{T^{\text{cut}}}\right) \mathbf{g}^{\text{cut}} \quad (44)$$

where only $\mathbf{D}^{\text{cut}} = I_2 \otimes D^{\text{cut}}$ is truly new. The spectral part is thus extremely straightforward.

The "finite volume part" over $[a_1, a_2]$ needs to be treated with more caution and discussed at greater length. In the end, it boils down to three equations, which also follow an "upwind" pattern. In the following equations, we will use $\Delta_a = a_2 - a_1$, i.e. the distance between the first and the second spectral node; note that the spacing between any other pair of nodes will be different.

Balancing inflows and outflows from the cell around (z_1, a_1) will write:

$$s_{1,2}g_{1,2}/\Delta_a + \lambda_2 g_{2,1} - \lambda_1 g_{1,1} = 0 \quad (45)$$

Note that, from the credit constraint and Proposition 1 we have $s_{1,1}$, so the $-s_{1,1}g_{1,1}/\Delta_a$ disappears and we have no mass "escaping" the grid. Inflows and outflows from the cell around (z_2, a_1) will write:

$$\lambda_1 g_{1,1} - \lambda_2 g_{2,1} - s_{1,2}g_{1,2}/\Delta_a = 0 \quad (46)$$

Here, there can be no savings inflows, as savings are positive and we are at $a_1 = \underline{a}$.

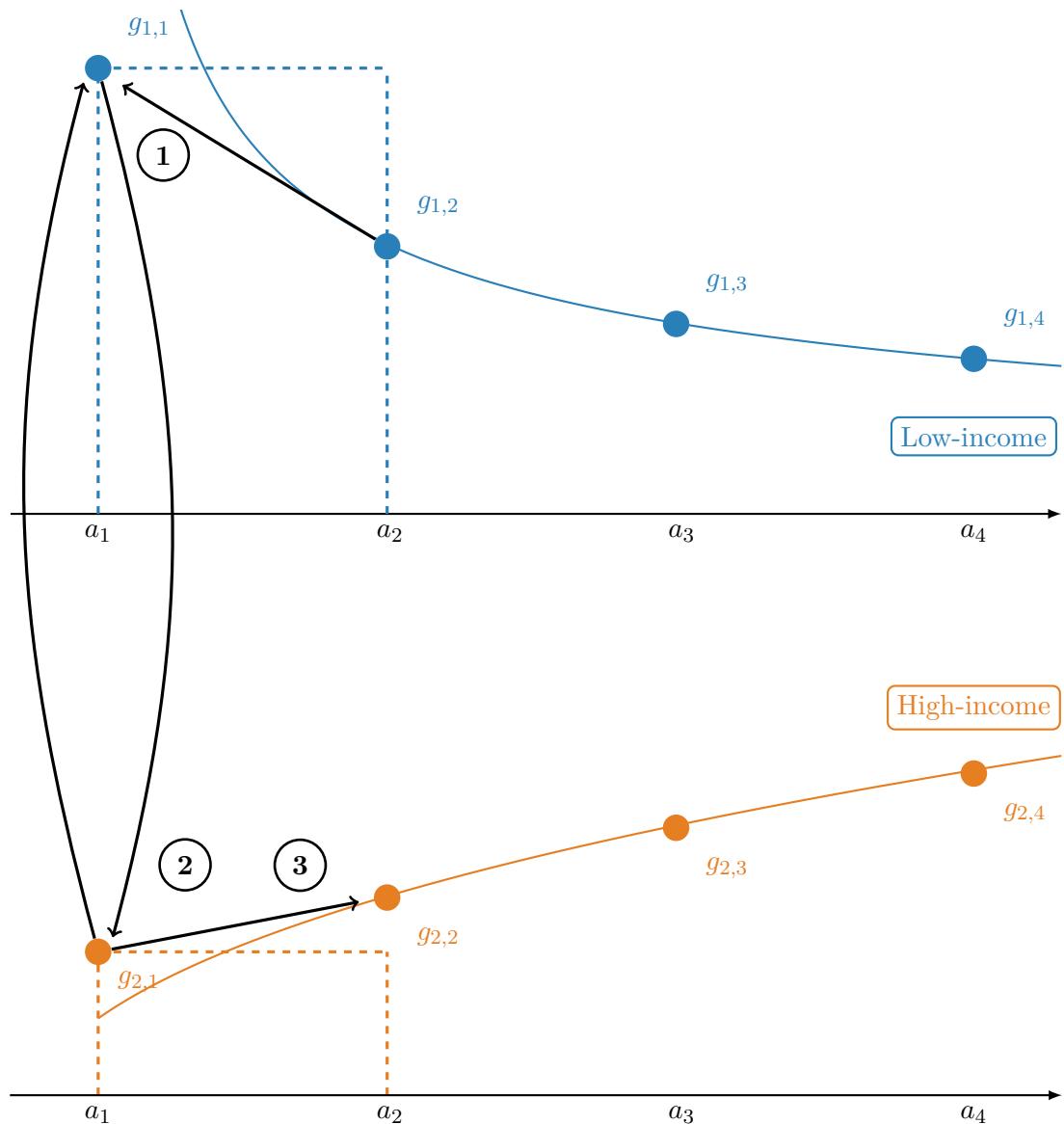
For low-income agents, the spectral part and the finite volume part are "connected" because inflows into the cell around (z_1, \underline{a}) are approximated using its density at its left boundary, $g_{2,1}$: this works because we are in a "backward" section of the state space (i.e. where savings are negative). To "connect" the spectral part and the finite volume part for high-income agents, we impose that savings outflows from (z_2, \underline{a}) should equal savings inflows into (z_2, a_2) , i.e.

$$s_{2,1}g_{2,1} - s_{2,2}g_{2,2} = 0 \quad (47)$$

Equations (45), (46) and (47) respectively correspond to the bullets (1), (2) and (3) in Figure 10.

This hybrid approach, though undeniably haphazard, leverages the power of each method over its optimal domain: it exploits the smoothness of the pseudospectral method over $[a_2, a_N]$, where the

Figure 10: Illustration of Mixed Spectral / Finite Volume Approach to Kolmogorov-Forward Equation



stationary distribution admits a density, and that of the finite volume approach over $[a_1, a_2]$, where the Dirac mass means that an in- and outflow-based discretization will be most useful.

Additionally, implementing this on the existing spectral grid implies that, for a given number of nodes, the finite volume part will be more narrowly focussed on the singularity than with a full finite difference discretization. Indeed, because we are using Chebyshev nodes, the second spectral node a_1 will be much closer to \underline{a} than the second uniformly spaced finite difference node a_2^{FD} . More precisely, normalizing $[\underline{a}, \bar{a}]$ to $[-1, 1]$, the first Chebyshev node will be placed at $a_2 = \cos(\pi - \frac{\pi}{2N}) \sim -1 + \frac{\pi^2}{8n^2}$, which decreases much faster with N than $a_2^{FD} = -1 + \frac{2}{N}$.

Results from the hybrid method are shown on the bottom right panel of Figure 8, and they reveal quite well how powerful this approach can be: over the density part, despite having very few nodes, the interpolated density tracks that of the finite difference approach very precisely; over the Dirac component, the approximation also works well despite having so few points because the width of the first cell is very small.

Speed tests in the benchmark from Section 8 further show that this approach requires few nodes to converge, so that for a given runtime it clearly dominates the other methods. Moreover, this hybrid method is quite general in the sense that it could be adapted to other models with Dirac masses at certain points, e.g. because of kinked transaction costs. Sadly however, this method is too complex to be made to work in models where the income process itself requires a complex discretization (i.e. any process more complex than a discrete-space Markov process), which is regrettable as these smooth, multidimensional models are exactly the contexts in which the pseudospectral approach to the HJB equation shines.

To illustrate both the power of the spectral method in this case and how one can nonetheless efficiently solve for the stationary distribution, we now turn to a model where income is continuous and follows an Ornstein-Uhlenbeck diffusion process.

5. Spectral method for a Huggett model with diffusive income

To showcase the power of the spectral approach for smooth problems, we now turn to a heterogeneous-agent model where, instead of following a two-state continuous-time Markov process, income follows a diffusion process.

5.1. A simple heterogeneous-agent model with Ornstein-Uhlenbeck income

More precisely, we assume income follows a Ornstein-Uhlenbeck process with drift θ around the mean \check{z} and variance σ^2 :

$$dz_t = \theta(\check{z} - z_t) dt + \sigma dB_t \quad (48)$$

where B is a standard Brownian motion. To make it representable on a computer, we additionally have to assume that the process is reflected at the boundaries $\underline{z} < \bar{z}$.

This application closely follows the extension to diffusive income in the Numerical Appendix to Achdou et al. (2021), so we only briefly outlining the PDEs it generates to on its numerical solution. The model yields the following set of Hamilton-Jacobi-Bellman and Kolmogorov Forward equations:

$$\rho v(a, z) = \max_c u(c) + (z + ra - c)\partial_a v(a, z) + \mu(z)\partial_z v(a, z) + \frac{\sigma^2}{2}\partial_{zz} v(a, z) \quad (49)$$

$$0 = -\partial_a[s(a, z)g(a, z)] - \partial_z[\mu(z)g(a, z)] + \frac{\sigma^2}{2}\partial_{zz}[g(a, z)] \quad (50)$$

where $\mu(z) = \theta(z - z)$. As in the discrete-income model, borrowing constraint $\underline{a} < a$ is equivalent to a boundary condition on the value function v :

$$\partial_a v(\underline{a}, z) \geq u'(z + ra), \quad \forall z \in [\underline{z}, \bar{z}] \quad (51)$$

However, the reflecting boundaries at \underline{z} and \bar{z} now add a second boundary constraint in the z dimension:

$$0 = \partial_z v(a, \underline{z}) = \partial_z v(a, \bar{z}), \quad \forall a \in [\underline{a}, +\infty[\quad (52)$$

As before, we choose to focus on a Hugget model where the interest rate r is fixed and exogeneous. More complicated model, e.g. with a production-side or studying the transition from one rate to another, will simply nest this simpler model within common iteration or time-marching routines.

5.2. Solving the diffusive Hamilton-Jacobi-Bellman equation

Solving the Hamilton-Jacobi-Bellman equation using a pseudospectral method is only marginally more complex than in the two-income case, and works just as well. This example shows the power of a polynomial approach for smooth problems over a multidimensional space.

5.2.1. Numerical procedure

As before, to collocate assets, we choose N Chebyshev nodes over $[\underline{a}, \bar{a}]$, which we denote a_1, \dots, a_N . Similarly, we now also have to choose M Chebyshev nodes over $[\underline{z}, \bar{z}]$, which we will write z_1, \dots, z_M . Our collocation grid is thus the tensor product of $\{a_1, \dots, a_N\}$ and $\{z_1, \dots, z_M\}$, with size $N \times M$. We thus approximate the value function v by its values $v_{n,m} = v(a_n, z_m)$. We write $\mathbf{v} = [v_{1,1}, \dots, v_{1,M}, \dots, v_{N,1}, \dots, v_{N,M}]$ the concatenated $N \times M$ vector. In practice, we choose $N = 30$ and $M = 20$.

As before, we use an implicit scheme to solve the nonlinear HJB equation. We start from an initial guess \mathbf{v}^0 given by the standstill value:

$$v_{n,m}^0 = \frac{u(z_m + ra_n)}{\rho} \quad (53)$$

At each step, we update \mathbf{v}^k via:

$$\frac{v_{n,m}^{k+1} - v_{n,m}^k}{\Delta} + \rho v_{n,m}^{k+1} = u(c_{n,m}^k) + (z_m + ra_n - c_{n,m}^k) \partial_a v_{n,m}^{k+1} + \mu(z) \partial_z v_{n,m}^{k+1} + \frac{\sigma^2}{2} \partial_{zz} v_{n,m}^{k+1} \quad \forall n, m \quad (54)$$

where $\partial_a v_{n,m}^{k+1}$, $\partial_z v_{n,m}^{k+1}$ and $\partial_{zz} v_{n,m}^{k+1}$ are intentionally loose notations for the derivative of the polynomial at each node. As before, these nodal derivatives can be expressed as linear operations on the value of each node, which can in turn be stacked into a differentiation matrix. We thus have to compute three differentiation matrices, which we will denote D_a (of size $N \times N$), D_z and D_{zz} (both of size $M \times M$). This is done in milliseconds by the `Dmsuite` package.

To give an example, the first z -derivative at the node (a_n, z_m) will be given by:

$$\partial_z v_{n,m}^{k+1} = D_{z,m} (v_{n,1}^{k+1}, \dots, v_{n,M}^{k+1})' \quad (55)$$

where $D_{z,m}$ is (very poor notation for) D_z 's m -th row. To operate correctly on the $N \times M$ -size vector \mathbf{v}^{k+1} , we have to assemble these into the following $NM \times NM$ matrices:

$$\mathbf{D}_a = D_a \otimes I_M \quad \mathbf{D}_z = I_N \otimes D_z \quad \mathbf{D}_{zz} = I_N \otimes D_{zz} \quad (56)$$

Exactly like in the discrete-income model, we additionally need to compute the consumption from the first order condition $c_{n,m}^k = u'^{-1}(\partial_a v_{n,m}^k)$. Using consumptions derived from the current value

function means we are, again, using a semi-implicit scheme. We then also need to compute savings $s_{n,m}^k = z_m + ra_n - c_{n,m}^k$, utilities $u_{n,m}^k = u(c_{n,m}^k)$ and stack both into the $N \times M$ vectors \mathbf{s}^k and \mathbf{u}^k . Finally, we have to assemble the vector $N \times M$ vector of drifts $\boldsymbol{\mu} = I_N \otimes (\mu(z_1), \dots, \mu(z_M))$. In matrix notation, the implicit update in the diffusive case then writes:

$$\frac{1}{\Delta}(\mathbf{v}^{k+1} - \mathbf{v}^k) + \rho \mathbf{v}^{k+1} = \mathbf{u}^k + \mathbf{s}^k \mathbf{D}_a \mathbf{v}^{k+1} + \boldsymbol{\mu} \mathbf{D}_z \mathbf{v}^{k+1} + \frac{\sigma^2}{2} \mathbf{D}_{zz} \mathbf{v}^{k+1} \quad (57)$$

We can again assemble these into $\mathbf{A}^k := \mathbf{s}^k \mathbf{D}_a + \boldsymbol{\mu} \mathbf{D}_z + \frac{\sigma^2}{2} \mathbf{D}_{zz}$. Finally, (21) can be rearranged into:

$$\mathbf{B}^k := \left(\frac{1}{\Delta} + \rho \right) \mathbf{I} - \mathbf{A}^k, \quad \mathbf{b}^k := \mathbf{u}^k + \frac{1}{\Delta} \mathbf{v}^k \quad (58)$$

$$\mathbf{B} \mathbf{v}^{k+1} = \mathbf{b}^k \quad (59)$$

where $\mathbf{I} = I_N \otimes I_M = I_{NM}$. We are thus back to a simple matrix inversion problem.

The similarities and differences between the finite differences and the spectral approach in the diffusive income case closely follow those of the discrete-income case. However, two points deserve some closer discussion. One of the main drawbacks of the spectral approach is that, as it relies on global instead of local approximations, its differentiation matrices are full, i.e. all coefficients in D_a , D_z and D_{zz} will be non-zero. However, the broadcasted versions \mathbf{D}_a , \mathbf{D}_z and \mathbf{D}_{zz} of size $NM \times NM$ are somewhat sparse: they will respectively have densities of $1/M$, $1/N$ and $1/N$. However, as can very well be seen in Figure 11, this is sparsity across different dimensions. When they are combined into \mathbf{A} , this matrix only has a density approximately equal to $1/N + 1/M - 1/NM$.⁷ For most practical values of N and M , this will be too high to obtain any performance gain from sparse matrix routines.

On the other hand, by their nature, diffusions induce very smooth functions, which applies here to the z -dimension of the value function. That's why a low number of nodes can be enough to very accurately approximate the value function over the whole space. While in a discrete-income model the spectral approach does not offer any additional gain, since every income state has to be described by a separate ODE, in the continuous-income case we can leverage its power to obtain a low-resolution but high-accuracy collocation of the income dimension.

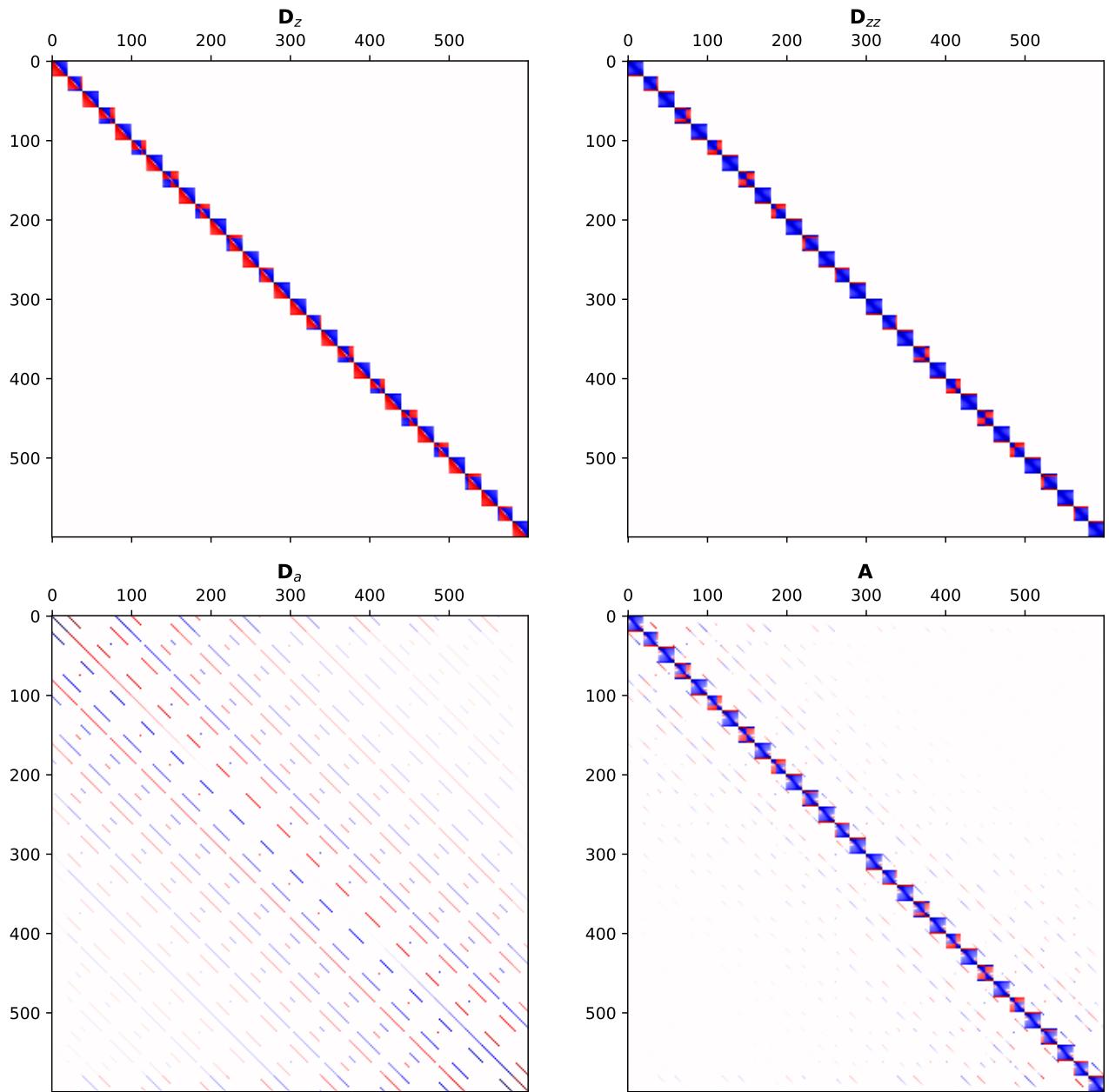
To incorporate the borrowing constraint and the reflecting boundary, which both translate into boundary conditions for the PDE (49), we again have to boundary-border the matrix \mathbf{A} . Adding the reflecting boundary (52) is very straightforward: at all income boundary nodes of the form (a_n, z_1) or (a_n, z_M) for all n , we have to set the first z -derivative of v equal to 0. We thus simply have to replace all corresponding rows in \mathbf{B} by \mathbf{D}_z , and all corresponding coefficients in \mathbf{b} by 0.

Incorporating the credit constraint is more involved, because it is an inequality constraint. Unlike in the two-income case, where theoretical results indicated where it binds and doesn't bind, with continuous incomes we don't have this luxury. A possible approach would thus be to solve (59) with a non-linear constraint, which is feasible but (i) nullifies the added value of the implicit scheme, whose whole point was to transform a non-linear problem into a linear one, (ii) is much slower given the size of the problem at hand.

A far more efficient approach, which yields equivalent results in practice, is to circumvent this problem using a semi-implicit approach. Instead, we (i) compute an unconstrained update $\tilde{\mathbf{v}}^{k+1}$ via (27) without imposing the borrowing constraint (51) (but keeping (52)); (ii) check whether $\tilde{s}_{1,m}^k = z_m + ra - u'^{-1}(\partial_a \tilde{v}_{1,m}^k) < 0$, i.e. where the constraint is violated; (iii) compute a constrained update \mathbf{v}^{k+1} via (59) while imposing (51) with equality at the violating nodes. Imposing the credit constraint

⁷In our numerical application, we use $N = 30$, $M = 20$ and \mathbf{A} has a density of $0.0817 \approx 1/30 + 1/20 + 1/30 \times 20 = 0.0816\bar{6}$

Figure 11: HJB under Diffusive Income - Assembly of \mathbf{A}



with equality is done via boundary-bordering exactly like in the two-income case, by setting \mathbf{B} equal to \mathbf{D}_a at the relevant rows and \mathbf{b} to $u'(z_m + r\underline{a})$ at the relevant coefficients.

As we now have to compute a constrained and an unconstrained update, this procedure doubles (!) the execution time of a spectral solution. However, as the spectral method is already remarkably fast, given its ability to achieve high precision with few nodes, this cost is not prohibitive. especially compared to the non-linearly constrained alternative. Moreover, as the computation of the polynomial derivative using D_a acts globally on all values of \mathbf{v} and not locally, the spectral approach allows for a fast convergence of the implicit scheme despite this non-linear boundary condition. The various tests on the spectral HJB solution, presented in Subsubsection 5.2.3, show that it indeed respects the boundary conditions while being optimal in all other testable respects.

Moreover, as we impose boundary conditions using global derivatives instead of local ones, it means that the solution respects the boundary condition "globally", in the sense that a naïve derivative of the approximate spectral solution will respect it. On the other hand, in the upwind finite differences case, imposing the boundary condition relies on meticulous derivation in the right direction and on using the right ghost nodes, which means that a naïve derivative of the solution will not seem to respect the boundary condition. This is especially important when computing savings from the finite difference solution, which has to be done very carefully or could otherwise produce boundary-violating results. In that respect, the spectral solution is thus much closer to the true solution of the HJB equation.

As in the two-income case, the interaction of local boundary conditions that the initial guess blithely violates and global updating can produce sharp corrections at the edges in the first few iterations. On the other hand, the later iterations are sufficiently precise to allow for extremely large step sizes. We thus, again, use increasing time steps of $\Delta_k = \times e^k$, which yield fast and robust convergences.

To synthesize, our algorithm to find a spectral solution to the HJB equation in the diffusive case takes the form:

1. Start with an initial guess \mathbf{v}^0 given by (29).
2. Compute \mathbf{c}^k using $c_{j,n}^k = u'^{-1}(\partial_a v_{n,m}^k)$.
3. Solve for $\tilde{\mathbf{v}}^{k+1}$ using (27), while imposing only the reflecting boundary constraint (52).
4. Solve for \mathbf{v}^{k+1} using (27), while imposing the reflecting boundary constraint (52) and the credit constraint (51) at $\tilde{\mathbf{v}}^{k+1}$'s violating nodes.
5. If $\|\mathbf{v}^{k+1} - \mathbf{v}^k\|_\infty < \varepsilon$, exit the iteration; otherwise, return to step 2.

5.2.2. Numerical results

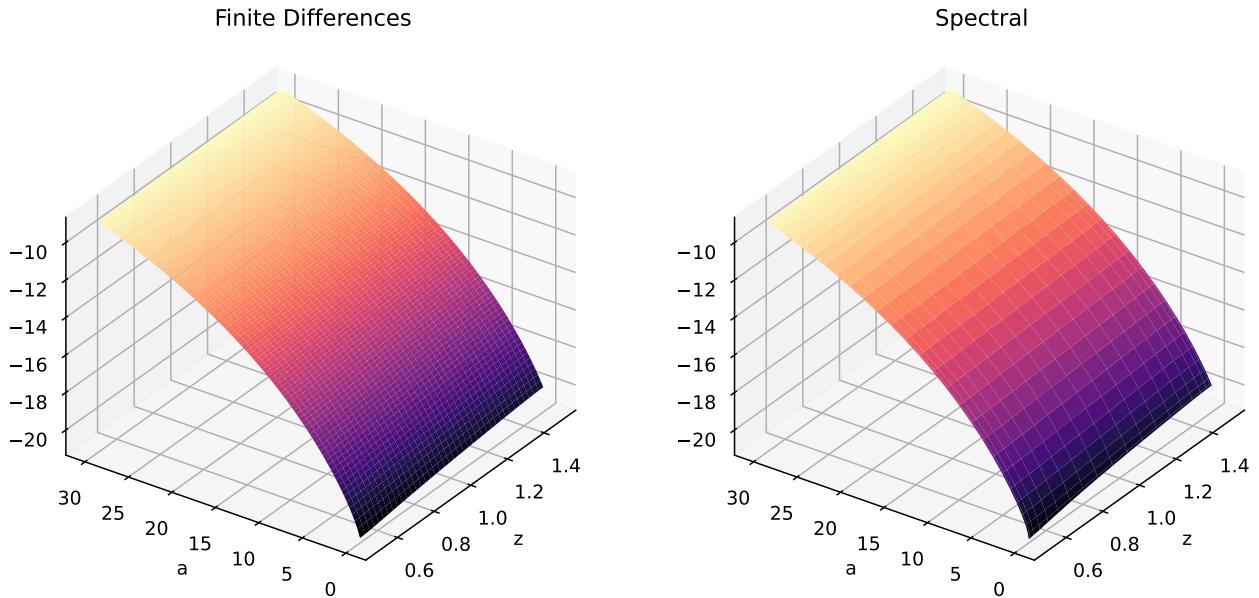
In our numerical application, we use an intentionally low number of nodes in each dimension to showcase the power of the spectral approach for smooth problems. We use $N = 30$ collocation nodes for the asset dimension and $M = 20$ nodes for the income dimension. We compare these with the results presented by Achdou et al., who use $I = 100$ and $J = 40$ nodes respectively. Our spectral application thus uses $30 \times 20 / 100 \times 40 = 15\%$ of the nodes of the finite difference approach. All other parameters follow those used by Achdou et al. for the sake of comparability, and are shown in Table 1. The finite difference and spectral approach take respectively 99.5 ms vs 90.7 ms⁸.

Figure 12 plots the value function obtained using both approaches, which look remarkably similar. Though it appears to be coarser given its larger nodes, this is only because the plotting function interpolates linearly, while the polynomial interpolant is actually remarkably smooth. Figure D.1 shows

⁸Computed as an average of 70 runs each.

the polynomial interpolated to the denser finite difference mesh: it is indistinguishable from the finite difference version, even though it uses far fewer collocation nodes.

Figure 12: Value Functions



Turning to savings functions, differences between both approaches start to show. At first sight, from Figure 13, both appear remarkably similar, and the interpolated version shown in Figure D.2 is again extremely close. However, the spectral savings functions features small oscillations at the lower and upper income limits. This is very likely because we impose the boundary condition on these nodes instead of the interior PDE. Slightly increasing the number of nodes, e.g. setting N to 50, remains remarkably fast and makes these imperceptible.

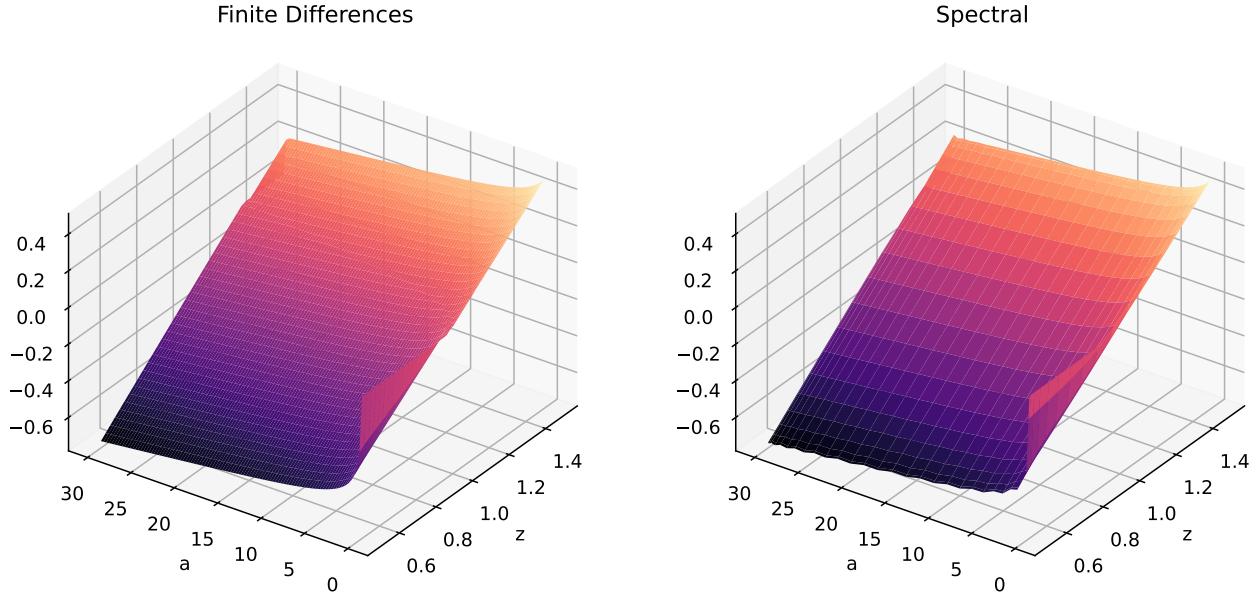
Although it does not have oscillations at the income boundaries, the finite difference savings function features a singularity roughly in the middle: when savings go from negative to positive (roughly around \check{z}), the upwind differentiation direction changes. This produces a discontinuity that can be seen in the middle of the savings function in Figure 13. Figure D.3 shows a zoomed-in version with cell edges that make this discontinuity starkly visible.

Again, it should be stressed that increasing the number of grid points for each method only modestly increases execution times while making all of these errors immaterially small, so that the point of this analysis is principally to understand and compare their error patterns.

Computing the difference between the interpolated versions obtained via both approaches, as is done in Figure 14, gives us an idea of the relative importance of these errors. The left panel shows errors in the value functions, which are quite small given it ranges roughly from -10 to -20 while errors are of the order ± 0.1 . Moreover, differences only appear at the edges, and are starker where the credit constraint binds (i.e. when both a and z are low, on the South-West of the graph): they reflect the previously-discussed phenomenon whereby the finite difference solution only makes sense when differentiated correctly, which does not apply when we (naively) look at differences in undifferentiated, interpolated value functions. The location of these errors thus show that they reflect the fundamentally different treatment of boundary conditions in both approaches.

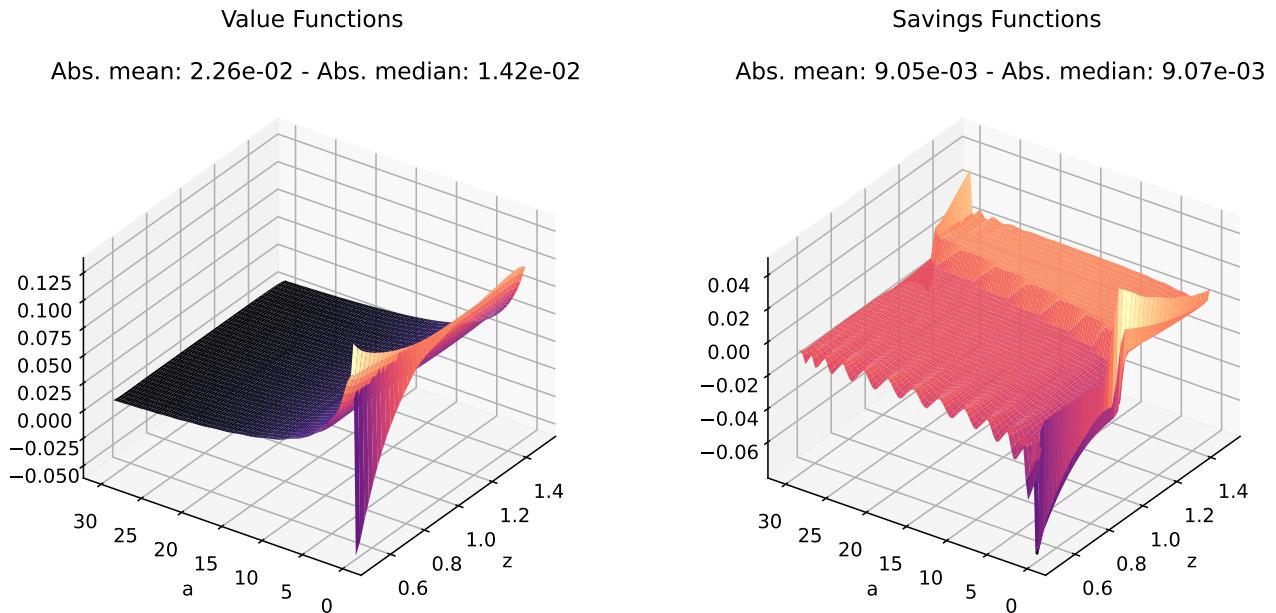
The right panel shows error in the savings functions, which reveal that the discontinuity error due to finite differences (clearly visible as a big step in the middle) is of the same order of magnitude as the

Figure 13: Savings Functions



oscillation errors due to the spectral approach. It could be argued that, in relative terms, errors at the edges are less important than errors in the interior of the domain, as there will be few agents at either extreme of the income distribution but far more around its mean. However, in any concrete implementation, both of these errors can be made arbitrarily small by increasing the side of the meshes.

Figure 14: Difference in (Interpolated) Value and Savings Functions



5.2.3. Boundary condition and consistency checks

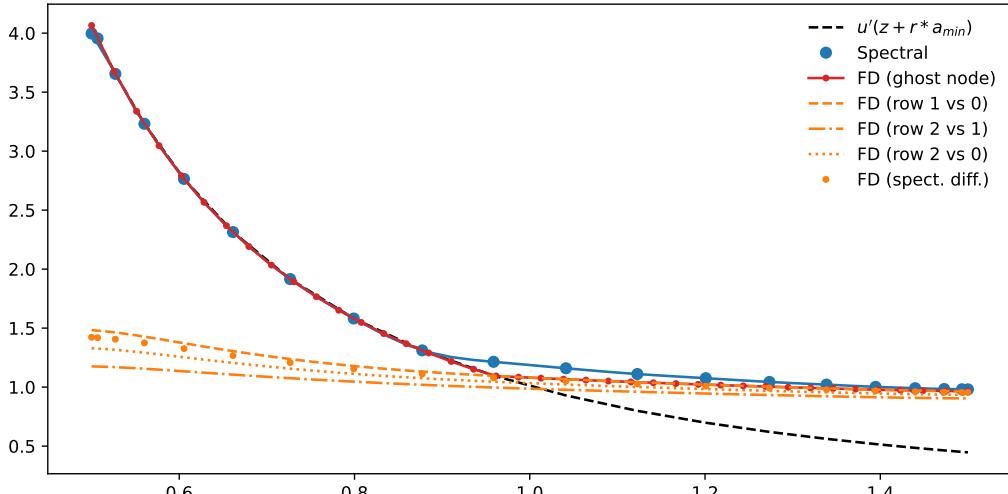
As a general remark, and as we have stressed above, we should underline that, by nature, the finite difference approach produces a piecewise affine approximate solution of the HJB solution that only makes sense if it is treated very carefully, especially when it is differentiated. This plays an important role when checking optimality and boundary conditions. On the other hand, the spectral approach that yields an approximate solution that is a polynomial, but that can then be treated very naively, and notably be differentiated directly using Chebyshev differentiation matrices.

To illustrate this phenomenon, we have (linearly) interpolated the finite differences solution to the spectral nodes, computed its derivatives using the already-assembled differentiation matrices and then calculated savings using the first order condition. Figure D.4 shows the difference between savings computed this way and savings obtained by interpolating the finite difference savings function (shown e.g. in the left panel of Figure 13) to the spectral nodes. The right panel repeats this for the spectral solution, which shows that errors are 0, a tautological result.

To inspect the borrowing constraint, Figure 15 plots the cross-section along z of the a -derivative of the value function at \underline{a} , i.e. $\partial_z v(z, \underline{a})$ as it appears in (51). In the pseudospectral case the derivative is easily computed via polynomial differentiation, and in the finite difference case it has to be carefully computed using ghost nodes. Out of curiosity, and to stress again the different incorporation of boundary conditions in both approaches, we have also computed the derivative of the value function using the difference between the first vs second, third vs second and third vs first rows, as well as via polynomial interpolation and then differentiation: these produce broadly similar estimates of the local derivative, which are all (i) very different from the derivative computed using ghost nodes, (ii) clearly in violation of the boundary condition. Again, this does not mean that either result is wrong, only that one should be careful when comparing their outputs because they do not have the same meaning.

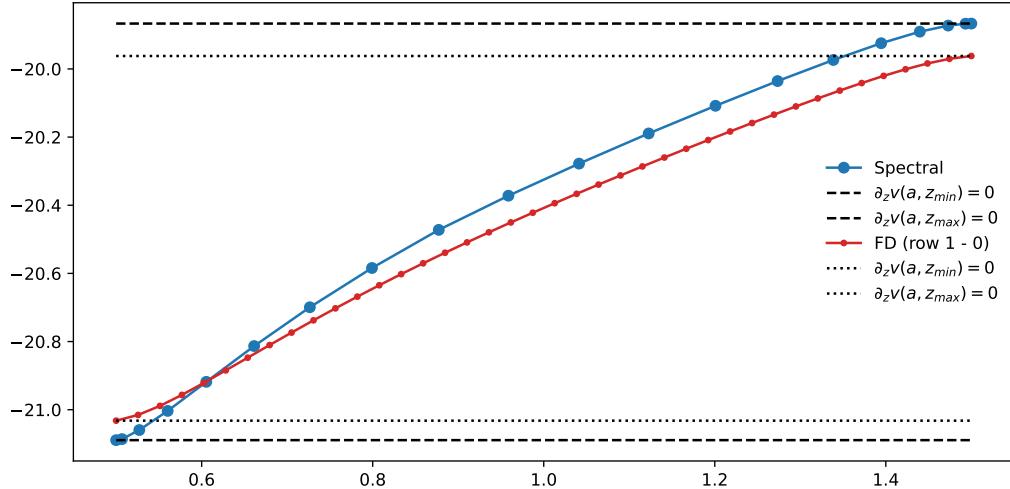
Figure D.5 plots the cross-section along z of the savings function at \underline{a} , which is just a monotonic transform of the previous Figure. The Figures show that both approaches satisfy the boundary condition, but also that the spectral solution is "too smooth": the boundary condition implies a high curvature in the value function at the point where it starts to bind, which the spectral solution does not catch because it has very few collocation nodes. The interesting part is then that it still manages to incorporate the savings function *globally*, i.e. respect it and integrate it into its general shape, which tracks the finite difference solution in lower-curvature areas very well. Again, it should be stressed that all of these differences disappear by increasing the number of collocation nodes, for which execution times remain largely trivial.

Figure 15: Cross-section along z of the a -derivative of the value function at \underline{a}



To inspect the reflecting boundary condition, which says that the value function should be flat around the income edges, we plot the cross-section along z of the value function at (say) \underline{a} in Figure 16. Indeed, we can see that both the finite difference and the spectral densities admit a zero derivative at each edge, i.e. that they respect the boundary condition. Additionally, Figure D.6 plots the z -derivative of the value function at \underline{z} , showing here again that the spectral solution reflects the boundary condition but that naively computed numerical derivatives of the finite differences approach do not.

Figure 16: Cross-section along z of the value function at a_{min}



Before concluding this Subsection, we repeat the consistency checks we had performed in the two-state example. Although simulation of the income process for a short period of time Δ_t is now slightly more involved, the rest of the procedure stays exactly the same. As before, we can also replace the simulation by a direct computation of the expectation; although in the spectral case, this can be done exactly and rapidly by interpolating the surface polynomial (tensor product of polynomials) and integrating it directly using SciPy. Simulated consistency checks are shown in Figure 17 while integrated consistency checks are shown in Figure D.7: errors are very small in both approaches, but slightly smaller in the case of the spectral method.

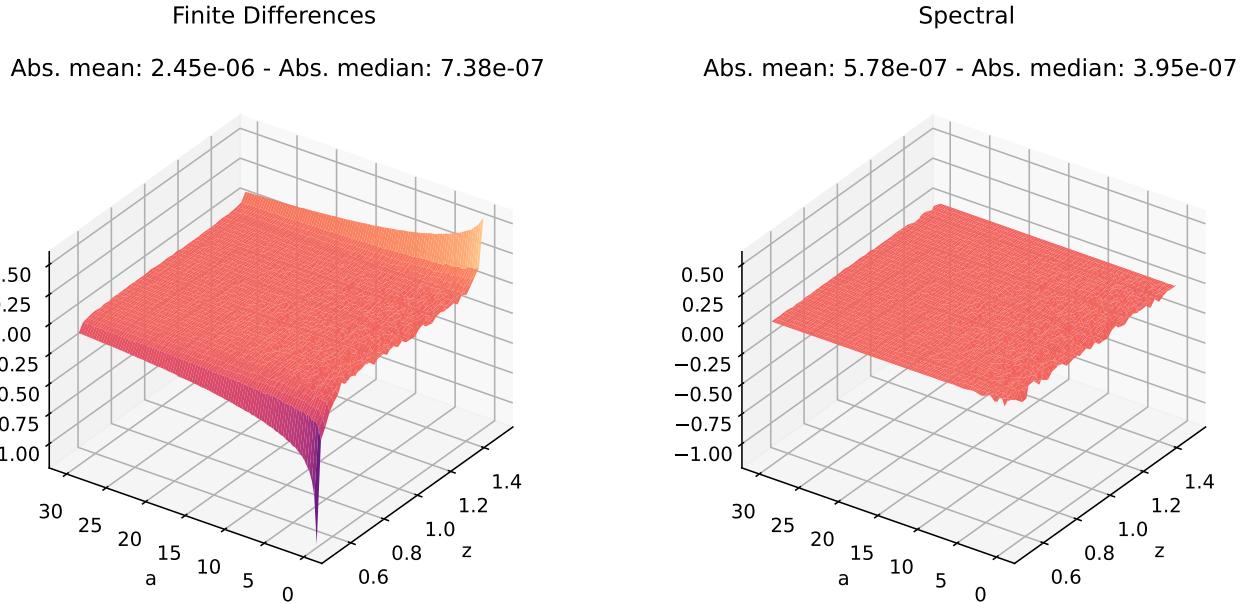
5.3. Solving the diffusive Kolmogorov Forward equation

Although there is no proposition in Achdou et al. (2021) or elsewhere establishing this formally, it seems intuitive, and numerical simulations seem to verify, that there is also a mass of (low-income) agents at the credit constraint in this diffusive-income model. This again means that the stationary distribution does not admit a density, and will be a considerable source of complications.

5.3.1. Spectral approach

A purely spectral approach to the Kolmogorov Forward equation entails assembling a differentiation matrix. Even though income is now much more complex, the same difficulty as in the two-income case now plays out: although the operators appearing in the HJB and KF equations are adjoint (as they are the generator and its adjoint), we can only transpose the discretization of the HJB to obtain a discretization of the KFE in the finite difference approach; in the spectral approach, this doesn't work, principally because spectral differentiation matrixes are not skew-symmetric.

Figure 17: Simulated value function consistency errors



However, as in the two-income case, assembling the spectral differentiation matrix for the KF equation is not a major problem *per se*, as we simply have to write:

$$0 = \mathbf{D}_a (\mathbf{s} \otimes \mathbf{g}) - \mathbf{D}_z (\boldsymbol{\mu} \otimes \mathbf{g}) + \frac{\sigma^2}{2} \mathbf{D}_{zz} \mathbf{g} \quad (60)$$

Defining $\mathbf{C} := \mathbf{D}_a \mathbf{s} - \mathbf{D}_z \boldsymbol{\mu} + \frac{\sigma^2}{2} \mathbf{D}_{zz}$, this simply becomes:

$$0 = \mathbf{C} \mathbf{g} \quad (61)$$

i.e. we are searching for \mathbf{C} 's eigenvector associated with the eigenvector zero. Note that, as we are using the same nodes as before, we can reuse all the differentiation block matrices from the HJB step, and only have to add them together (very slightly) differently. Searching for the zero-eigenvector can also be done very quickly using ScipPy's `eigs`, which searches for the lowest-magnitude eigenvalue and eigenvector. However, as always, \mathbf{C} will not be sparse.⁹

Although in the two-income case, brutishly using the spectral approach to the KFE equation could work when the borrowing constraint was weakly binding and the mass of credit-constrained agents was low, practice shows that it essentially never works under the diffusive income. Indeed, having a two-dimensional income space means that using a comparatively high number of nodes (say $N \sim 200$) becomes essentially impossible. A purely spectral approach is thus confined to the very poor, somewhat comical results shown in Figure 18.¹⁰

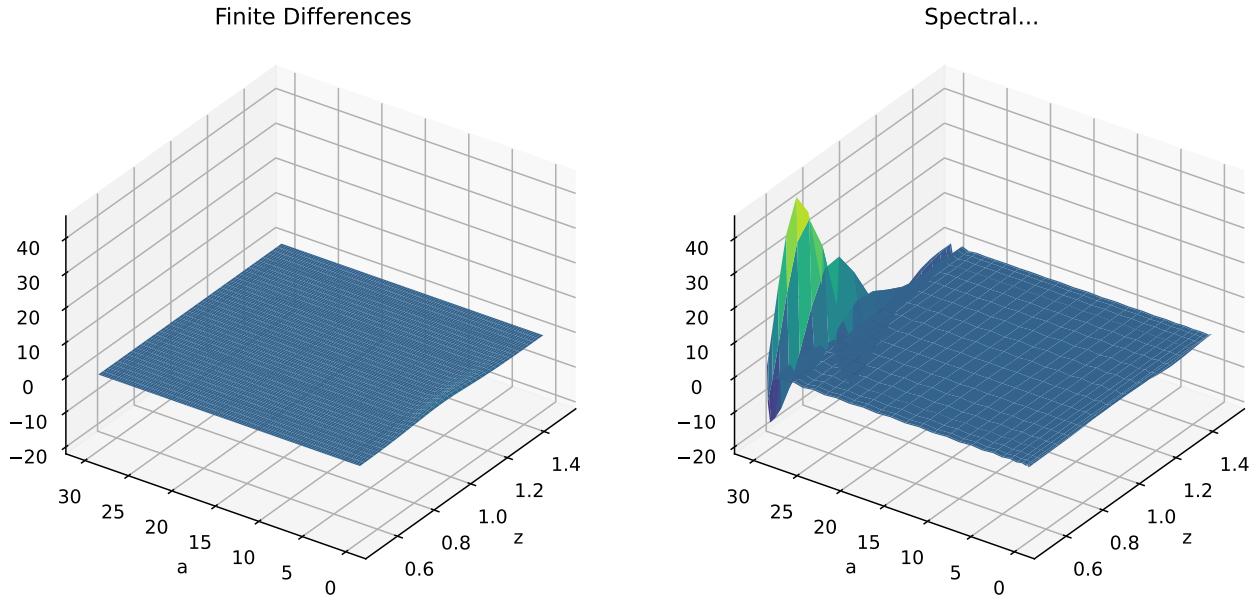
5.3.2. Finite volume approach

The simplest and by far most effective solution to this problem is to sidestep it entirely and solve the KFE via finite differences / finite volumes. As the direction of savings is already known,

⁹Or, more precisely, not sparse *enough* to warrant using sparse matrix routines. It has exactly the same sparsity structure as \mathbf{A} , which is discussed above.

¹⁰The scale of values is somewhat wild because the density is normalized to integrate to 1 (using exact, two-dimensional polynomial integration); as it is negative in some areas, this yields a range that is very different from that of the finite difference approach.

Figure 18: Stationary Distributions



it is remarkably simple to assemble upwind differentiation matrices. Figure 19 shows the stationary distribution obtained by discretizing the KFE on the finite difference mesh, which uses $I = 100$ and $J = 40$ points. Both distributions look near-identical, although the spectral method exhibits a higher peak around the mean income \bar{z} : this reflects (i) the excess smoothness of the spectral savings function around the borrowing constraint, (ii) the discontinuity in the finite difference savings function when savings change sign, because the differentiation direction also changes.

As the KFE is solved in a single step using sparse matrix routines and not via an iterative scheme that involves inverting a matrix at each step, we can interpolate the spectral savings function to a much finer grid than a pure finite differences would allow.¹¹ We thus end up leveraging the power of the smooth method for the smooth problem, the HJB, and of the coarse method for the singular problem, the KFE.

Moreover, as in the two-income case, a carefully-assembled finite volume discretization of the KFE can be interpreted as the (transpose of the) generator of the income-savings Markov process. This ensures that it always admits a zero-eigenvalue, but can be exploited further to study its full spectrum of eigenvalue, which is done more extensively in Section 7.

As a final illustration of the added value of the spectral approach, we can run a simple experiment: try to assess whether it accurately computes aggregate capital. It turns out that it does so rather poorly, but less poorly than finite differences, in a sense which will now make clear. Using the same parameters as described up to now, finite differences evaluate aggregate capital at $k_{FD} = 0.7413$, while the spectral method for the HJB combined with the finite volume approach to the KFE yield $k_{Sp} = 0.643$.

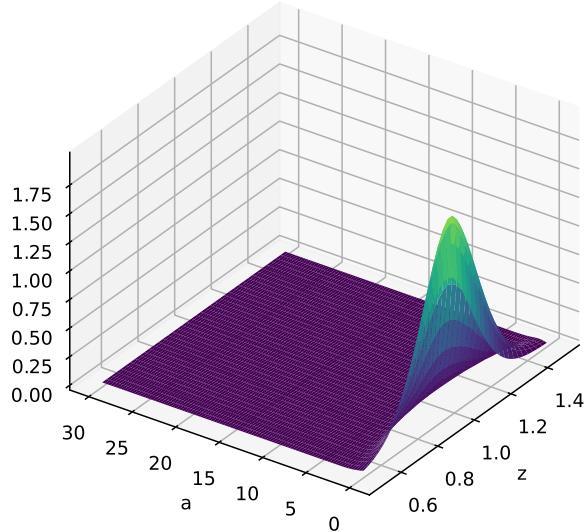
As stressed before, as the KFE step only involves computing one eigenvector of a sparse matrix once, we can do it on a very large matrix in reasonable time. We can thus leverage the smoothness and natural interpolability of the spectral solution to compute the savings function on a "high definition" grid, for which we take $I_{HD} = 10 \times I = 1,000$ and $J_{HD} = 10 \times J = 400$ so that $I_{HD} * J_{HD} = 400,000$.

Although this matrix is quite large, the whole procedure remains very fast: it takes 20.3 ms to set up the grid and interpolate the spectral savings function, 156 ms to build the differentiation matrices

¹¹For a given runtime

Figure 19: Stationary Distributions

Finite Differences



Spectral + Finite Volume

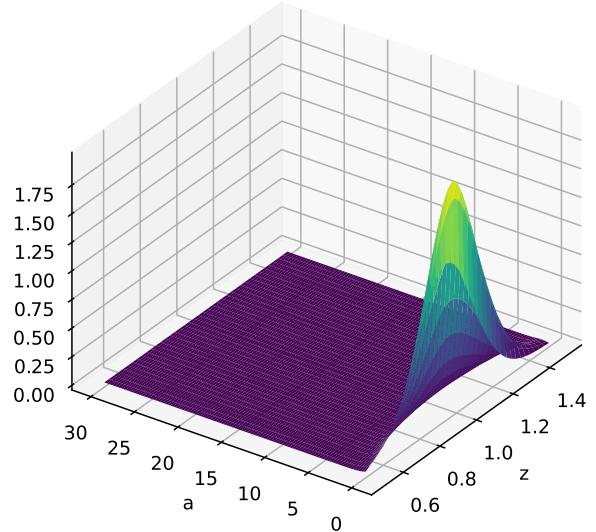
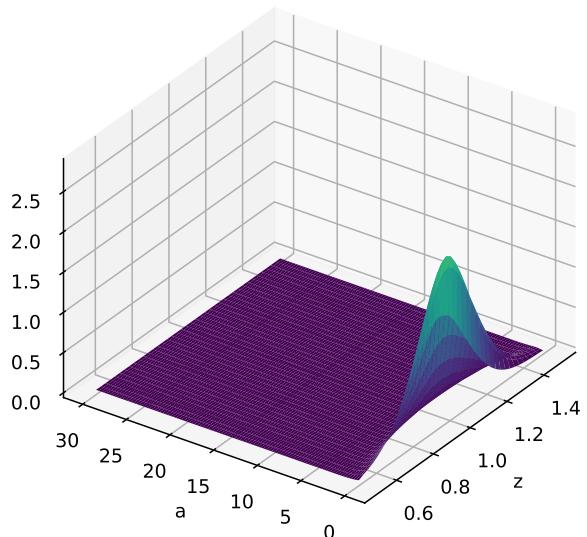
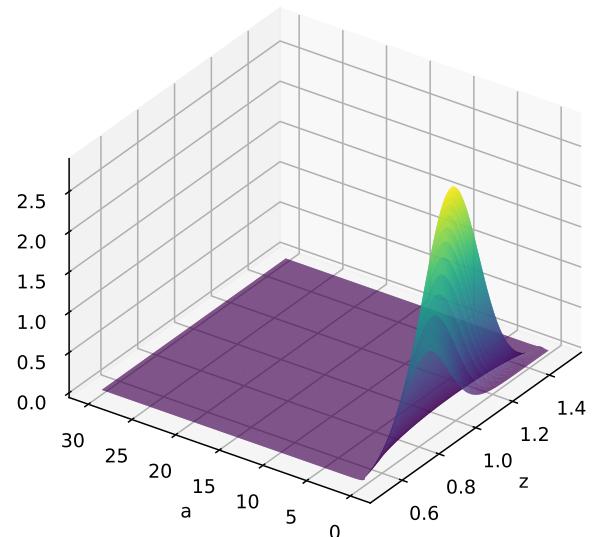


Figure 20: Stationary Distributions

Spectral + Finite Volume



Spectral + High Definition Finite Volume



and 9.08 s to assemble \mathbf{C}_{HD} and compute its zero eigenvector. These times only have to be added to the very fast spectral time for the HJB. This yields an estimate of aggregate capital of $k_{Sp}^H D = 0.5372$. Figure 20, which contrasts the stationary distribution obtained via finite volumes and high definition finite volumes, shows that this is largely driven by the mass of agents close to the credit constraint: as in the baseline calibration the asset grid is now relatively coarse, it accounts quite poorly for the effects of the mass of agents at the borrowing constraint. Increasing the resolution of the grid allows us to better approximate this mass.

Coming back to finite differences, this is a cause for concern: as the savings functions are very similar and we are using the same method to solve the KFE, this same coarseness should plague the finite difference solution, and it does. If we reimplement finite volumes on the "high definition" grid, i.e. with 10 times more points in each dimension than the baseline specification reported in Achdou et al. (2021), (a) a code that runs much more slowly, because the HJB now has to be solved iteratively on a much finer grid, taking 1min 25s in total, (b) a much lower estimate for aggregate capital of $k_{FD}^{HD} = 0.5094$. Again, it should be stressed that increasing the resolution in both dimension ten-fold decreases the estimate for capital by -31%, *a worrying sign*. Moreover, this new estimate is quite close to the one obtained using the spectral method for the HJB and a high resolution grid for the KFE, which ran in far less time.

A natural avenue would be to solve the HJB on a fine grid using finite differences and then the KFE on an even finer grid, but that would largely defeat the elegance and time gain of the transposition approach to the KFE, which crucially requires using the same grid for both. It should be stressed that this resolution issue plagues the spectral and the finite differences approach equally, but that it is far more easily circumvented under the spectral approach, because it is much easier to combine with a high-definition grid for the KFE stage only.

In the end, the case of the Hugget model with diffusive income again shows that the spectral approach has a number of advantages, accompanied by a number of disadvantages which speak in favour of using finite differences, but that certain contexts and issues can decidedly play to its strengths. Almost poetically, it seems that the best-performing approach uses a combination of both.

6. Spectral method for a life-cycle model

To illustrate the power of the spectral approach for smooth, higher-dimensional problems, we now turn to a heterogeneous-agent problem with a life-cycle.

6.1. A simple life-cycle model

To build a vaguely plausible model of heterogeneous agents featuring a life-cycle, we first change the income process from a Ornstein-Uhlenbeck process to an exponential Ornstein-Uhlenbeck process, which generates a log-normal stationary distribution. Put differently, we now assume that the log of income z_t follows an Ornstein-Uhlenbeck process:

$$d \ln z_t = \theta(\ln \hat{z} - \ln z_t) dt + \sigma dB_t \quad (62)$$

It follows that log-income has a normal stationary distribution $\ln z \sim \mathcal{N}(\ln \hat{z}, \frac{\sigma^2}{2\theta})$. A simple application of Ito's Lemma yields the drift and diffusion corresponding to z_t ,

$$dz_t = \left(\theta(\ln \hat{z} - \ln z_t) + \frac{\sigma^2}{2} \right) z_t dt + \sigma z_t dB_t , \quad (63)$$

To incorporate the life-cycle, we assume that agents live (or, rather, are economically active) from period $\underline{t} = 25$ to $\bar{t} = 65$. We also incorporate the age-income gradient by adding a geometric term

$\psi_t z_t$ to the income drift. We add the same drift to the median of the exponential Ornstein-Uhlenbeck process $\ln \dot{z}_t$.

To admit a stationary distribution, our model needs to feature some kind of reinjection process for agents who die at \bar{t} , which can naturally be interpreted as the birth of a descendant. To do this in the simplest possible way, we assume that agents who die with wealth and income $(z_{\bar{t}}, a_{\bar{t}})$ are reincarnated at $(\frac{z_{\bar{t}}}{(1+\psi)^{\bar{t}-\underline{t}}}, a_{\bar{t}})$. This means that agents transmit to their offspring their current income, deflated by the age-income gradient as descendants are now young, as well as their current wealth, without any deflation.

Myriad more complex reinjection processes could be imagined. In fact, the main constraint is that to solve for the stationary distribution using finite volume, one has to keep matrices sparse i.e. the process local: a reinjection distribution would be hard to account for numerically. The pseudospectral solution to the HJB equation uses full matrices and models reinjection quite flexibly (see below), so that far more realistic mechanisms could easily be implemented.

Finally, we have to define a terminal value given a certain wealth and income at the moment of death. We model it quite simply by hypothesizing that dying agents receive a fraction $\alpha \in [0, 1]$ of the value they leave to their descendants, where higher α implies higher altruism. This translates into a very simple, quasi-periodic boundary condition:

$$v(a, z, \underline{t}) = v\left(a, \frac{z}{(1+\psi)^{\bar{t}-\underline{t}}}, \bar{t}\right) \quad \forall a, z \quad (64)$$

In the end, the heterogeneous-agent model with a life-cycle gives rise to the following Hamilton-Jacobi-Bellman equation:

$$\rho v(a, z, t) = \max_c u(c) + (z + ra - c)\partial_a v(a, z, t) + \mu(z)\partial_z v(a, z, t) + \frac{\sigma^2(z)}{2}\partial_{zz} v(a, z, t) + \partial_t v(a, z, t) \quad (65)$$

where $\mu(z) = (\theta(\ln \dot{z}_t - \ln z_t) + \frac{\sigma^2}{2} + \psi)z$ and $\sigma(z) = \sigma z$, to which one should add the borrowing constraint (51), the reflecting boundary constraints (52) and the terminal value constraint (64). The Kolmogorov Forward Equation is now slightly more involved and writes:

$$0 = -\partial_a[s(a, z, t)g(a, z, t)] - \partial_z[\mu(z)g(a, z, t)] + \partial_{zz}\left[\frac{\sigma^2(z)}{2} - g(a, z, t)\right] - \partial_t g(a, z, t), \quad (66)$$

coupled with the reinjection condition:

$$g(a, \frac{z}{(1+\psi)^{\bar{t}-\underline{t}}}, \bar{t}) = g(a, z, \underline{t}) \quad \forall a, z \quad (67)$$

Even though our model of income, savings and life-cycle is very primitive, we have tried to tune parameters to roughly match the U.S. household income distribution. As log-income will follow $\ln z \sim \mathcal{N}(\ln \dot{z}, \frac{\sigma^2}{2\theta})$, the median income will be \dot{z} and the mean income $\exp(\ln \dot{z} + \frac{\sigma^2}{4\theta}) = \dot{z} \exp(\frac{\sigma^2}{4\theta})$. The U.S. Census Bureau estimates households' median income at \$68,703 and mean income at \$98,088 in 2019 (Donovan et al., 2021), which we round up to \$70,000 and \$100,000. Using high-quality administrative data, Bloom et al. (2017) estimate a 90th-10th percentile difference in yearly log income growth of around 0.8 in the early 2010s, which we use to calibrate σ (by assuming that these shocks are normally distributed, which they are clearly not). Writing Φ^{-1} the inverse normal PDF, we thus obtain three

calibrated parameters:

$$\dot{z} = 70,000 \quad \sigma^2 = \left(\frac{0.8}{\Phi^{-1}(0.9) - \Phi^{-1}(0.1)} \right)^2 \approx 0.0974 \quad \theta = \frac{\sigma^2}{4 \ln(70,000/100,000)} \approx 0.0683 \quad (68)$$

We assume that all of these apply to income at age $\frac{\bar{t}+\bar{t}}{2} = 45$. In Guvenen et al. (2021), authors again use administrative data to compute growth rates for median income over the 25-55 age period (Table D1), which add up to $37.04 + 22.12 - 1.29 = +57.87\%$ for the cohort born in 1980 across both genders. We hypothesize that this trend would have continued over ages 45-65 (which is probably optimistic). Rounding this up to $+60\%$ and annualizing yields a calibration of:

$$\psi = 1.6^{1/30} - 1 \approx 1.579\% \quad (69)$$

Finally, we use an altruism parameter $\alpha = 0.1$, which we simply made up.

6.2. Solving the life-cycle Hamilton-Jacobi-Bellman equation

Solving the HJB equation (65) is very similar to the previous examples, except that one now has to add a third dimension for time, and tensorize collocation nodes accordingly. We will thus focus on a few specific points of attention. As there is now an explicit time dimension, the HJB iteration algorithm does not admit a natural "walking back in time" interpretation, though it can be roughly thought of as living the same life multiple times, with each iteration being a reincarnation.

One potential difficulty is the treatment of the reinjection between \bar{t} and \underline{t} , which translates into a boundary condition on v . Naively, one would like to formulate this as a condition linking the time-edge nodes, but the income reinjection targets (i.e. the $z_j/(1+\psi)^{\bar{t}-\underline{t}}$'s) have every reason not to lie on a node $z_{j'}$. One thus needs a reliable, and ideally fast, procedure to interpolate the (tentative) value function at the income-reinjected nodes.

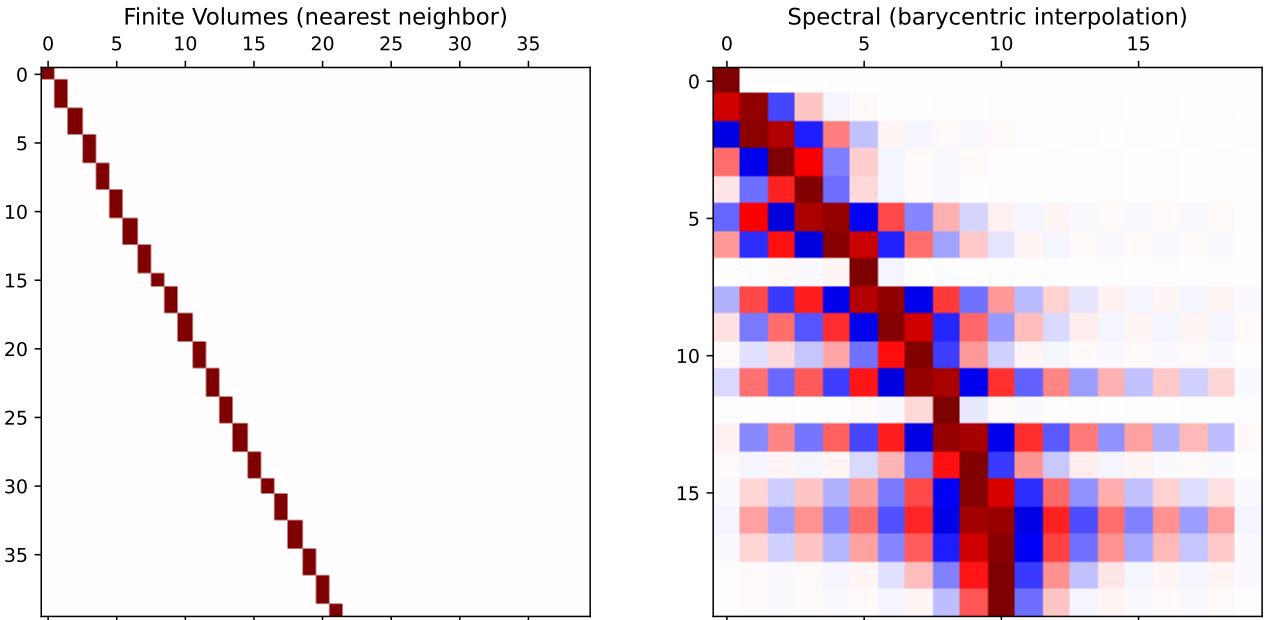
As the reinjection targets are known in advance (because the age-income gradient and the income discretization nodes are known in advance), one can use the barycentric interpolation formula to pre-compute this polynomial interpolation as a linear operation. As we are interpolating on Chebyshev nodes, interpolation weights can even be computed analytically, which makes the whole procedure even faster and more robust numerically. This can then be assembled into an interpolation matrix \mathbf{R} , and we can enforce the boundary condition by imposing $\mathbf{v}^{k+1} = \mathbf{R}\mathbf{v}^k$ on the appropriate \underline{t} nodes.¹²

Figure 21 contrasts the offspring matrices in both approaches. In the spectral approach (right panel), the global nature of the matrix is clearly visible as it has mostly non-zero entries: the precision gain from (exact) polynomial interpolation comes at the cost of sparsity. However, R is quick to assemble thanks to analytical formulae and can be preassembled (i.e. it is the same at each iteration). On the other hand, the finite volume reinjection matrix, which we will use to solve the Kolmogorov Forward Equation, is based on matching each reincarnation node with its nearest neighbour, which is significantly less precise but conserves sparsity. Slightly more involved local interpolation schemes, e.g. linear or low-order polynomial interpolation could be used instead, but any higher-order method will make the matrix non-sparse and thus the procedure impractically slow.

For the distribution to be representable numerically, we have to work with a bounded interval $[\underline{z}, \bar{z}]$ with reflecting boundaries. We thus have to ensure that reinjections remain in this interval. In the numerical implementation, we thus use $z_{\bar{t}} \rightarrow z_{\underline{t}} = z_{\bar{t}} - \bar{z}/(1+\psi)^{\bar{t}-\underline{t}} + \underline{z}$, i.e. we only deflate the income above the lower boundary. This is equivalent to (67) as $\underline{z} \rightarrow 0$. We could alternatively have used a max operator, i.e. $z_{\bar{t}} \rightarrow z_{\underline{t}} = \max(\bar{z}, z_{\bar{t}}/(1+\psi)^{\bar{t}-\underline{t}})$ but that makes the terminal value locally constant, which is both numerically and economically undesirable.

¹²This is, again, an explicit approach to the boundary condition. One could also enforce it implicitly by imposing $\mathbf{v}^{k+1} = \mathbf{R}\mathbf{v}^{k+1}$, but we have found the explicit approach to be more stable, whereas the end result should be virtually the same since it will be within the convergence tolerance.

Figure 21: Offspring Matrices R , R_{FD}



In practice, we use $N = 30$ nodes for assets, $M = 15$ for incomes and $O = 10$ for time periods. All other parametrization are indicated in Table 1. As the total dimension now becomes quite large ($N * M * O = 4,500$), iterations become slower. An easy way to speed up the convergence of the code is to first compute the solution of the static problem at age 45 and then use it as an initial guess for the life-cycle problem. On a standard laptop, solving the static problem takes 640 ms and then solving the life-cycle problem takes 18min 14s.

Figures 22 shows the corresponding value function. In the early periods, it has roughly the same shape as in the static model; however, as one becomes older, differences in utility become smaller: as one has limited altruism for one's descendants ($\alpha \ll 1$), the terminal value from transmitting one's wealth is low, and the same goes for income (which will, moreover, drop). A keen eye could, perhaps, make out that the first and last-period utilities are (non-homothetic) linear transformations of one another.

Figures 23 and E.1 respectively plot the associated savings and consumption functions. One can see that the young tend to save across most asset and income states, that the wealthy middle-aged dis-save, and that most of the old dis-save (except for the very income-rich, asset-poor). All of these are largely in line with intuition.

6.3. Solving the life-cycle Kolmogorov Forward equation

Solving for the Kolmogorov Forward Equation via finite volumes is very straightforward. As the process is now even more involved, we have again not attempted to implement any hybrid spectral / finite volume scheme.

The only changes to the finite volume procedure are the three-dimensional tensorization of differentiation matrices and the incorporation of the reinjection. As discussed above, we have chosen the easiest route, which was to match each reinjection target $z_j/(1+\psi)^{t-t_0}$ with its nearest node $z_{j'}$. One could also have chosen a linear interpolation scheme and interpreted it as a transition with a Bernoulli distribution over the two neighbouring nodes. In any case, this results in a sparse but less precise offspring matrix R_{FD} shown in Figure 21.

As solving the KFE only involves computing one sparse matrix eigenvector once, we can do that

Figure 22: Value over the Life-Cycle

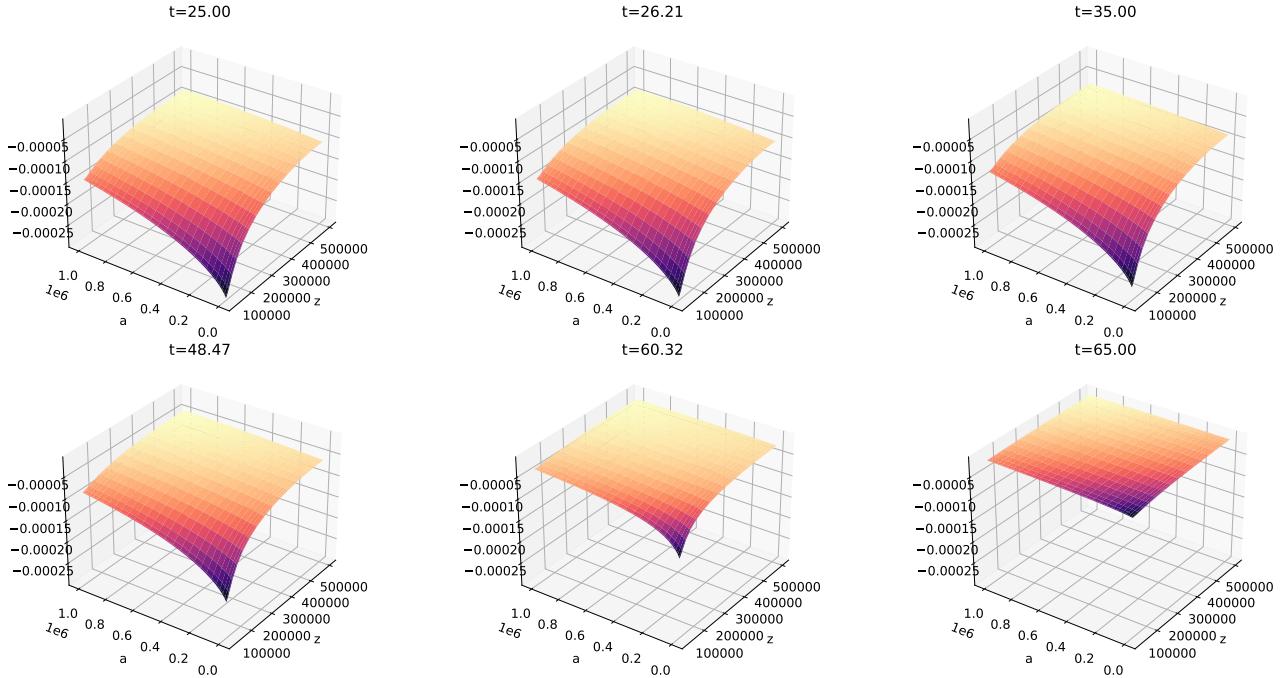
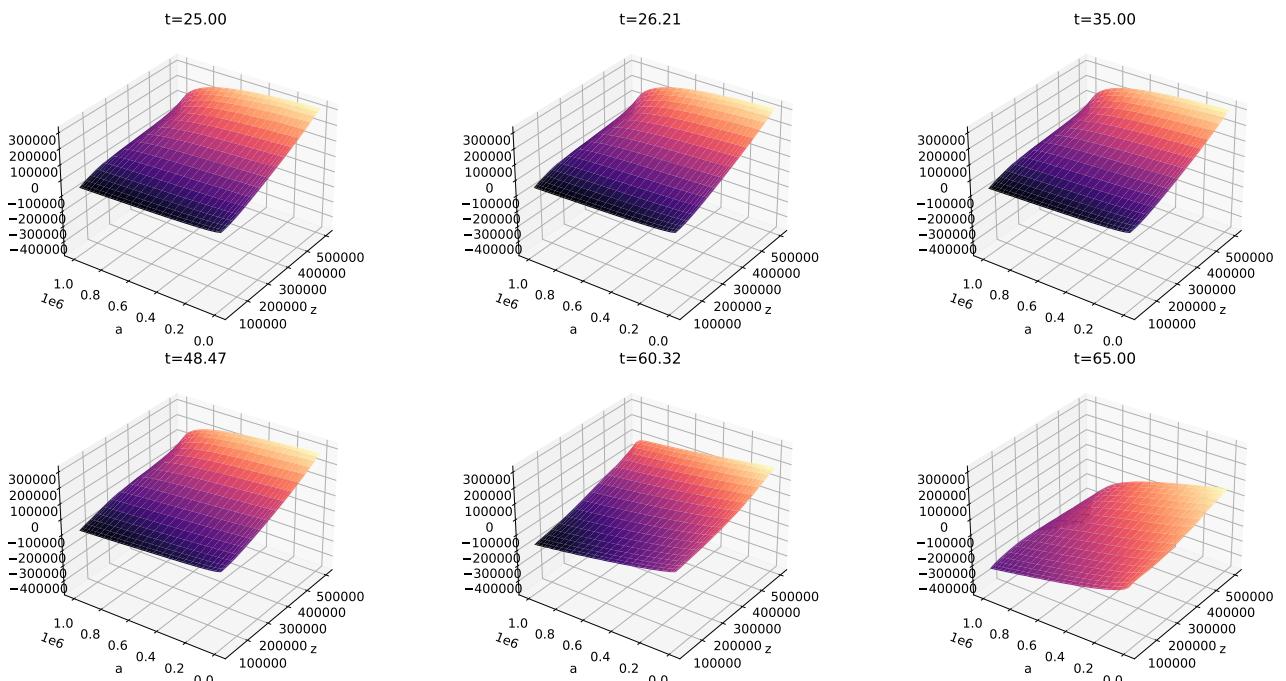


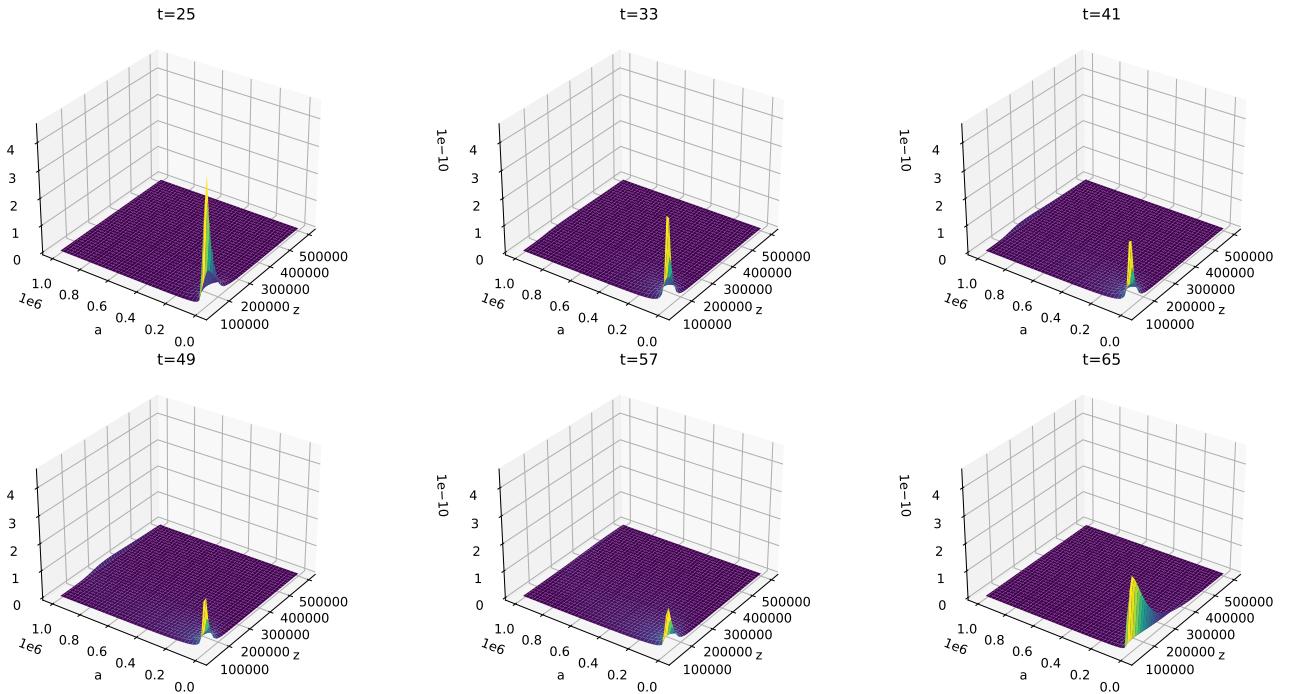
Figure 23: Savings over the Life-Cycle



on a very large sparse matrix in very little time, which is why we chose $I = 80$ nodes for assets, $J = 40$ nodes for income and $K = 41$ nodes for age (i.e. one node per year). This yields a total size of $I * J * K = 131,200$, which is very large but can still be solved fairly quickly. Solving both the HJB and KFE equation via finite differences would have required a lower resolution, as the HJB equation relies on a much-slower iterative scheme; alternatively, one could have solved the HJB on a (relatively) coarse FD grid, interpolated the results to a denser one and then solved the KFE on the second grid, which would have meant re-assembling all differentiation matrices etc. Combining the spectral and finite volume approach is thus not much more involved, and allows us to leverage the smoothness of the spectral solution in the interpolation step.

Figure 24 shows the stationary distribution of agents across their life cycle. Along the income dimension, the mass of agents shifts towards higher incomes as age increases because of the age-income gradient; in turn, mass is much more condensed around low incomes in the early periods of life because of the income-deflating offspring reinjection. Along the wealth dimension, one can see that most agents begin their life with relatively little income, then progressively save up depending on their income realization, and that in the final instant of life there are relatively few asset-rich agents, because the utility gained from transmitting it is comparatively low.

Figure 24: Distribution of Wealth and Income over the Life-Cycle

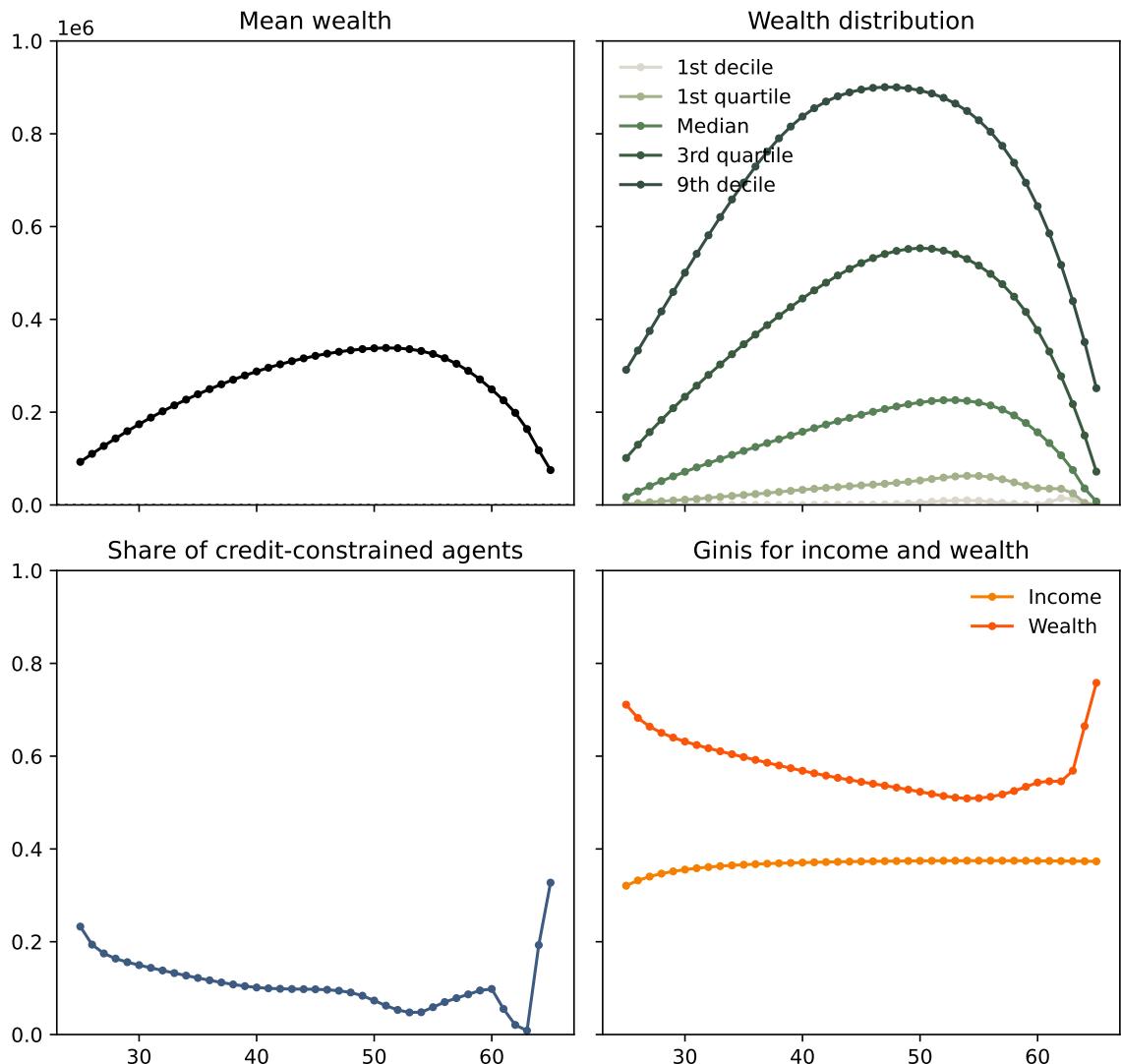


Finally, Figure 25 plots a number of simple distributional moments over the life cycle. As expected, we can see that mean wealth increases and then decreases, peaking around \$400,000 around age 50. The quantiles of the wealth distribution all follow a similar pattern, though the peak is earlier for wealthier households: the top decile wealth peaks at around \$900,000 around 45, that of the top quartile around \$500,000 around 50 and that of the median around \$200,000 around 55. Interestingly, the wealth of the lower quantiles is much flatter: the lower quartile does not build up a sizeable capital at all during its lifetime, while the lower decile is essentially always credit constrained. This is also reflected in the lower left panel: only in the latest periods of life does the share of credit-constrained agents briefly fall significantly below 10%, before rising again just before death. Income inequality, measured by the Gini coefficient of the income distribution, is mostly flat over the whole period, as expected. On the other hand, wealth inequality decreases up to around age 55, when it starts to

significantly increase again: over their life-time, agents thus accumulate more and more homogeneous levels of wealth, but death and bequests break this slow convergence. In the end, wealth inequality remains high, and particularly much higher than income inequality, over the whole life cycle.

Although it is far from realistic, this simple example shows that even a very crude model of life-cycle savings set in continuous time can generate interesting inequality dynamics as well as high and time-varying proportions of credit constrained agents. In turn, it shows that the spectral approach can be a useful tool to help solve these higher-dimensional heterogeneous-agent models.

Figure 25: Distributional Moments over the Life-Cycle



7. Application: "slow transitions" with wealth?

As a last illustration of the added value of the spectral approach, we can study the transition speeds between inequality regimes in the footsteps of Gabaix et al. (2016). In this article, Gabaix, Lasry, Lions & Moll show that while there are many theories to explain the Power tail of income distributions, these all generate slow transitions between stationary distributions that are incompatible with the fast rise in income inequality observed in the second half of the 20th century across advanced economies.

More precisely, they write down a model of income with random growth, and show that (i) it generates a Power law-distributed income, (ii) the eigenvalues of the generator of that income process, which control the transition speed between stationary distributions, can be characterized analytically and (iii) that for all reasonable parametrizations, these generate "slow transitions". In their main calibration, they obtain a half-life of ~ 20 years for transitions of the income distribution; in Appendix E, authors then write down a model of (uncontrolled) random growth for wealth, which also generates a Power tail, and yields a half-life of ~ 26 years for the wealth distribution.

We can, modestly, extend this fascinating work to an economy with income *and* wealth inequality. As linking both via a (constrained) savings problem becomes very difficult to solve analytically, we have to approach it numerically. Luckily, this type of analysis is greatly simplified by the combined spectral-finite volume approach to heterogeneous-agent models: we can use the speed and smoothness of the spectral method for the HJB to solve the savings problem over a fairly large state space, and then the finite volume approach to the KFE to obtain an approximation of the (transpose of the) generator of the income-savings process. Computing the eigenvalues, and especially the spectral gap, of this generator is then fast and easy.

7.1. A simple model of Power-law income with credit-constrained savings

We start off by writing a standard "random growth" model of income, in which log-income follows a simple Brownian motion with drift:

$$d \ln z_t = g dt + s dB_t \quad (70)$$

For income to admit a stationary distribution, one needs to add some kind of "stabilizing force". Although many forms and variations have been proposed, we chose the simplest route, which involves adding a reflecting boundary at some lower bound \underline{z} : agents whose income crosses that level are brought back to \underline{z} . To be able to represent the distribution on a computer, we also have to add an upper reflecting boundary \bar{z} ¹³.

From the Kolmogorov Forward equation associated with (70), one can show that the stationary income distribution follows

$$f(x) = Cx^{-\zeta-1} \quad \text{with} \quad \zeta := 1 - \frac{2g}{v^2}. \quad (71)$$

C is a normalizing constant which can be computed by noting that

$$\int_{\underline{z}}^{\bar{z}} z^{-\zeta-1} dz = \left[\frac{z^{-\zeta}}{-\zeta} \right]_{\underline{z}}^{\bar{z}} = \frac{\underline{z}^{-\zeta} - \bar{z}^{-\zeta}}{\zeta}, \quad (72)$$

so that $C = \frac{\zeta}{\underline{z}^{-\zeta} - \bar{z}^{-\zeta}}$. We can then compute the mean income \check{z} :

$$\check{z} := \int_{\underline{z}}^{\bar{z}} z C z^{-\zeta-1} dz = C \left[\frac{z^{1-\zeta}}{1-\zeta} \right]_{\underline{z}}^{\bar{z}} = \frac{\zeta}{\zeta-1} \frac{\underline{z}^{1-\zeta} - \bar{z}^{1-\zeta}}{\underline{z}^{-\zeta} - \bar{z}^{-\zeta}} \quad (73)$$

¹³Which is admittedly a bit awkward in discussions of tail inequality.

Note that taking $\bar{z} \rightarrow \infty$ yields the well-known $\check{z} = \frac{\zeta}{\zeta-1} z$. See Gabaix (2009) for a far more detailed treatment.

We use these relations to calibrate model parameters. We take $z = \$50,000$ and $\bar{z} = \$1,000,000$, which have no particular economic interpretation and are mainly numerical parameters: we have chosen an as wide as possible income range that keeps the model estimable. As in the life-cycle model, we take $\check{z} = \$100,000$. Solving (73) numerically yields $\zeta = 1.87$. Keeping $s^2 = 0.0974$ from the life-cycle model, (71) implies a log-income growth rate of $g = -0.0422$.

Finally, we have to choose an asset support, for which we pick $a = -z = -\$50,000$, i.e. we allow agents to borrow up to a lower-bound annual income, and $\bar{a} = 50 \times \bar{z} = -\$50,000,000$, i.e. agents can save up to 50 upper-bound annual incomes. As we are only solving a two-dimensional problem and care less about execution time than precision here, we choose $N = 70$ and $M = 80$. Similarly, for the finite volume grid in the KFE equation, we use high resolutions of $I = 300$ and $J = 300$.

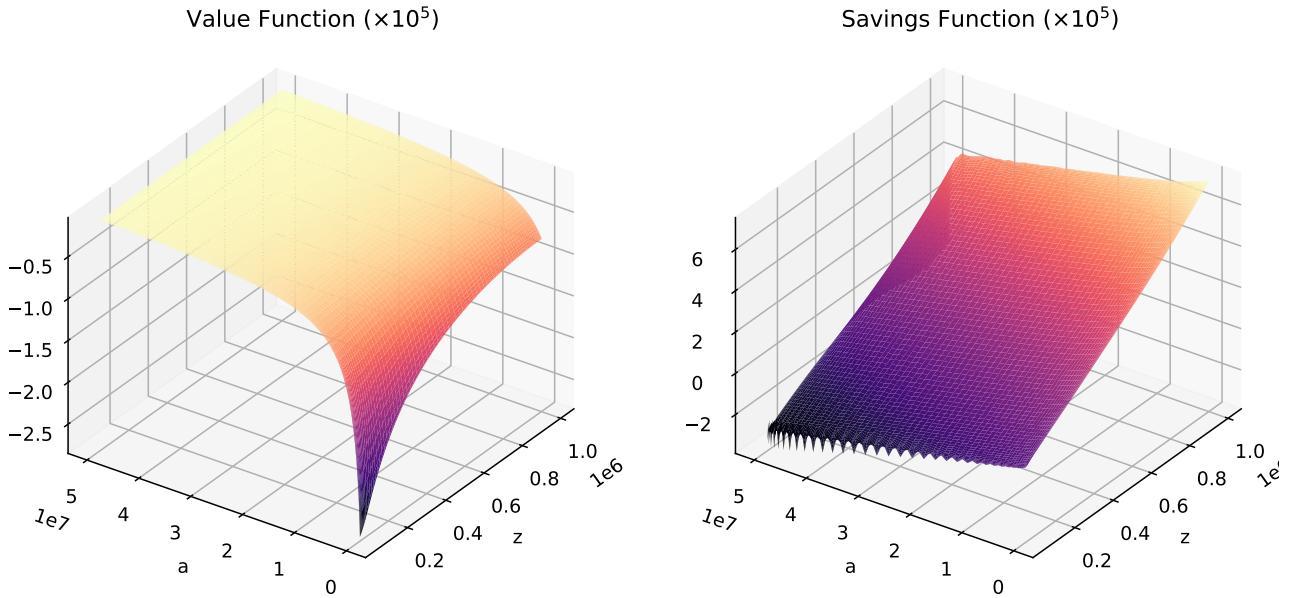
We note that Ito's Lemma yields:

$$dz_t = gz_t dt + sz_t dB_t, \quad (74)$$

so that we use $\mu(z) = gz$ and $\sigma(z) = sz$ in the HJB and KFE equations. Except for the larger income and asset range, both equations are solved exactly as in the previous applications.

The value and savings function obtained from the HJB equation are shown in Figure 26. The downward-bending part around \bar{a} in the savings function reflects the upper borrowing constraint $a_t < \bar{a}$. The oscillations in the savings function along z are also relatively large because of the sharp curvature in the value function around (\underline{z}, \bar{a}) , but should have a limited impact on the rest of the analysis because they concern a region which has little mass, i.e. very few agents are extremely asset-rich and extremely income-poor at the same time.

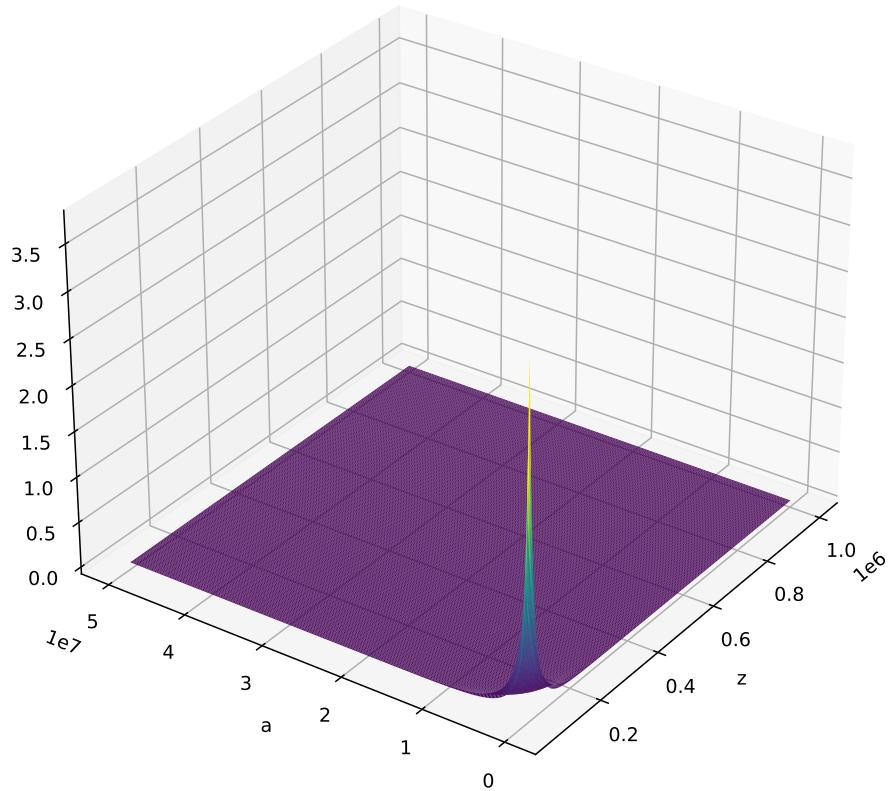
Figure 26: Value and Savings Functions for Credit-Constrained Savings under Power Law Income



This intuition is confirmed by the stationary distribution of agents obtained by solving the KFE equation using finite volumes, which is shown in Figure 27. To check if income is Power-law distributed, and to better analyse their shape, we have plotted the cross-section of both distributions, as well as their log-log transform, in Figure 28. The income distribution exhibits the typical straight line,

confirming that it follows a Power law. The wealth distribution however has a downward sloping log-log density, which means that it has thinner tails than a Power law. This could reflect the artificial upper asset constraint, but also deeper economic behaviour, especially since its shape is already rounded far below the upper barrier.

Figure 27: Wealth and Income Distribution for Credit-Constrained Savings under Power Law Income



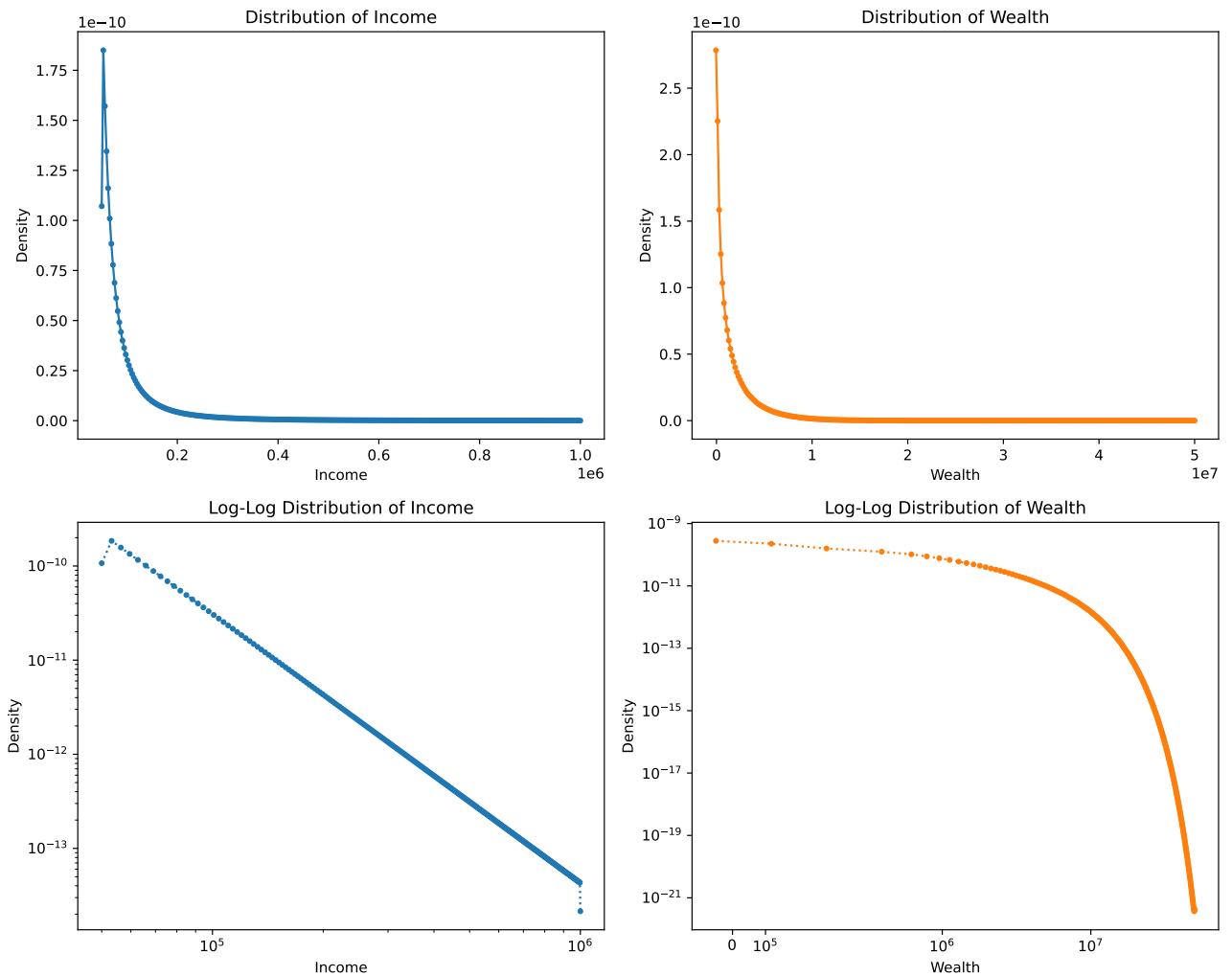
7.2. Characterizing transition speeds using the spectral gap

As we have stressed in the previous sections, the finite difference / finite volume discretization of the KFE is, at its core, a discretization of the (adjoint of the) generator \mathcal{A} of the continuous-time, *continuous*-state income-savings process. The resulting discretized operator is a matrix A that has all the features of a continuous-time, *discrete*-state Markov process generator: negative diagonal entries, positive off-diagonal entries and rows that sum to zero. We can thus interpret it as the transition rate matrix of an approximation of the income-savings process.

In particular, the dynamics of this process should broadly reflect the characteristics of the underlying continuous process, especially when the number of discretization points becomes large. This can probably be made more precise, but is beyond the scope of this paper. For our purposes, it is enough to know that the transition speeds of the discretized process can give us *an idea* of the transition speed of income and wealth inequality in a continuous-state economy.

As the discretization of the KFE gives us a generator matrix A , it is tempting to look at its eigenvalues. Indeed, in the "ergodic case" of their Proposition 1, Gabaix et al. rely on the spectral gap of the operator \mathcal{A} to characterize the speed of the L^1 convergence from an initial distribution g_0

Figure 28: Cross-sections for Credit-Constrained Savings under Power Law Income



to the steady-state g_∞ . However, linking the spectral gap of \mathcal{A} to an L^1 convergence requires some additional assumptions and a fair bit of mathematical work.

As our goal is to follow the easy route, we only compute and plot the eigenvalues of A , but do not claim to give it any particular interpretation. A is a $90,000 \times 90,000$ matrix, but its smallest eigenvalues can be computed very quickly using sparse matrix routines. To compare the income-savings process to the pure income process, we also discretize a KFE for the income process only, which yields a 300×300 matrix A_{income} . Results for both generators are presented in Figures 29 and F.1. Gabaix et al.'s analytical formula for the spectral gap of the Power-law income generator $\mathcal{A}_{\text{income}}$ is $l_{\text{income}} = \frac{1}{2} \frac{g^2}{s^2} = 0.0091$, which is quite close to the spectral gap of the discretized A_{income} , which is $l_{\text{disc.}} = 0.0073$. Clearly, the first eigenvalues of $A_{\text{income},\text{wealth}}$ are much smaller, both in their real and in their imaginary part, than those of A_{income} .

Less heroically, we focus on the much more natural criterion for Markov chain convergence, the L^2 norm. For non-reversible processes such as ours, which feature a non-self-adjoint generator, it is controlled by a slightly different spectral gap λ . Writing π It is defined using the Dirichlet form of the Markov process \mathcal{E} ,

$$\lambda = \min \left\{ \frac{\mathcal{E}(f, f)}{\text{Var}_\pi(f)}; \text{Var}_\pi(f) \neq 0 \right\} \quad \text{where} \quad \mathcal{E}(f, g) := \Re(\langle (I - K)f, g \rangle) \quad (75)$$

and where π is the stationary distribution of the process. In our discrete-state case, it can more simply be expressed as the second-smallest eigenvalue of the matrix $E := \frac{1}{2}(A + A^*)$, where A^* is the adjoint of A on $l^2(\pi)$ i.e. $A(x, y) = \frac{\pi(y)}{\pi(x)}K(y, x)$. Writing the semigroup of the process $H_t = e^{-tA}$, one can show that its convergence to the stationary distribution satisfies:

$$\forall f \in \ell^2(\pi), \quad \|H_t f - \pi(f)\|_2^2 \leq e^{-2\lambda t} \text{Var}_\pi(f) \quad (76)$$

This in turn implies the extremely useful corollary about probabilities:

$$|H_t(x, y) - \pi(y)| \leq \sqrt{\pi(y)/\pi(x)} e^{-\lambda t} \quad (77)$$

Tighter bounds, for example using the log-Sobolev constant, can also be obtained, but we focus on the simplest results. See e.g. Saloff-Coste (1996), pp. 326-331, for derivations and a far more extensive discussion of these topics.

In light of these powerful results, the overarching difficulty remains computing the spectral gap λ . Luckily for us, even in the case of the 90,000-state income-savings process, we can assemble E , which is of size $90,000 \times 90,000$, in less than 10 milliseconds and then compute its spectral gap in less than 1 second. Although they have no straightforward interpretation, we can easily compute the next 19 eigenvalues of E for good measure. Results are plotted in Figures 30 and F.2.

To make these values more interpretable, we can compute the associated half-lives as $t_{1/2} = \log(2)/\lambda$. For the Power-law income process only, we obtain a spectral gap of $\lambda_{\text{income}}^{\text{disc.}} = 0.0149$, corresponding to a half-life of $t_{1/2}^{\text{income}} = 46.39$ years. For the joint income-savings process, we obtain a much lower $\lambda_{\text{income, wealth}}^{\text{disc.}} = 0.0003069$, which yields a much higher half-life of $t_{1/2}^{\text{income, wealth}} = 2258.30$ years

Although the precise values are probably highly dependent on the economic calibration and numerical parameters, what matters here is the general impression: that a joint income-wealth process, with Power law income and credit-constrained savings, exhibits *even slower* transitions, considerably deepening the macroeconomic puzzle.

Hopefully, this section serves as an illustration of the kind of simple but (vaguely) interesting analyses one can perform by leveraging the combined powers of the spectral and finite difference approaches.

Figure 29: First 20 Generator Eigenvalues in Different Models

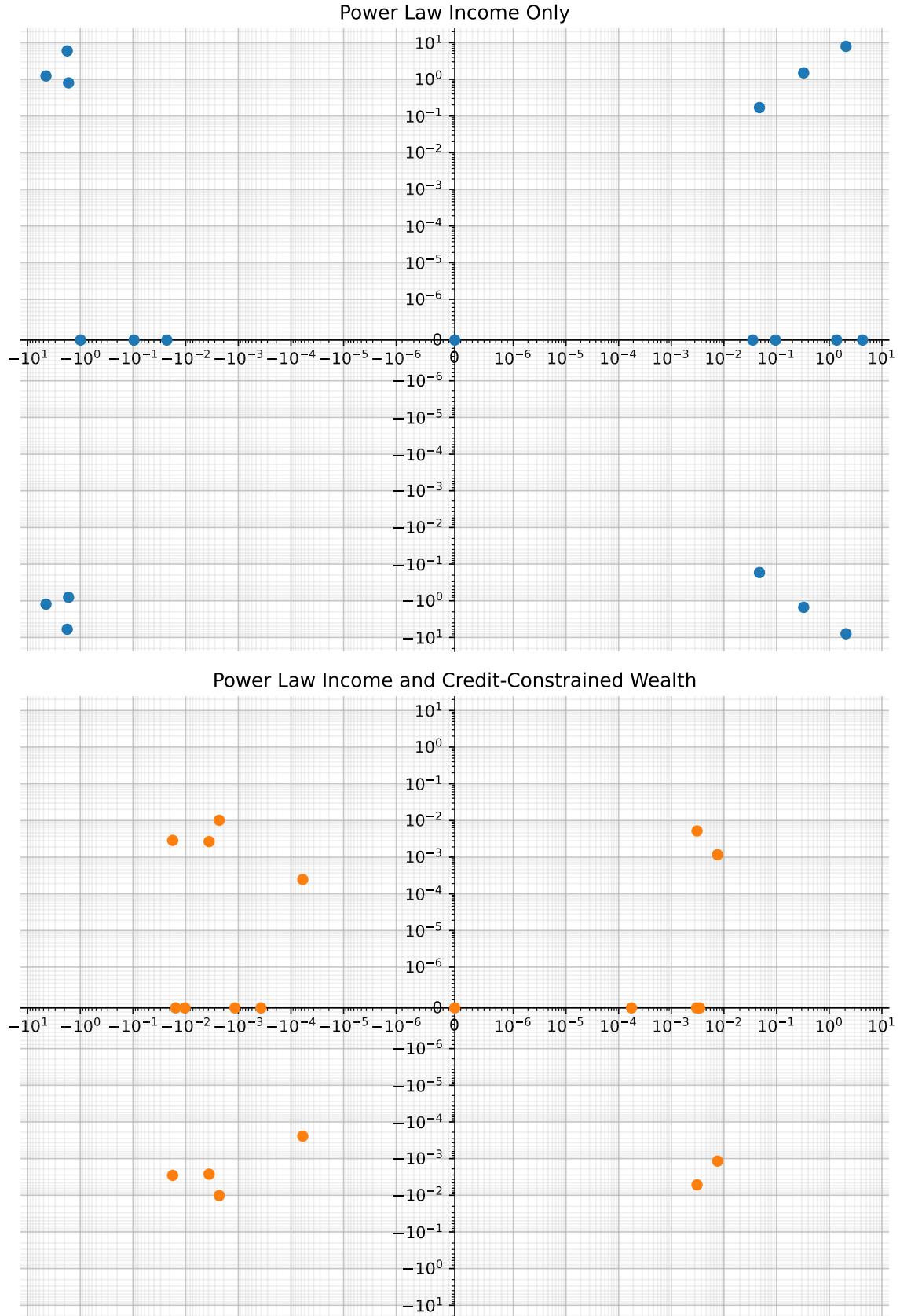
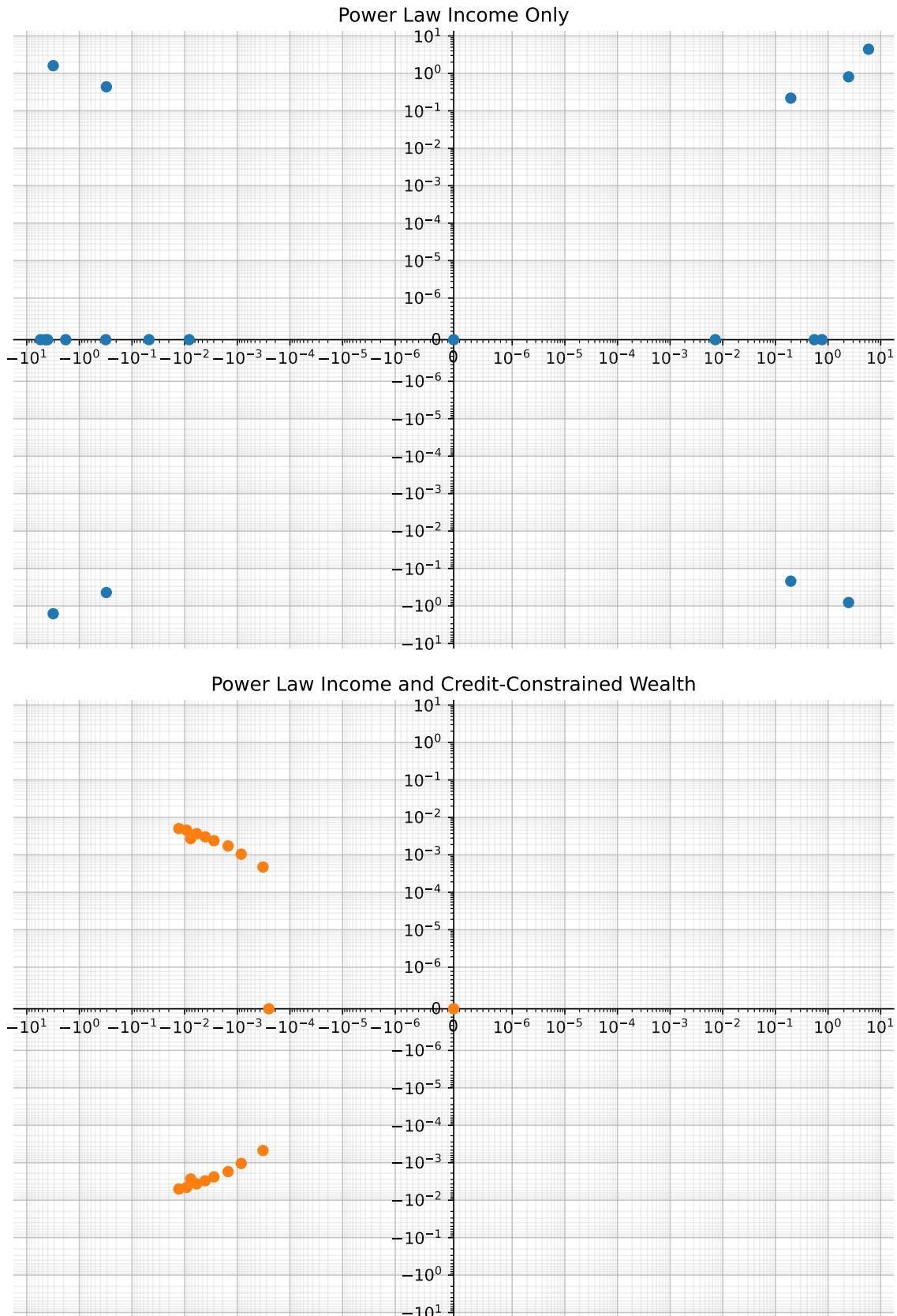


Figure 30: $E := \frac{1}{2}(A + A^*)$'s First 20 Eigenvalues in Different Models



8. Benchmark: are spectral methods actually better?

To assess whether spectral methods actually improve anything compared to finite differences, we have benchmarked both approaches in the case of the two-income model,. We have kept all parameter values and implementation details from this example, but varied the number of asset nodes. Put briefly, for a given number of nodes, the spectral approach is vastly more precise than finite differences, and convergence is much faster; however, it also runs much more slowly (and non-linearly so) because matrices are full. Trading off both effects, spectral methods nonetheless achieve much faster convergence for a given number of nodes.

Throughout this section, it should be stressed that, thanks to the magic of continuous time, all codes are extremely fast: ~ 0.2 seconds for finite differences with $I = 5000$ nodes, ~ 0.6 seconds for collocation with $N = 500$ nodes. Performance comparisons are therefore of very limited practical relevance, but are useful to gauge the relative precision of both approaches for more complex, higher-dimensional problems.

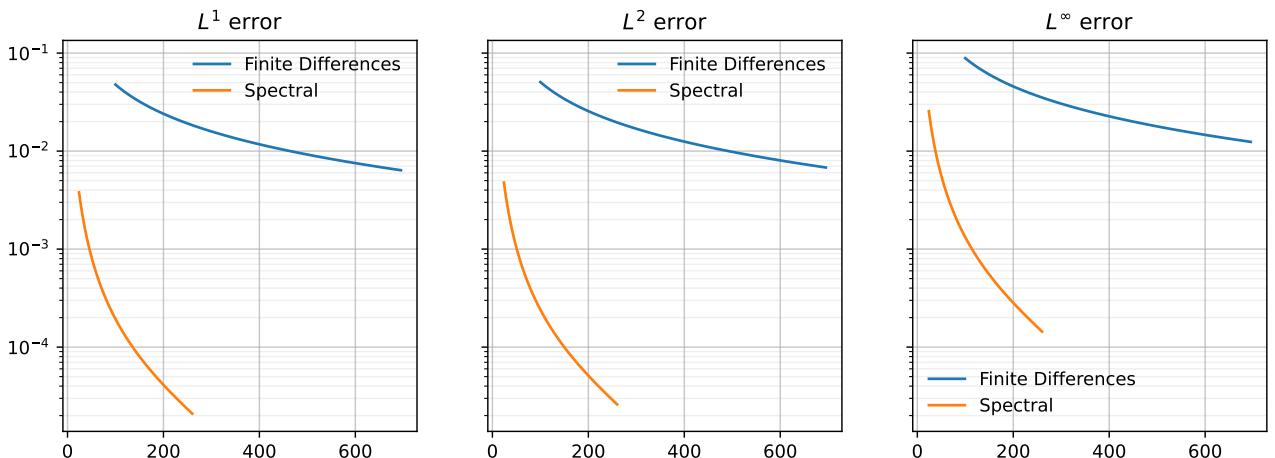
8.1. Errors for a given number of nodes

Figure 31 presents rough estimates of the L^1 (mean), L^2 (mean square) and L^∞ (maximum) error made for the value function, using a given number of nodes in the spectral and finite differences. More precisely, we have run each approach on a high-precision grid ($N = 500$ and $I = 5000$ and respectively), and then compare this solution to lower resolutions. On this and similar figures, we focus on the relevant part which gives us a rough idea of the speed of convergence by looking at resolutions below $N = 250$ and $I = 2500$ respectively.

Clearly, the level of the error curves shows that, even for a relatively low number of nodes, the pseudospectral method achieves remarkably low L^1 and L^2 errors, even compared to high-resolution finite differences. Unsurprisingly, L^∞ errors are larger, though they remain smaller using the spectral approach. Moreover, the slope of the error curves show that spectral methods converge at a much faster rate, especially when the number of nodes is lower.

In the Appendix, Figure G.1 presents the same results for savings, which show the same pattern. The spectral solution to the HJB equation is thus remarkably efficient, both in terms of absolute precision and in terms of convergence speed, which is an encouraging sign for applications using the spectral approach to the HJB and finite differences / finite volumes for the KFE.

Figure 31: Errors in the Value Functions by Number of Nodes



Turning to the stationary distribution, we now have to distinguish the three possibilities to solve the KFE after having solved the HJB equation spectrally: using finite volumes on a fine mesh, using finite volumes on the (coarse, unequally-spaced) spectral mesh and using the hybrid spectral / finite volume approach to deal with the boundary mass. For the high-definition finite volumes approach, we have used $I' = 10 * N$ nodes for each iteration with N spectral nodes. Otherwise, all methods are implemented exactly as described in Section 4.

Figure 32 shows different, economically relevant moments of the empirical distribution (mass of credit-constrained agents, Gini coefficient, total capital) for a given number of nodes. Note that, to make the plot vaguely legible, we have used a different horizontal range for the spectral and finite difference resolutions, with the latter being 10 times larger than the former. For each method, moments are computed differently using the appropriate mix of node-based (linear interpolation, Riemann integration etc.) and polynomial-based (polynomial interpolation, polynomial root-finding, polynomial integration etc.).¹⁴

One can easily see that the dense finite volume and mixed approaches achieve much lower errors for a given number of nodes and converge much faster than the others. Unsurprisingly, coarse finite volumes perform relatively poorly, as it needs $N = 500$ nodes or above to start to compete with the other approaches. Figure G.2 plots the same graph for quantiles of the stationary distribution, exhibiting the same pattern.

In the end, it is thus fairly clear that, for a given number of nodes, spectral methods (using either dense finite volumes or the mixed approach for the KFE) have lower errors and converge much faster than finite differences. Of course, this benchmark is so far a bit disingenuous as, for a given number of nodes, finite differences will be much faster because it uses sparse matrices.

8.2. Runtime for a given number of nodes

We thus turn to comparing the runtimes of the different methods. Figure 33 presents execution times for finite differences and the three spectral approaches. All times are given as the median of seven runs on 2017 MacBook Pro with two 2.3 GHz Intel i5 cores and 8 GB of RAM.¹⁵

Execution times are broken down between the initialization phase (computing and assembling differentiation matrices), the HJB phase, the KFE phase and the computation of moments. For finite differences, the HJB phase makes up essentially all of the computation time; for the spectral approach, the HJB remains dominant but the initialization phase also takes a bit of time. For the dense finite volume and mixed methods for the KFE, one has to recombine differentiation matrices which also takes a visible (but small) amount of time.

The graph makes plainly clear that, for a given amount of nodes, finite differences are much faster to compute (recalling that the horizontal axis is ten times larger on the corresponding subplot than on the others). Moreover, execution time grows more slowly and roughly linearly with the number of nodes, though it becomes more dispersed above $I \approx 3000$. Indeed, recall that the number of non-zero elements in the differentiation and generator matrices grows linearly with the number of nodes.

On the other hand, all spectral approaches are much slower; even worse, they grow non-linearly with the number of nodes. It seems likely that they scale roughly at the matrix inversion complexity $O(N^3)$. This would of course be very bad news, if we didn't have the hope that it might be offset by the higher precision of spectral methods outlined above.

¹⁴As can be seen in Figure 33, polynomial-based method are usually slower than discrete methods to compute moments. However, it seems intuitive that they are more precise, e.g. integrating a polynomial density and computing roots to find the quantiles of a distribution should be more precise than interpolating linearly between nodal masses. It would be interesting to see how much of the precision gain of spectral methods can be attributed to this more secondary advantage, and to what extent they make up for the performance loss.

¹⁵So are all other runtimes dispersed throughout this Master Thesis.

Figure 32: Moments by Number of Nodes

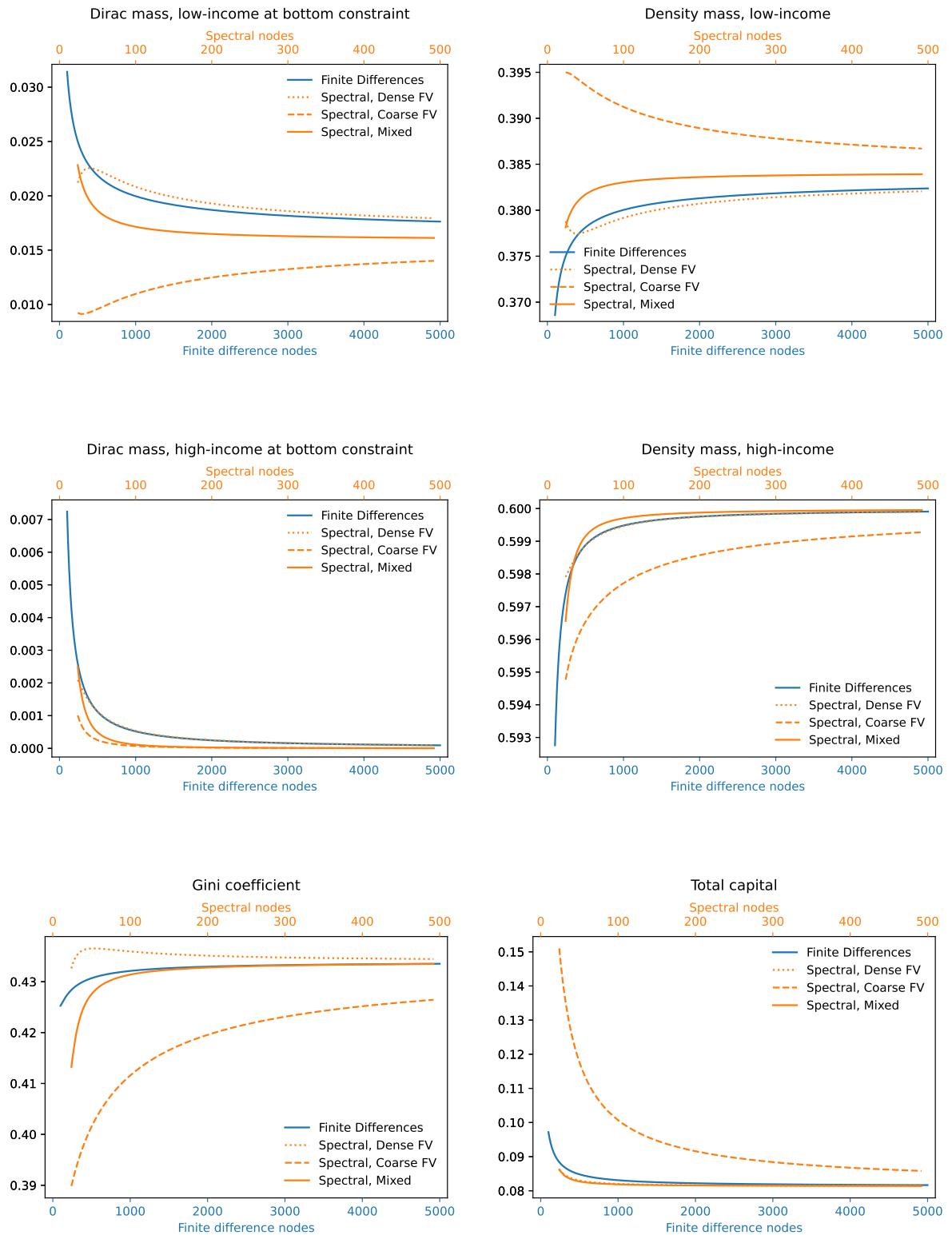


Figure 33: Execution Times by Number of Nodes

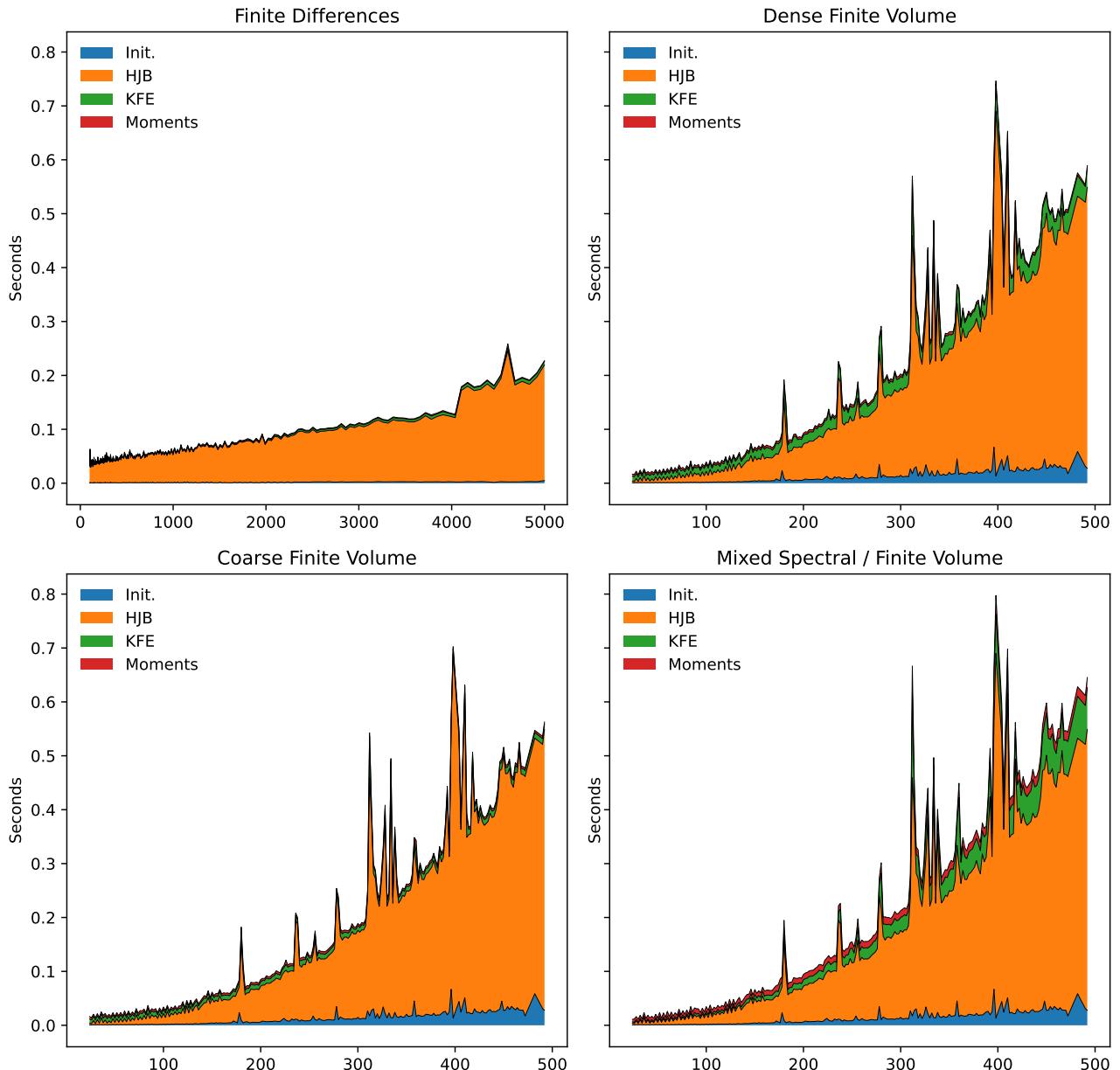
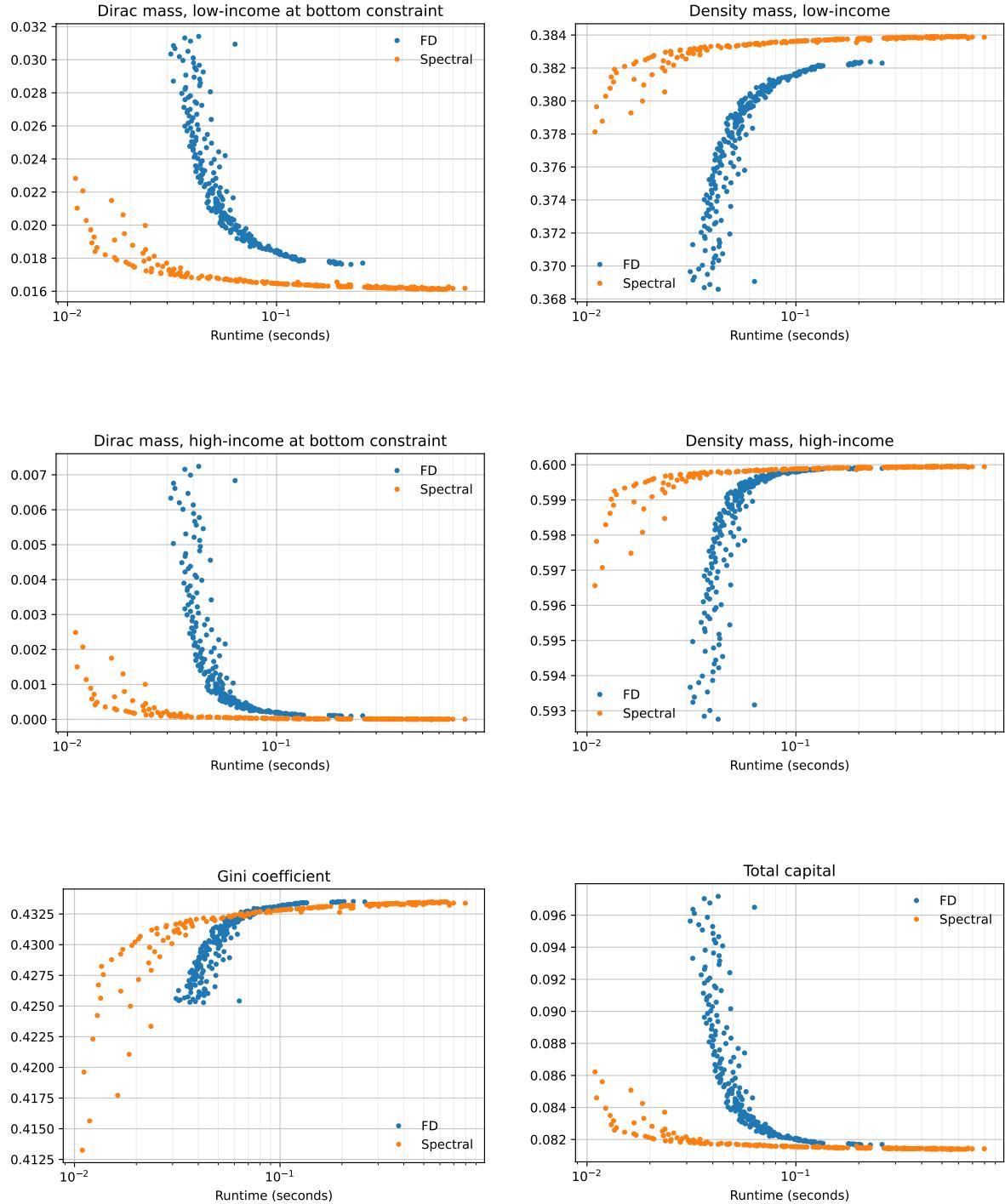


Figure 34: Moments by Execution Times



8.3. Errors for a given runtime

To compare both methods along the truly relevant metric, errors for a given execution time, we recompute the plots from Subsection 8.1 while replacing the abscissa with each iteration’s execution time. As this is a roughly monotonic transform, graphs while look somewhat analogous save for the noise coming from imperfectly measured execution time. To keep them vaguely legible, we thus only show results for the most promising spectral approach, namely solving the HJB via collocation and the KFE via a mixed scheme. Results for the other methods are broadly similar, i.e. slightly slower methods do slightly worse and far slower methods do much worse.

Figure 34 shows that, in terms of execution time, distributional moments converge much faster in the spectral approach than with finite differences. This is especially true for the global moments like total capital and the Gini coefficient. As can be seen in Figure G.3, this also holds for quantiles of the distribution. For a given runtime, the mixed approach is thus much more precise than finite differences.

In the end, the results of this benchmark in the two-income model are highly encouraging. For a given number of nodes, spectral methods are more precise, which is key to reduce the number of required nodes in high-dimensional problems. Even though they require more time for a given number of nodes, they display higher precision for a given runtime, which is key when simple models have to be solved many times over, e.g. when nested in an interest rate iteration for models with a production block, when computing transitions etc.

9. Conclusion

Pseudospectral methods are thus a hopefully useful, complementary approach to solve heterogeneous-agent models in continuous time.

They can be used to efficiently solve Hamilton-Jacobi-Bellman equations in credit-constrained savings problems under complex income processes and with multiple dimensions. In simple problems, they can be combined with finite volumes into hybrid schemes that solve the Kolmogorov Forward Equation for the stationary distribution quickly and precisely. In more complex problems, the smoothness of their resulting savings function can be leveraged to interpolate it to a very fine mesh, which can be easily assembled into a finite volume discretization of the KFE to yield a very precise estimation of the stationary distribution, while taking less time than a coarser pure finite differences approach.

Although we have focussed on very simple, almost an-economic models without markets, production, capital, financial sectors etc., a heterogeneous-household block could be inserted into a broader numerical procedure and solved pseudospectrally within it. As we have stressed and tried to illustrate, the added value of spectral methods lies in their low node requirement for smooth dimensions, which make them particularly suited to be inserted into large, high-dimensional macroeconomic models.

Much like adaptive sparse grids, as in Schaab and Zhang (2021), have extended the realm of applicability of finite differences methods for heterogeneous-agent models to higher-dimensional problems, Smolyak-type sparse methods could perhaps be adapted to a spectral or pseudospectral approach to push the dimensional frontier even further. Promising work along those lines in the field of optimal control, as in Kalise and Kunisch (2018) and Dolgov et al. (2021), could be a good starting point for such future endeavours.

Besides considerable opportunities for complication, a more fundamental quality of the pseudospectral approach is its simplicity: polynomial differentiation matrices are easy to understand intuitively and require minimal effort to be assembled, so that a heterogeneous-agent model can be solved quickly with very little thought and even less code. On the other hand, upwind finite differentiation can quickly become hard to follow, both on paper and in code.

Pseudospectral methods could thus be used by researchers as a simple tool to quickly simulate a

new model idea with very little coding effort, to easily sanity-check simulation results obtained via other methods and, perhaps, even be used to teach the power and beauty of continuous-time methods for heterogeneous-agent models.

References

- Achdou, Y. (2013). Finite difference methods for mean field games. In *Hamilton-Jacobi equations: approximations, numerical analysis and applications*, pp. 1–47. Springer.
- Achdou, Y. and I. Capuzzo-Dolcetta (2010). Mean field games: numerical methods. *SIAM Journal on Numerical Analysis* 48(3), 1136–1162.
- Achdou, Y., J. Han, J.-M. Lasry, P.-L. Lions, and B. Moll (2021). Income and wealth distribution in macroeconomics: A continuous-time approach.
- Achdou, Y. and Z. Kobeissi (2020). Mean field games of controls: Finite difference approximations. *arXiv preprint arXiv:2003.03968*.
- Ahn, S. (2019). Computing the distribution: Adaptive finite volume methods for economic models with heterogeneous agents. Technical Report 10/2019, Norges Bank.
- Ahn, S., G. Kaplan, B. Moll, T. Winberry, and C. Wolf (2018). When inequality matters for macro and macro matters for inequality. *NBER Macroeconomics Annual*.
- Aiyagari, S. R. (1994). Uninsured idiosyncratic risk and aggregate saving. *The Quarterly Journal of Economics* 109(3), 659–684.
- Algan, Y., O. Allais, and W. J. Den Haan (2008). Solving heterogeneous-agent models with parameterized cross-sectional distributions. *Journal of Economic Dynamics and Control* 32(3), 875–908.
- Almulla, N., R. Ferreira, and D. Gomes (2017). Two numerical approaches to stationary mean-field games. *Dynamic Games and Applications* 7(4), 657–682.
- Aruoba, S. B., J. Fernandez-Villaverde, and J. F. Rubio-Ramirez (2006). Comparing solution methods for dynamic equilibrium economies. *Journal of Economic dynamics and Control* 30(12), 2477–2508.
- Auclert, A., B. Bardóczy, M. Rognlie, and L. Straub (2021). Using the sequence-space jacobian to solve and estimate heterogeneous-agent models. *Econometrica* 89(5), 2375–2408.
- Barles, G. and E. Jakobsen (2007). Error bounds for monotone approximation schemes for parabolic hamilton-jacobi-bellman equations. *Mathematics of Computation* 76(260), 1861–1893.
- Barles, G. and E. R. Jakobsen (2002). On the convergence rate of approximation schemes for hamilton-jacobi-bellman equations. *ESAIM: Mathematical Modelling and Numerical Analysis* 36(1), 33–54.
- Barles, G. and P. E. Souganidis (1991). Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic analysis* 4(3), 271–283.
- Berrut, J.-P. and L. N. Trefethen (2004). Barycentric lagrange interpolation. *SIAM review* 46(3), 501–517.
- Bewley, T. (1986). Stationary monetary equilibrium with a continuum of independently fluctuating consumers. *Contributions to mathematical economics in honor of Gérard Debreu* 79.
- Bloom, N., F. Guvenen, L. Pistaferri, J. Sabelhaus, S. Salgado, and J. Song (2017). The great micro moderation.
- Bokanowski, O., J. Garcke, M. Griebel, and I. Klompmaker (2013). An adaptive sparse grid semi-lagrangian scheme for first order hamilton-jacobi bellman equations. *Journal of Scientific Computing* 55(3), 575–605.

- Boyd, J. P. (2001). *Chebyshev and Fourier spectral methods*. Courier Corporation.
- Boyd, J. P. (2010). The legendre–burgers equation: When artificial dissipation fails. *Applied Mathematics and Computation* 217(5), 1949–1964.
- Brumm, J. and M. Grill (2014). Computing equilibria in dynamic models with occasionally binding constraints. *Journal of Economic Dynamics and Control* 38, 142–160.
- Brumm, J., C. Krause, A. Schaab, and S. Scheidegger (2021). Sparse grids for dynamic economic models. [Available at SSRN 3979412](#).
- Brumm, J. and S. Scheidegger (2017). Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica* 85(5), 1575–1612.
- Caldara, D., J. Fernandez-Villaverde, J. F. Rubio-Ramirez, and W. Yao (2012). Computing dsge models with recursive preferences and stochastic volatility. *Review of Economic Dynamics* 15(2), 188–206.
- Carlini, E. and F. J. Silva (2014). A fully discrete semi-lagrangian scheme for a first order mean field game problem. *SIAM Journal on Numerical Analysis* 52(1), 45–67.
- Carmona, R. and M. Laurière (2021). Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games i: the ergodic case. *SIAM Journal on Numerical Analysis* 59(3), 1455–1485.
- Chassagneux, J.-F., D. Crisan, and F. Delarue (2019). Numerical method for fbsdes of mckean–vlasov type. *The Annals of Applied Probability* 29(3), 1640–1684.
- Chen, X., H. Mai, Z. Zhang, and F. Gu (2021). A novel adaptive pseudospectral method for the optimal control problem of automatic car parking. *Asian Journal of Control*.
- Cherrier, B. (2021). The many origins of heterogeneous agent models in macroeconomics. Technical report, CREST.
- Crandall, M. G. and P.-L. Lions (1983). Viscosity solutions of hamilton–jacobi equations. *Transactions of the American mathematical society* 277(1), 1–42.
- Dastyar, M. E., A. Malek, and S. Yousefi (2022). Solving two-dimensional bioheat optimal control problem in solid and vessel domain by pseudospectral discretization. *Transactions of the Institute of Measurement and Control*, 01423312221083770.
- Den Haan, W. J. (1997). Solving dynamic models with aggregate shocks and heterogeneous agents. *Macroeconomic dynamics* 1(2), 355–386.
- Den Haan, W. J. (2010). Assessing the accuracy of the aggregate law of motion in models with heterogeneous agents. *Journal of Economic Dynamics and Control* 34(1), 79–99.
- Dolgov, S., D. Kalise, and K. K. Kunisch (2021). Tensor decomposition methods for high-dimensional hamilton–jacobi–bellman equations. *SIAM Journal on Scientific Computing* 43(3), A1625–A1650.
- Donovan, S. A., M. Labonte, and J. Dalaker (2021). The us income distribution: Trends and issues. CRS Report, US Congressional Research Service, Washington, DC.
- Duarte, V. (2018). Machine learning for continuous-time economics. [Available at SSRN 3012602](#).

- Elnagar, G., M. A. Kazemi, and M. Razzaghi (1995). The pseudospectral legendre method for discretizing optimal control problems. *IEEE transactions on Automatic Control* 40(10), 1793–1796.
- Eymard, R., T. Gallouët, and R. Herbin (2000). Finite volume methods. *Handbook of numerical analysis* 7, 713–1018.
- Fernández-Villaverde, J., S. Hurtado, and G. Nuno (2019). Financial frictions and the wealth distribution. Technical report, National Bureau of Economic Research.
- Fernández-Villaverde, J. and O. Levintal (2018). Solution methods for models with rare disasters. *Quantitative Economics* 9(2), 903–944.
- Gabaix, X. (2009). Power laws in economics and finance. *Annu. Rev. Econ.* 1(1), 255–294.
- Gabaix, X., J.-M. Lasry, P.-L. Lions, and B. Moll (2016). The dynamics of inequality. *Econometrica* 84(6), 2071–2111.
- Gaspar, J. and K. L. Judd (1997). Solving large-scale rational-expectations models. *Macroeconomic Dynamics* 1(1), 45–75.
- Gordon, G. (2020). Computing dynamic heterogeneous-agent economies: Tracking the distribution. *Economic Quarterly* (2Q), 61–95.
- Greenberg, A. (2003). *Chebyshev spectral method for singular moving boundary problems with application to finance*. Ph. D. thesis, California Institute of Technology.
- Guo, X., A. Hu, R. Xu, and J. Zhang (2019). Learning mean-field games. *Advances in Neural Information Processing Systems* 32.
- Guvenen, F., G. Kaplan, J. Song, and J. Weidner (2021). Lifetime earnings in the united states over six decades. *University of Chicago, Becker Friedman Institute for Economics Working Paper* (2021-60).
- Han, J., A. Jentzen, et al. (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in mathematics and statistics* 5(4), 349–380.
- Hemker, P. W. (2000). Application of an adaptive sparse-grid technique to a model singular perturbation problem. *Computing* 65(4), 357–378.
- Hopenhayn, H. A. (1992). Entry, exit, and firm dynamics in long run equilibrium. *Econometrica: Journal of the Econometric Society*, 1127–1150.
- Huggett, M. (1993). The risk-free rate in heterogeneous-agent incomplete-insurance economies. *Journal of Economic Dynamics and Control* 17(5-6), 953–969.
- Judd, K. L. (1992). Projection methods for solving aggregate growth models. *Journal of Economic theory* 58(2), 410–452.
- Judd, K. L. (1998). *Numerical methods in economics*. MIT press.
- Judd, K. L., L. Maliar, S. Maliar, and R. Valero (2014). Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics and Control* 44, 92–123.

- Kalise, D. and K. Kunisch (2018). Polynomial approximation of high-dimensional hamilton–jacobi–bellman equations and applications to feedback control of semilinear parabolic pdes. *SIAM Journal on Scientific Computing* 40(2), A629–A652.
- Kang, W. and L. C. Wilcox (2017). Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and hjb equations. *Computational Optimization and Applications* 68(2), 289–315.
- Kopriva, D. A. (2009). *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*. Springer Science & Business Media.
- Krueger, D. and F. Kubler (2004). Computing equilibrium in olg models with stochastic production. *Journal of Economic Dynamics and Control* 28(7), 1411–1436.
- Krusell, P. and A. A. Smith, Jr (1998). Income and wealth heterogeneity in the macroeconomy. *Journal of political Economy* 106(5), 867–896.
- Lasry, J.-M. and P.-L. Lions (2007). Mean field games. *Japanese journal of mathematics* 2(1), 229–260.
- Lauriere, M. (2021). Numerical methods for mean field games and mean field type control. *arXiv preprint arXiv:2106.06231*.
- Lauriere, M. and O. Pironneau (2016). Dynamic programming for mean-field type control. *Journal of Optimization Theory and Applications* 169(3), 902–924.
- Lazarov, R., I. D. Mishev, and P. S. Vassilevski (1996). Finite volume methods for convection-diffusion problems. *SIAM Journal on Numerical Analysis* 33(1), 31–55.
- Lepsky, O. (2000). Spectral viscosity approximations to hamilton–jacobi solutions. *SIAM journal on numerical analysis* 38(5), 1439–1453.
- Mai, H. (2022). Convergence for the optimal control problems using collocation at legendre-gauss points. *Transactions of the Institute of Measurement and Control* 44(6), 1263–1274.
- Malin, B. A., D. Krueger, and F. Kubler (2011). Solving the multi-country real business cycle model using a smolyak-collocation method. *Journal of Economic Dynamics and Control* 35(2), 229–239.
- Mashadi, B. and M. Majidi (2014). Two-phase optimal path planning of autonomous ground vehicles using pseudo-spectral method. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics* 228(4), 426–437.
- Miranda, M. J. and P. L. Fackler (2004). *Applied computational economics and finance*. MIT Press.
- Moll, B. (2018). Hopenhayn model in continuous time. Lecture notes, London School of Economics.
- Nakamura-Zimmerer, T., Q. Gong, and W. Kang (2021). Adaptive deep learning for high-dimensional hamilton–jacobi–bellman equations. *SIAM Journal on Scientific Computing* 43(2), A1221–A1247.
- Nikooeinejad, Z., A. Delavarkhalafi, and M. Heydari (2016). A numerical solution of open-loop nash equilibrium in nonlinear differential games based on chebyshev pseudospectral method. *Journal of Computational and Applied Mathematics* 300, 369–384.
- Nuño, G. and B. Moll (2018). Social optima in economies with heterogeneous agents. *Review of Economic Dynamics* 28, 150–180.

- Nuño, G. and C. Thomas (2019). Optimal monetary policy with heterogeneous agents. Documentos de trabajo/Banco de España, 1624.
- Pugh, D. R. (2016). Pycollocation.
- Reiter, M. (2004). Recursive computation of heterogeneous agent models. Technical report, Universitat Pompeu Fabra.
- Ross, I. M. and M. Karpenko (2012). A review of pseudospectral optimal control: From theory to flight. Annual Reviews in Control 36(2), 182–197.
- Ruthotto, L., S. J. Osher, W. Li, L. Nurbekyan, and S. W. Fung (2020). A machine learning framework for solving high-dimensional mean field game and mean field control problems. Proceedings of the National Academy of Sciences 117(17), 9183–9193.
- Ruttscheidt, S. (2018). Adaptive Sparse Grids for Solving Continuous Time Heterogeneous Agent Models. Ph. D. thesis, Master's Thesis in Mathematics, University of Bonn.
- Saloff-Coste, L. (1996). Lectures on finite markov chains. ecole d'été de probabilités de saint-flour xxvi. Lecture Notes in Math, 1665301–413.
- Schaab, A. and A. T. Zhang (2021). Dynamic programming in continuous time with adaptive sparse grids.
- Shaker Akhtekhane, S. (2017). Firm entry and exit in continuous time. Technical report, Working paper, Ohio State University.
- Sirignano, J. and K. Spiliopoulos (2018). Dgm: A deep learning algorithm for solving partial differential equations. Journal of computational physics 375, 1339–1364.
- Stokey, N. L. (2008). The economics of inaction. In The Economics of Inaction. Princeton University Press.
- Tadmor, E. (1990). Shock capturing by the spectral viscosity method. Computer Methods in Applied Mechanics and Engineering 80(1-3), 197–208.
- Tarkiainen, R. and M. Tuomala (1999). Optimal nonlinear income taxation with a two-dimensional population; a computational approach. Computational Economics 13(1), 1–16.
- Trefethen, L. N. (2000). Spectral methods in matlab.
- Vlassenbroeck, J. and R. Van Dooren (1988). A chebyshev technique for solving nonlinear optimal control problems. IEEE transactions on automatic control 33(4), 333–340.
- Winberry, T. (2018). A method for solving and estimating heterogeneous agent macro models. Quantitative Economics 9(3), 1123–1151.

Appendices

A. Codes

All codes written for this Master Thesis have been posted to GitHub and can be found in the repository `spectral-heterogeneous-agents`. More precisely, it includes:

1. `backend`: various helper functions, notably for manipulating and assembling differentiation matrices, that are used in all other codes of this project
2. `two_state`: code for the two-income Hugget model which is solved in Section 4 and used for the benchmark of Section 8
3. `two_state_weakly_binding`: code for the two-income model with a weakly binding credit constraint, in which the pseudospectral solution to the KF equation is not too terrible
4. `diffusion`: code for the diffusive income Hugget model solved in Section 5
5. `life_cycle`: code for the life-cycle model solved in Section 6
6. `slow_transitions`: code for the Power law model and transition speed analysis from Section 7

All code is (hopefully) straightforward and relies only on `Numpy` and `SciPy`'s `sparse`. The only vaguely notable innovation is the use of `Pandas`' label-based indexing to handle Kronecker-broadcasting and array-indexing: although the matrices themselves are standard arrays, it is often useful to define a `MultiIndex` alongside it that contains the correctly-tiled combinations of a , z and t (depending on the model). It can then be used to return integer indexes corresponding to label-based Pandas indexes or slices, which is especially useful to avoid headaches while imposing boundary conditions. This is exclusively a convenience tool, which has no positive effect on performance although it only adds extremely limited overhead, especially since all indexing operations are fast and can be precomputed.

B. Calibrations and numerical parameters

Table 1: Calibrations and numerical parameters across applications

	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	\$1,00,000
λ_1, λ_2	1.5, 1.0			
σ^2		0.05	0.0974	0.0974
θ		1 ¹⁶	0.0682	0.0682
Assets				
$\frac{a}{\bar{a}}$	-0.02	-0.1	\$0	-\$50,000
	1.5 ¹⁷	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:	<i>huggett_partialeq.m</i>	<i>huggett_diffusion_partialeq.m</i>	<i>See Section 6.1</i>	<i>See Section 7</i>

¹⁶The code uses $\theta = -\ln(0.9)$, which leaves $\sim 60\%$ of the stationary mass outside of the income bounds; we thus use the far more reasonable $\theta = 1$.

¹⁷As savings are negative above ~ 0.5 , we have cut the asset space from $\bar{a} = 3$ to $\bar{a} = 1.5$ to reduce precision waste.

C. Additional plots for the two-income Hugget model

Figure C.1: Value Function Consistency Check - One-period Expectation

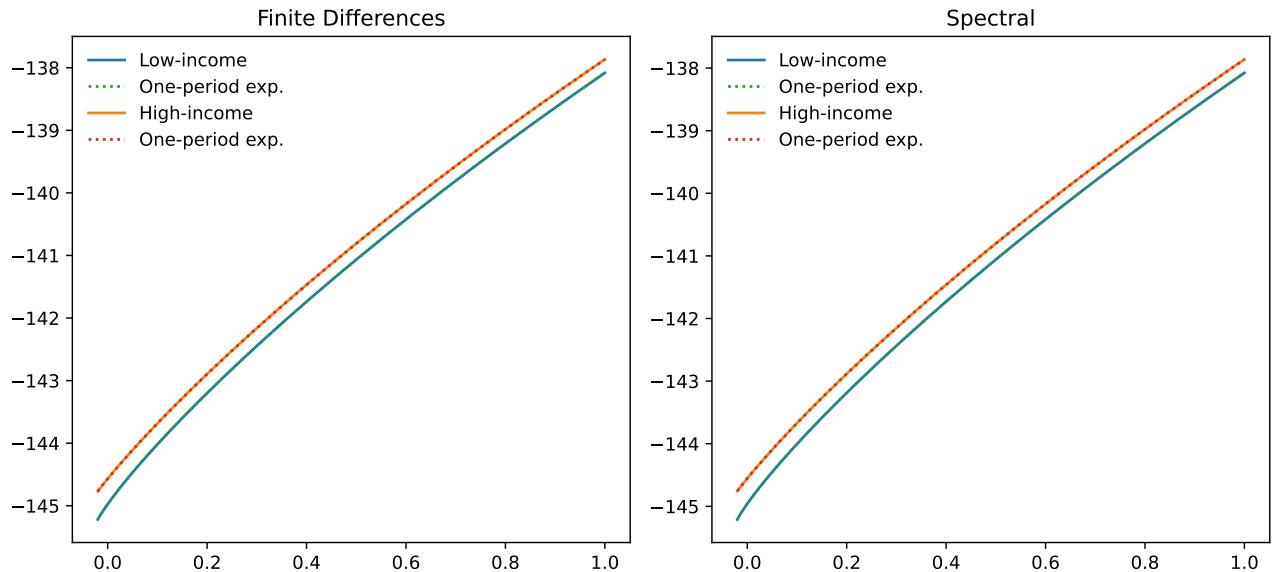


Figure C.2: Value Function Consistency Check - One-period Expectation Error

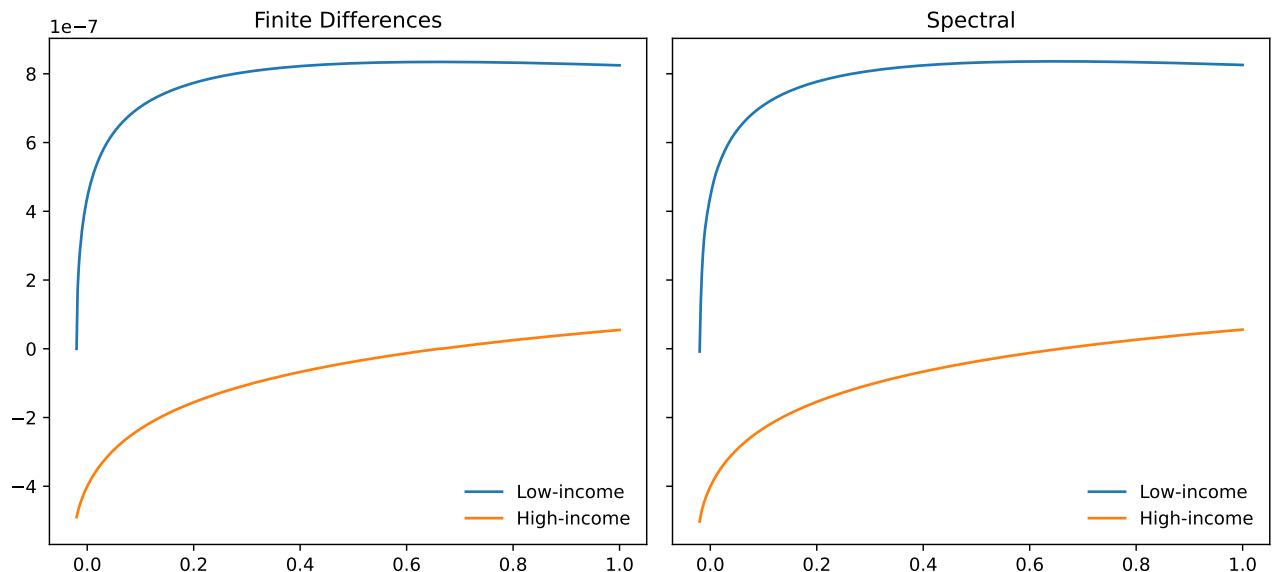
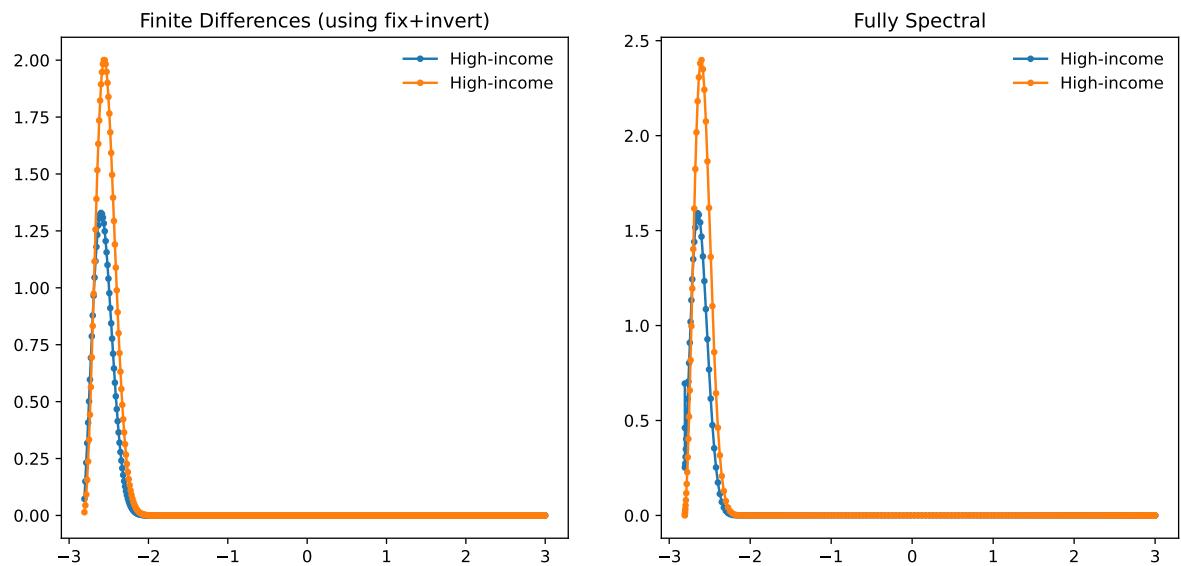


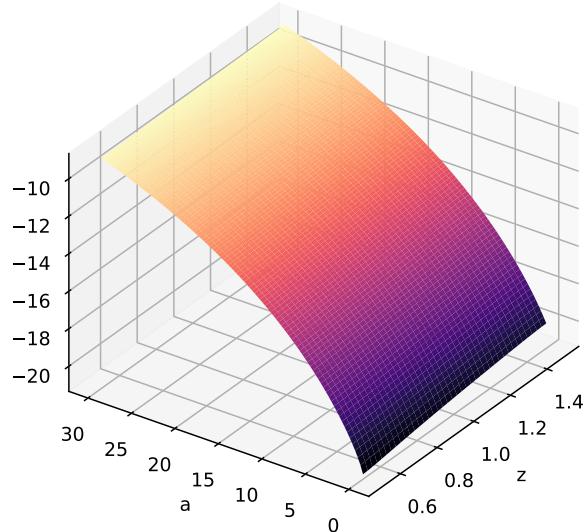
Figure C.3: Stationary Distribution - Fully Spectral Approach under Weakly Binding Credit Constraint



D. Additional plots for the diffusive income Hugget model

Figure D.1: Value Functions

Finite Differences



Spectral (interpolated)

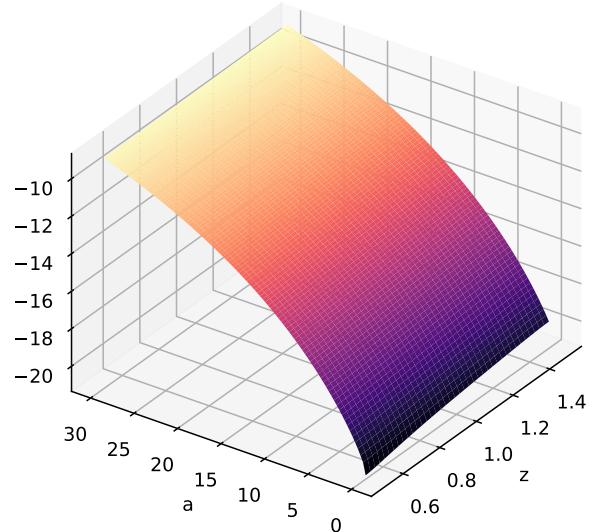
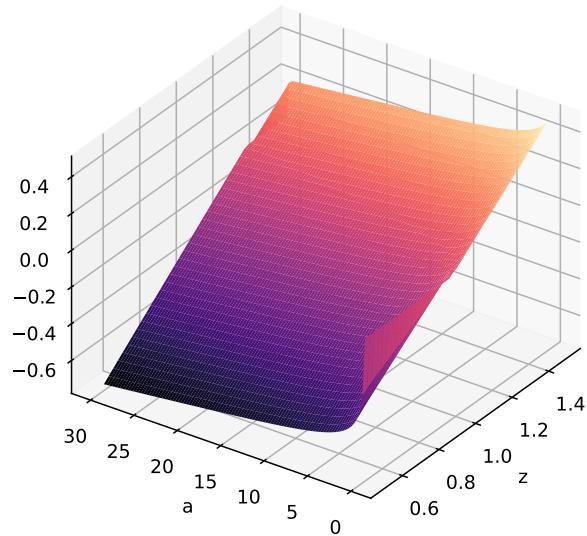


Figure D.2: Savings Functions

Finite Differences



Spectral (interpolated)

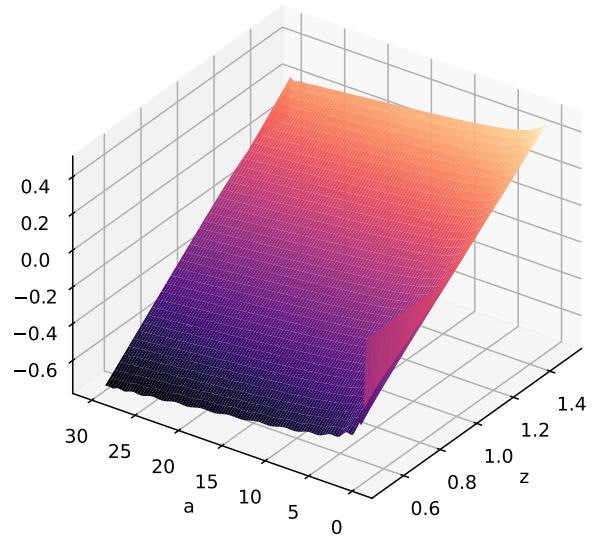
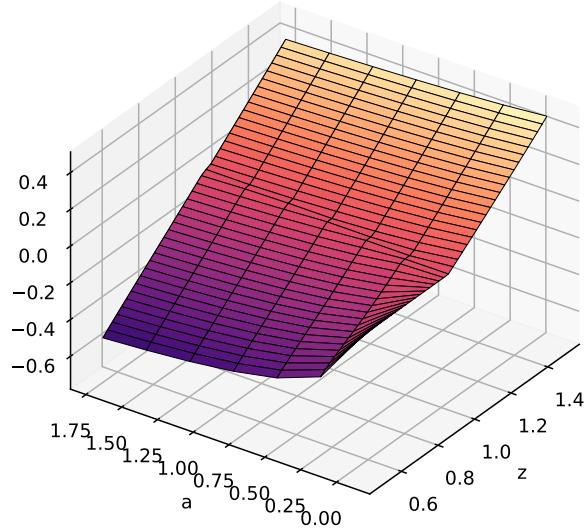


Figure D.3: Savings Functions (Zoom on Borrowing Constraint)

Finite Differences



Spectral (interpolated)

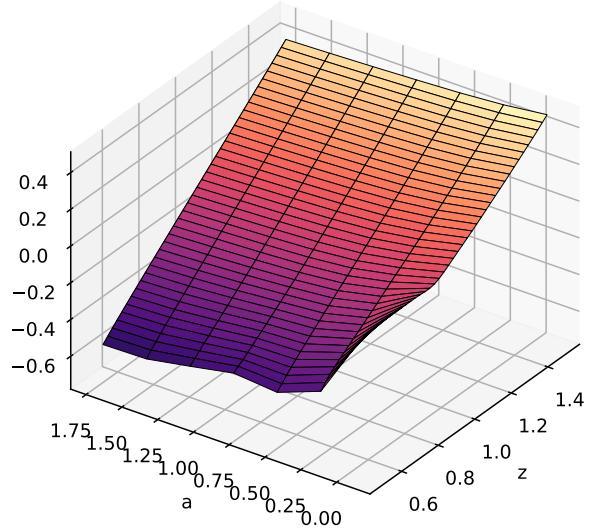
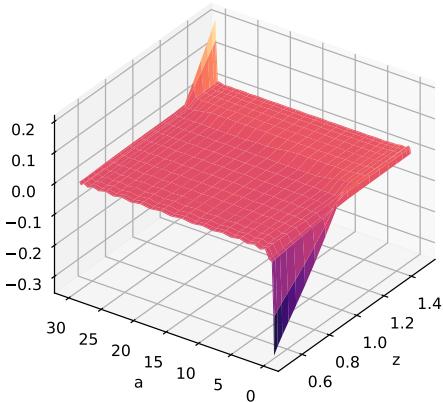


Figure D.4: Difference Between Savings Computed Using Interpolation and Actual Savings

Finite Differences

Abs. mean: 1.65e-02 - Abs. median: 7.45e-03



Spectral

Abs. mean: 2.59e-12 - Abs. median: 1.56e-12

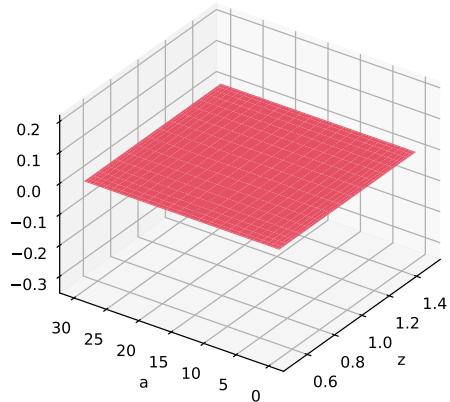


Figure D.5: Cross-section along z of savings at \underline{a}

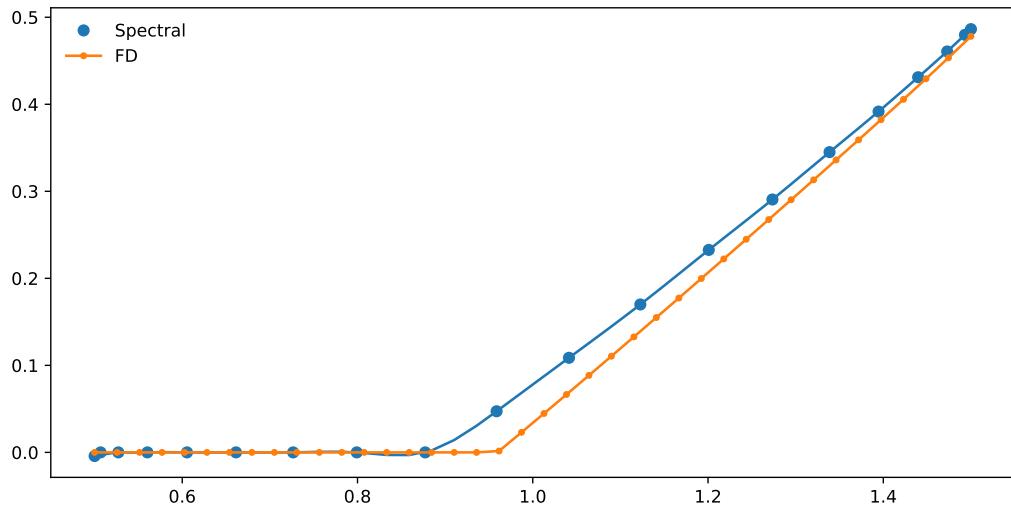


Figure D.6: Cross-section along a of the z-derivative of the value function at z_{min}

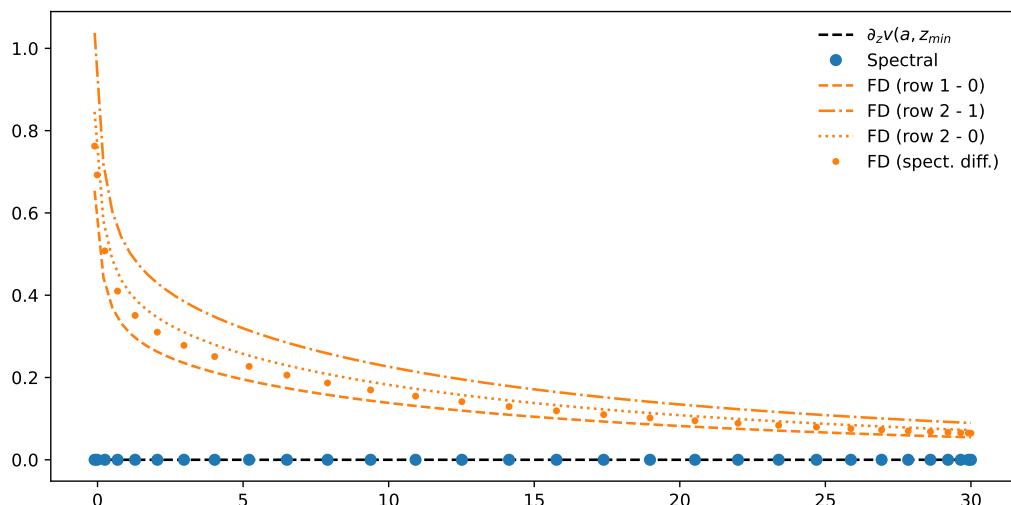


Figure D.7: Integrated value function consistency errors

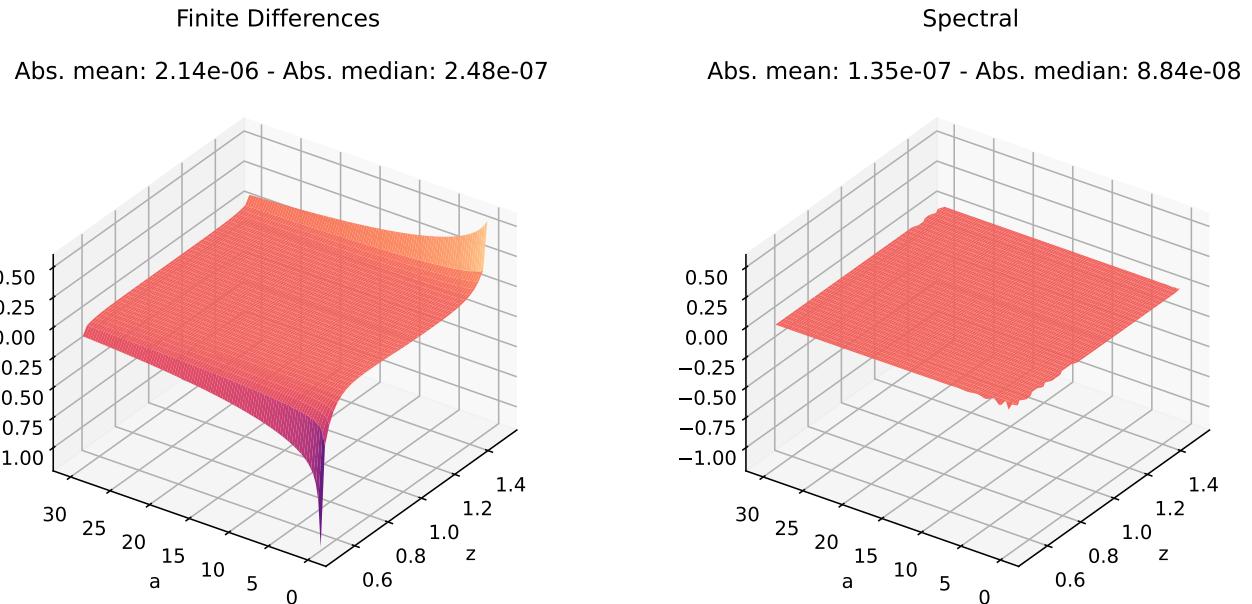
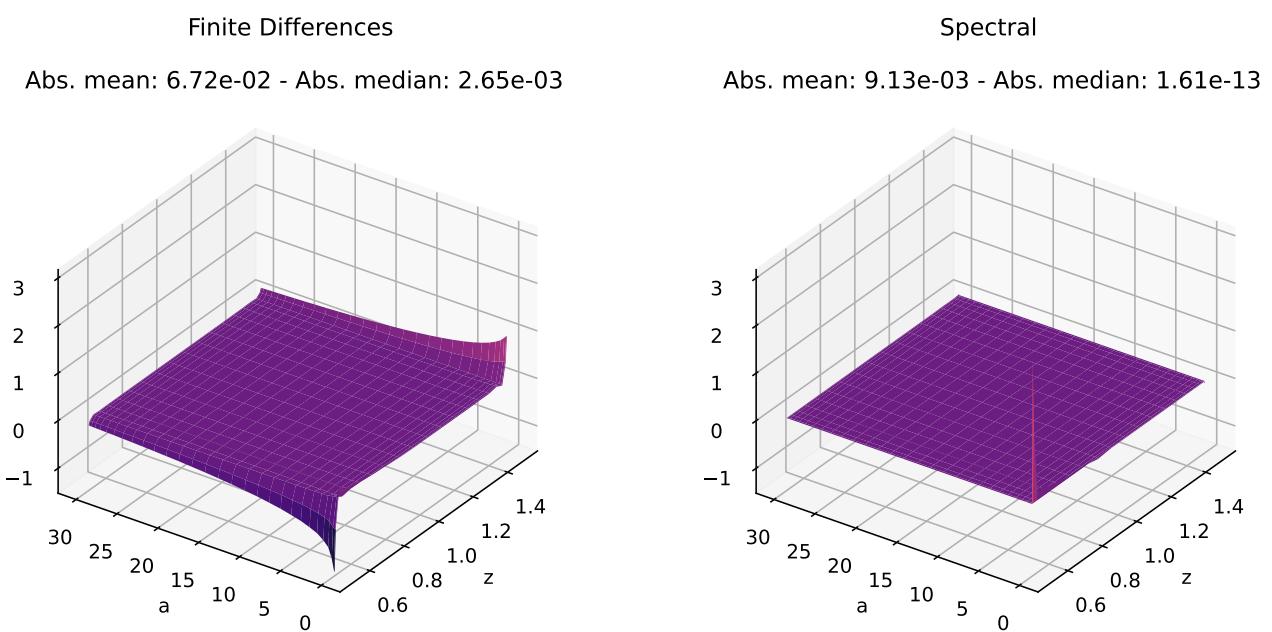


Figure D.8: Errors in Hamiltonian-form HJB Equation



E. Additional plots for the life-cycle model

Figure E.1: Consumption over the Life-Cycle

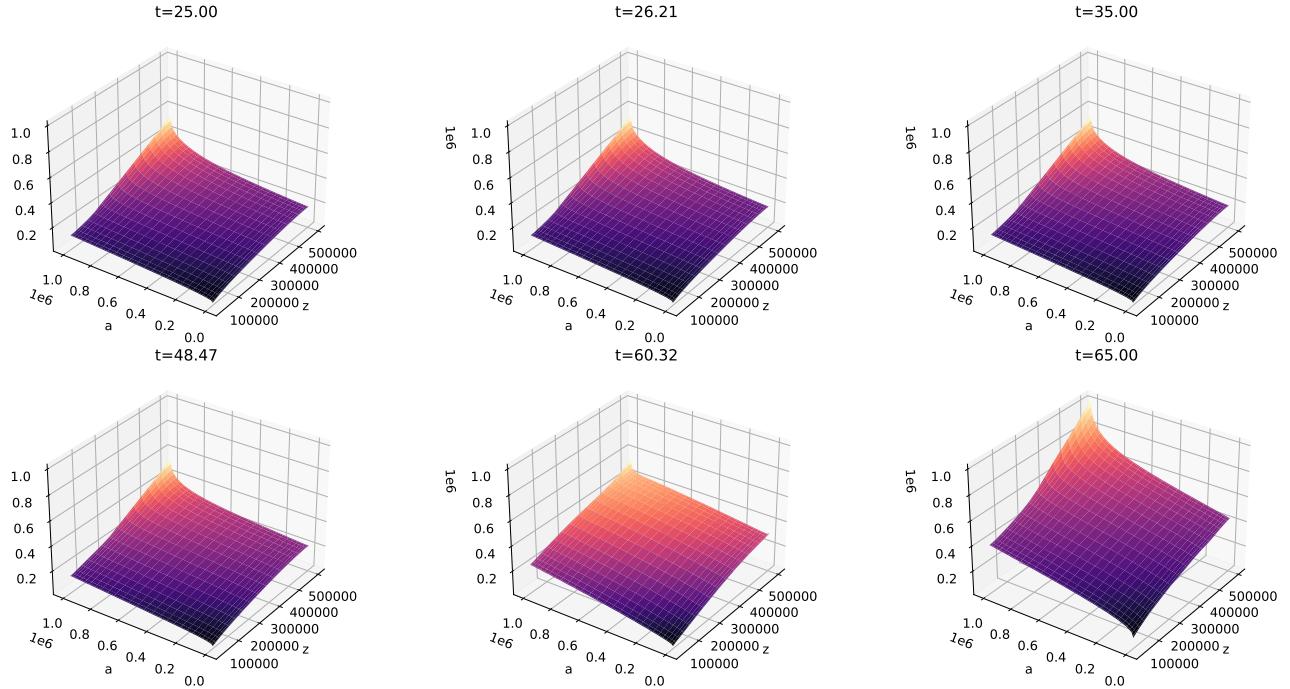


Figure E.2: Interpolated Value over the Life-Cycle

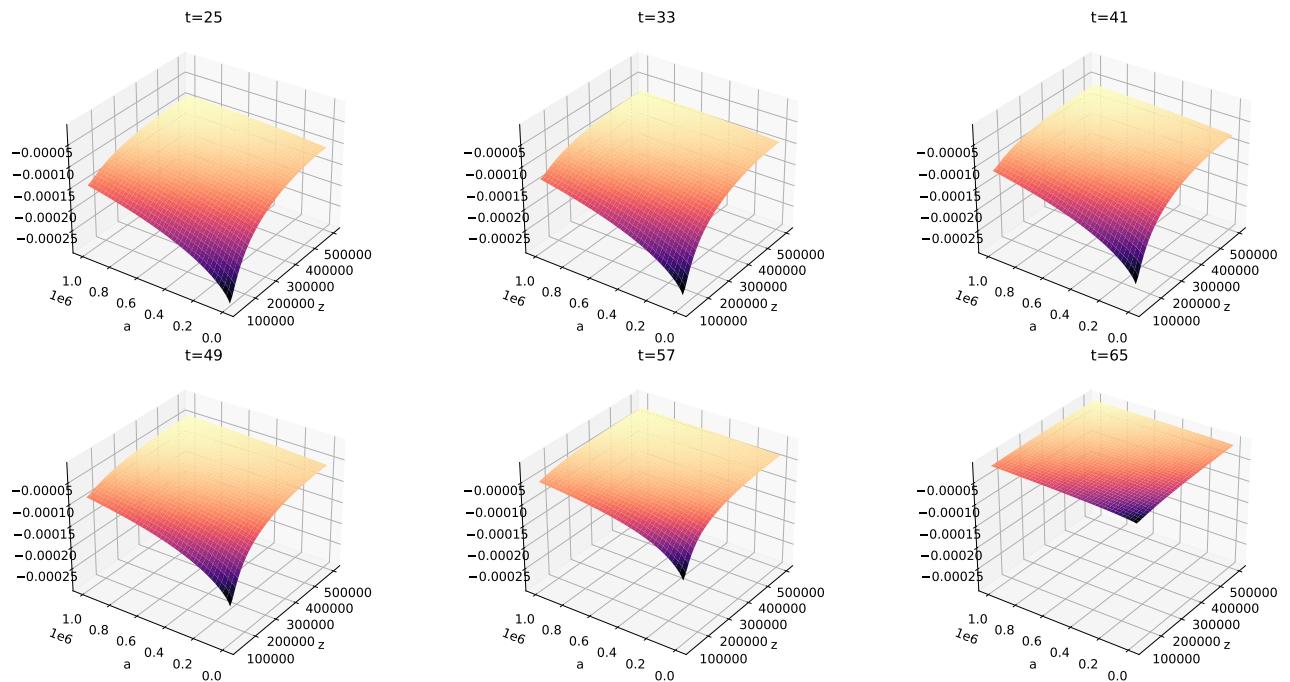


Figure E.3: Interpolated Savings over the Life-Cycle

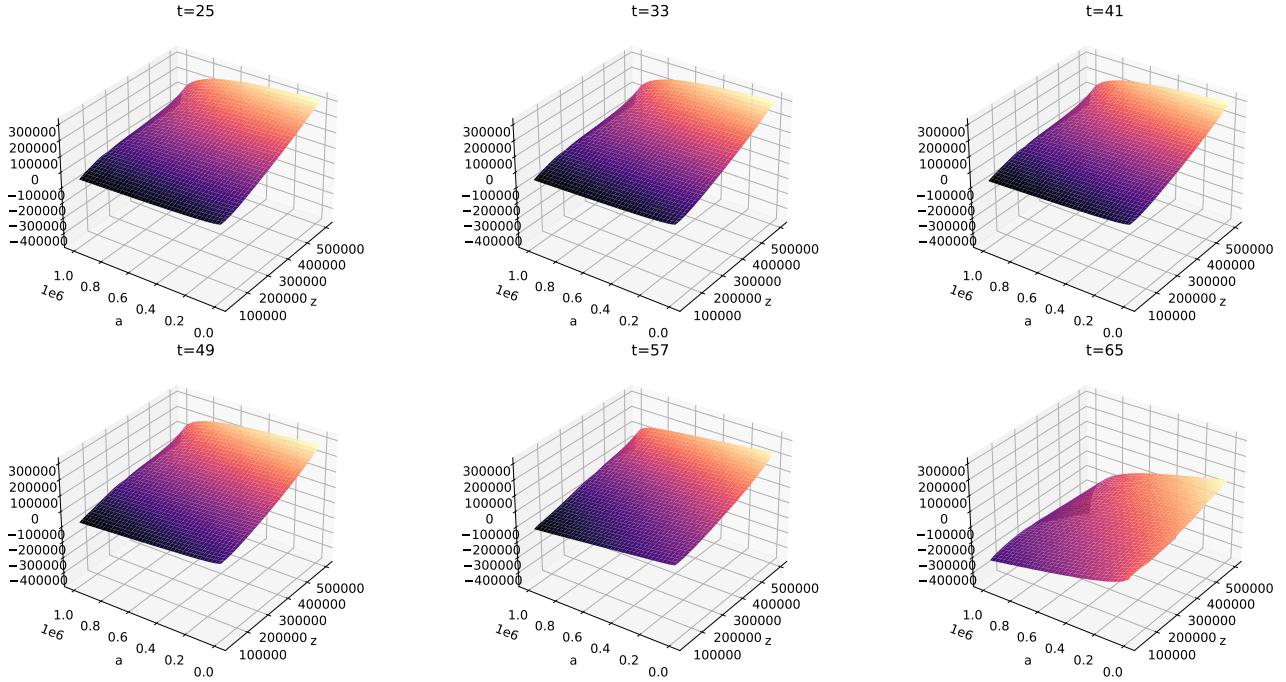
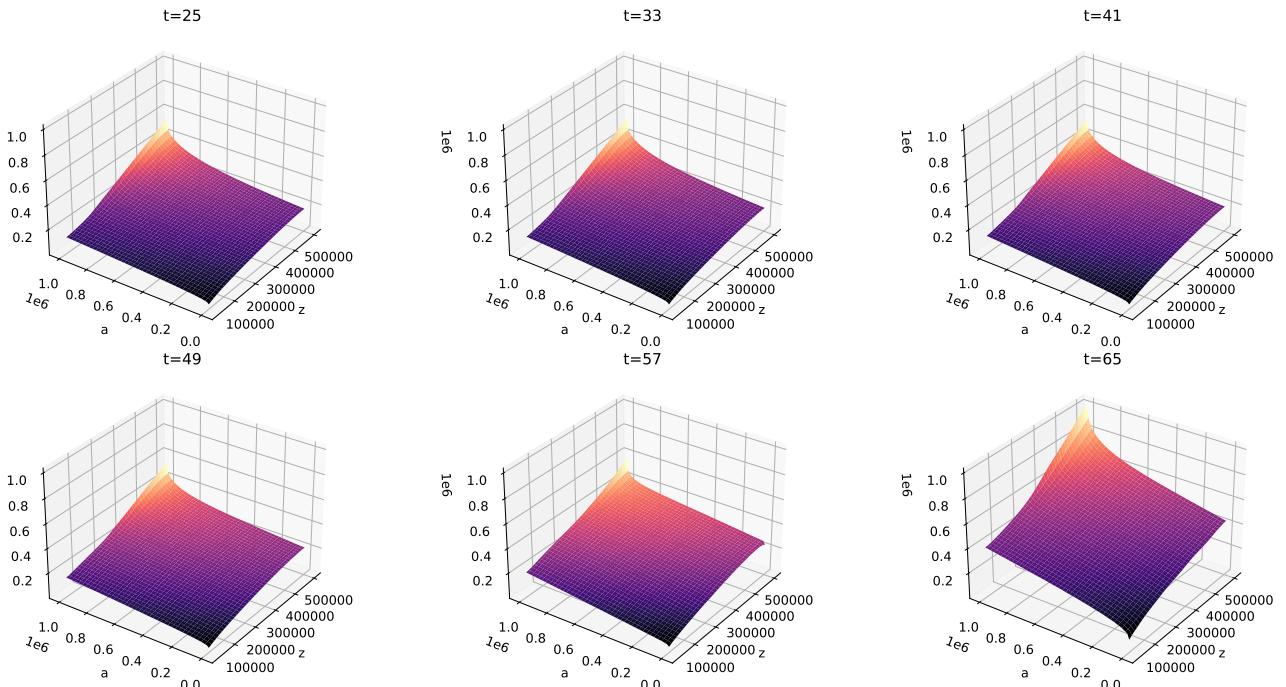


Figure E.4: Interpolated Consumption over the Life-Cycle



F. Additional plots for the "slow transitions" application

Figure F.1: First 20 Generator Eigenvalues in Different Models

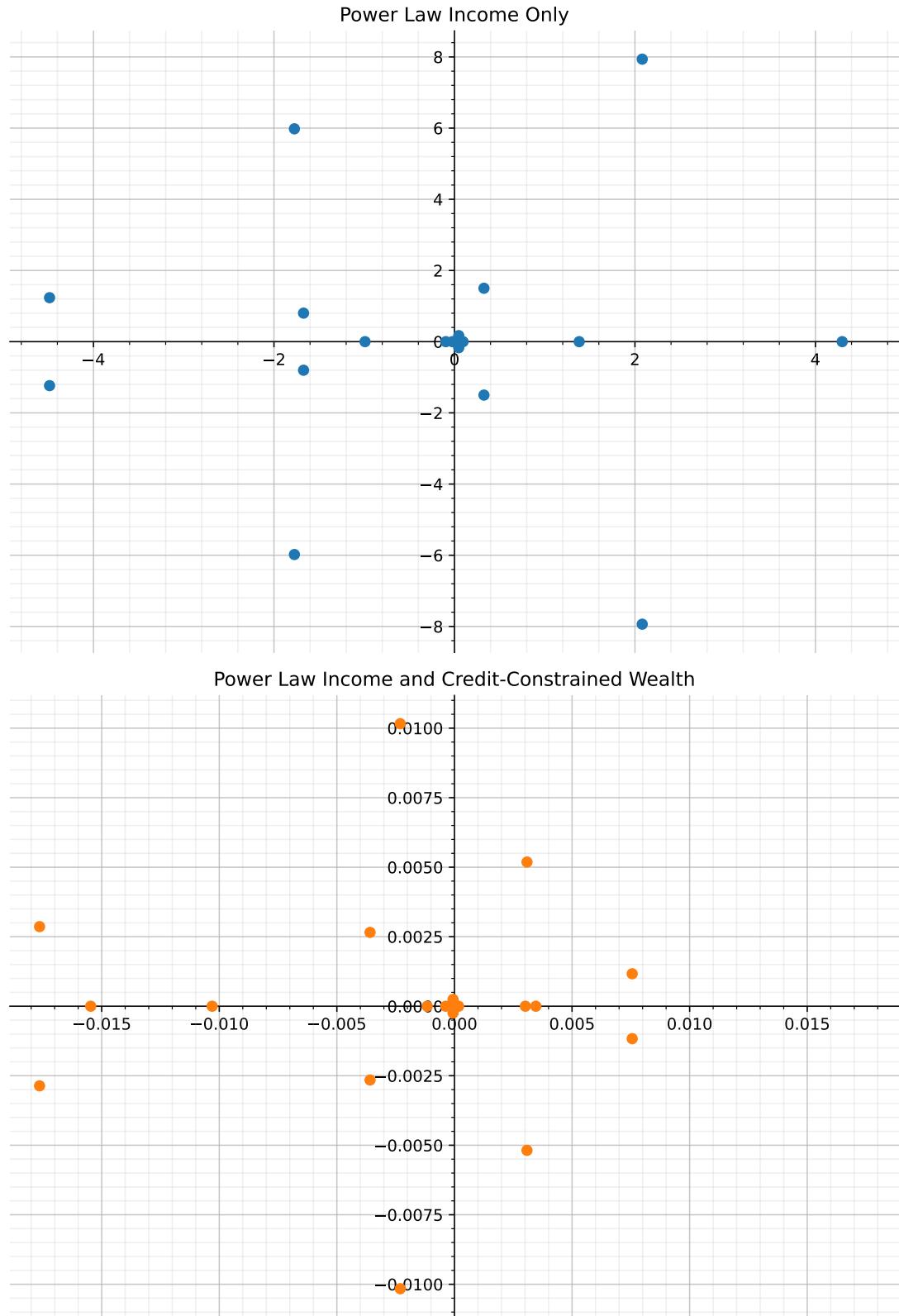
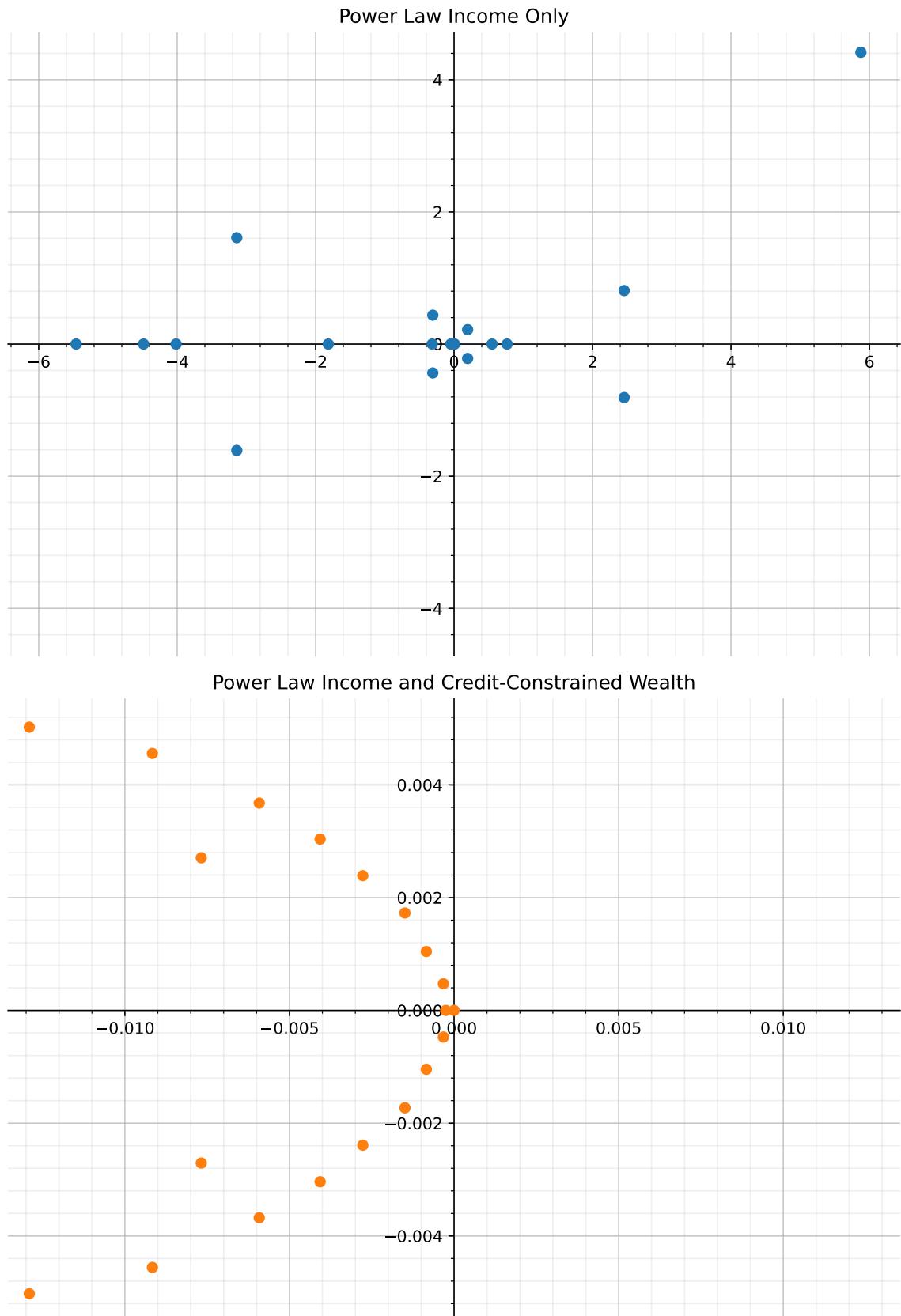


Figure F.2: $E := \frac{1}{2}(A + A^*)$'s First 20 Eigenvalues in Different Models



G. Additional plots for the two-income benchmark

Figure G.1: Errors in the Savings Function by Number of Nodes

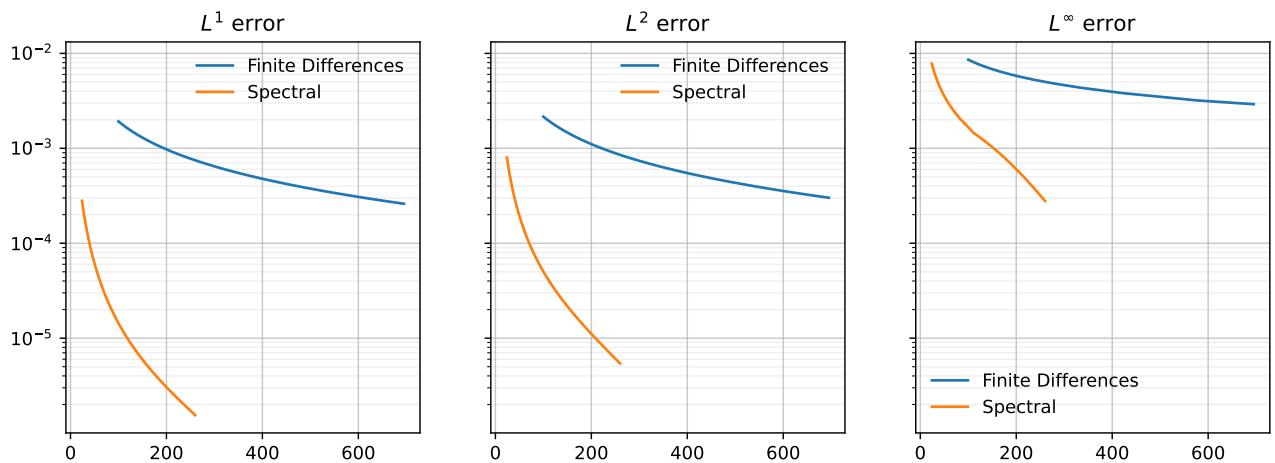


Figure G.2: Quantiles by Number of Nodes

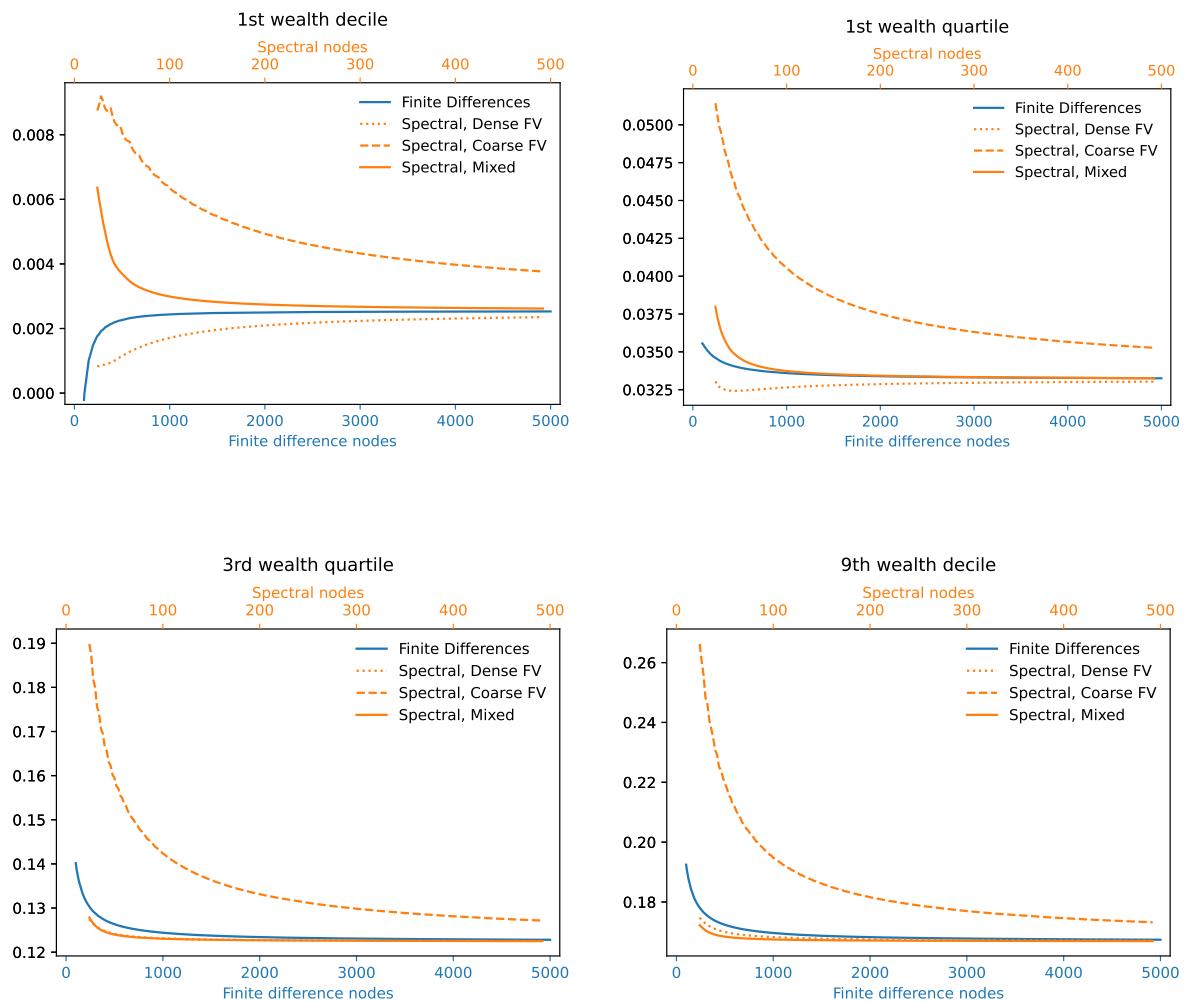


Figure G.3: Quantiles by Execution Times

