Solving Heterogeneous agent models in Continuous Time with Adaptive Sparse Grids

Du, Zaichuan

University of Munich, Faculty of Economics

4 July 2024

Solving Heterogeneous agent Models in Continuous Time with Adaptive Sparse Grids

Zaichuan Du*

July 4, 2024

Abstract

This paper proposes a new approach to numerically solving a wide class of heterogeneous agent models in continuous time using adaptive sparse grids. I combine the sparse finite difference method with the sparse finite volume method to solve the Hamilton-Jacobian-Bellman equation and Kolmogorov Forward equation, respectively. My algorithm automatically adapts grids and adds local resolutions in regions of the state space where both the value function and the distribution approximation errors remains large. I demonstrate the power of my approach in applications feature high-dimensional state spaces, occasionally binding constraints, lifecycle and overlapping generations.

JEL codes: C61, C63, E21

Keywords: heterogeneous agent, mean field game, continuous time, adaptive sparse grids, occasionally binding constraints, overlapping generation, lifecycle

*University of Munich. Email: duzaichuan@hotmail.com

I am grateful to Andreas Schaab for his valuable advice and comments.

1 Introduction

Heterogeneous agent models, namely Hamilton-Jacobian-Bellman (HJB) and Kolmogorov Forward (KF) equation system, are widely used in economic applications. In the Mean Field Games (MFG) literature in mathematics, the system of coupled HJB and KF equations is known as the "backward-forward MFG system." Their numerical implementation usually employs uniform grids, which suffer from the curse of dimensionality. Adaptive sparse grids help overcome this challenge by refining the grid to best suit the underlying economic model, placing grid knots where increased resolution is valuable and removing them elsewhere. There are cases in which we want to use an adaptive sparse grid. For example, this is the case if the function does not fulfill the required smoothness conditions or the function is flat in most parts of the domain but has some steep regions in other parts. Most heterogeneous agent applications in economics have such features.

In this paper, I propose a new approach to solving heterogeneous agent models in continuous time using adaptive sparse grids. My starting point is two negative results: Firstly, in continuous time, solving heterogeneous agent models is based on the solution of differential equations, HJB and KF. Existing adaptive sparse grid methods are only able to solve the HJB equation on the sparse grid, but not at the same time the KF equation. This incompatibility is due to different properties between the differential operator of HJB and the integration operator of KF. Using solely finite difference method (Achdou et al., 2021) works perfectly with regular (non-)uniform grids, but fails in the adaptive sparse grid. Secondly, grid adaptation merely based on the value function is sometimes inefficient (Ruttscheidt, 2018). In most cases of economic applications, we are interested in both policy decisions of households and their associated distribution in heterogeneous agent models. They are a coupled system: the HJB equation implicitly depends on the distribution solved in the KF equation through prices, while the latter is formed by policy functions solved in HJB.

To tackle these issues, I structure the grid space into two compatible and interactive systems through their shared hierarchical approximation space. Such that we can solve the HJB by a sparse finite difference method (FDM) and the KF equation by a sparse finite volume method (FVM). FVM helps preserving the mass of distributions in the adaptive sparse grid by design and is thus more suitable for handling the dynamics of distribution. I develop a new general equilibrium iteration algorithm—based on the combination of FDM and FVM—to solve heterogeneous agent problems and further speed up the computation of high dimensional economics models. The power of the algorithm is greater than the sum of its parts (FDM or FVM). The common hierarchical structure of the grid points used by FDM and FVM are central to my approach. Not only does the hierarchization give measures of local approximation error of VFI, but also it allows the projection of policy decisions approximated by FDM to the points used in describing the evolution of the distribution by FVM. The latter then determines the evolution of prices being taken as the input in the HJB equation. This interdependence of the HJB-KF equations promotes better grid refinement for the MFG system: Refining the grid points used by solving the KF equation provides accuracy gain for refining those used by solving the dynamic programming HJB equation, and vice verse.

The paper explores grids and their cell structure, which discretizes the state space of different economic applications. In higher dimensions, uniform grids, featuring regularly spaced nodes, suffer from the curse of dimensionality, as the number of grid points associated with the full tensor product grows exponentially. Sparse grids aim to remove grid points through judging the value of each point in terms of its marginal contribution to the accuracy of approximations. Adaptive sparse grids utilize details of a specific economic problem to locally refine the grid, placing more points where the economic application features non-linearity. For solving the policy and value functions associated with the dynamic optimization problem—the analogue of the Bellman equation in discrete time, Schaab and T.Zhang (2022) develop a sparse FDM to interpolate the value function onto non-uniform adaptive grid points. I extend their algorithm, making it also possible to solve the distribution KF function on the adaptive sparse grid. To achieve this goal, I fit the FVM well into the same hierarchical space of FDM, such that we are able to solve heterogeneous agent models entirely on the adaptive sparse grid.

FVM, as the name of finite volume method suggests, tracks the volume change between histograms instead of the difference between points. It is widely used in the mathematics/engineering literature and is introduced to economics by Ahn (2019). It preserves mass and positivity of integration. FVM can be considered as a continuous time version of (Young, 2010)'s method, being widely used in discrete time heterogeneous agent models. Different from FDM and Young's method, FVM does not rely on interpolations to compute the dynamics from the given (approximate) distribution to the resulting (approximate) distribution consistent with individual behaviors. FVM takes advantage of the grid cell structure and accounts for flows between cells. It is this feature that makes FVM able to preserve mass and positivity in transition dynamics of the distribution on the adaptive sparse grid. Thanks to the cell structure, it is intuitive to consider boundary conditions and endogenous flows of distributions. The overlapping generation (OLG) model is a good example of this. See Section 5.2 for more details.

I demonstrate the power of my approach in the context of typical applications, showing that adaptive sparse grids substantially outperform uniform and even regular sparse grids. Many economic applications feature occasionally binding constraints such as borrowing and collateral constraints. When being constrained, households' behavior often becomes highly sensitive to (non-linear in) changes in fundamentals. Adaptive sparse grid methods excel in placing grid points in the region of the state space where these constraints bind. I illustrate this in Section 4 and Section 5.1. Dynamic general equilibrium models with rich heterogeneity are notoriously challenging to solve, since the high-dimensional cross-sectional distribution becomes part of the state space. In Section 5.1, I globally solve a variant of the Krusell et al. (1998) model with a 14-dimensional representation. Despite the large state space, my algorithm requires less than 4 minutes on a personal computer. Many important economic problems feature finite horizons, e.g. lifecycle and OLG models. Solving these models accurately often requires high grid resolution near the born, retirement and death. I show in Section 5.2 that a lifecycle portfolio choice problem requires substantial localized resolution to accurately characterize households' behavior near these periods. As people in different part of their lifecycle have different wealth levels, the refinement on the wealth dimension differs in their age. The same

accuracy can be achieved using only 14% of the grid points required by the uniform grid for the OLG model.

Related Literature. In a groundbreaking work, Achdou et al. (2021) uses the finite difference methods to solve both HJB and KF on the uniform grid in the continuous time. They also provide a workaround for mass preservation on the non-uniform regular grid (Figure 1(b)), but it is no longer the FDM and the patch does not have a theoretical foundation. Ahn (2017), Ruttscheidt (2018) and Schaab and T.Zhang (2022) propose similar sparse FDM to consistently solve the HJB equation on adaptive grids, but the KF equation still has to be solved on the uniform-spaced grid. Ahn (2019) introduces the FVM to economics from the mathematics/engineering literature to solve the KF equation on the adaptive sparse grid. The applications in his paper, however, restrict to solving the KF equation in the partial equilibrium and do not solve the HJB equation. Pflüger and Franzelin (2018) develop a method to preserve the density function value range by extending the original sparse grid used in HJB with a minimal extension set. The size of the extension set, however, differs among applications and could be almost as large as the corresponding full grid. With the combination of FDM and FVM, my approach not only preserves the original sparsity of FDM grid, but also manages to compute the KF equation on an adaptively sparse grid. While my approach uses two set of sparse grid points for computing the HJB and KF equations, respectively, those grids share a same hierarchical space. Hence, the projection between these two sets of grids is straightforward and of little computational cost, but rather facilitates the refinement of the coupled HJB-KF system. I know of no other papers that have yet employed this useful combination.

The idea of this paper is implicitly related to recent progress of the Master equation in the Mean Field Games literature (Bilal, 2023; Cardaliaguet et al., 2019). It consists of a single Bellman equation that describes the entire movement of a system of interacting agents. This Master Equation includes as state variables households' idiosyncratic states and the distribution of all other households of the economy. Based on this theoretical result, I consider whether it is better to refine the grid in the complete system of both the HJB and KF equations, rather than solely the dynamic programming of the value function. My approach builds on Schaab and T.Zhang (2022)'s sparse FDM to solve the dynamic programming problem of high dimensional models and extends their VFI algorithm to one that allows adaptively refined KF equation rather than regular dense grid. Why can't we simply use sparse FDM also for solving the KF equation? The key reason is that the KF equation describes the movement of the distribution, which is a measurement. A measurement is not simply a function in the sense that measurements have to preserve mass and positivity. Once we start dealing with non-uniform spaced grid (Figure 8(b) and 8(c)), FDM fails in preservation. FVM thus comes to rescue. It has been around for decades in the math/engineer literature on discretization of partial differential equations.

The rest of this paper proceeds as follows. Section 2 introduces hierarchical basis functions and shows how to construct sparse grid for the HJB equation. Section 3 presents the FVM grid and intuition of preserving the mass when approximating the distribution. Section 4 presents a general equilibrium iteration algorithm on adaptive sparse grids with a Huggett economy. Section 5 presents more economic applications to demonstrate the power of my approach.

Finally, Section 6 discusses and concludes.

2 Hierarchical Spaces and Value Function Adaptation

Sparse grids emerge as finding the most efficient grid structure for approximating a given class of functions to represent the solution of an economic model. For ease of exposition, I start by studying the one-dimensional case. The goal is to approximate $f : \Omega \to R$ on some grid $G \subset \Omega$ that discretizes the domain. Denoting by l the level and i the index, a grid knot is constructed as the numeric value $x_{l,i} = i \cdot 2^{-l}$. Grid G is a collection of J such grid points and is represented as a $J \times d$ matrix. The local mesh size is $h_l = 2^{-l}$. At a given level l, the maximum active index is $2^l - 1$. Intuitively, an efficient grid can achieve a desired accuracy in approximating f with a relatively few grid points, and the value of a single grid point is then its marginal contribution to this approximation accuracy. Next, I present a basis function representation to interpolate the approximation of f onto points in the domain Ω that are not members of G. After formally developing the interpolation structure, we can systematically remove grid points whose marginal contribution is low and add them where the local function approximation error remains large.

Let us focus on the linear interpolation with piecewise linear hat functions $\phi's$ as basis functions.

$$\phi_{l,i}(x) = \phi\left(\frac{x - ih_l}{h_l}\right), \qquad \text{where } \phi(x) = \begin{cases} 1 - |x| & \text{for } x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The subscript (l, i) indicates that the hat function is centered around the grid point $x_{l,i}$. Denote a collection of basis functions with a nodal index set of level l as $\{\phi_{l,i}(x)|i \in I_l \text{and} x \in \Omega\}$, where $I_l = \{i \in N|1 \leq i \leq 2^l - 1\}$. We call the span of this set of basis functions the nodal basis. Figure 1 (a) displays the hat functions $\{\phi_{l,i}(x)\}$ for level 3, where the grid point $x_{3,2}$ takes the value $x_{3,2} = i \cdot 2^{-l} = 2 \cdot 2^{-3} = 0.25$. The associated hat function $\phi_{3,2}(x)$ is centered on $X = 0.25$ and has symmetric support over $x \in [x_{3,2} - 2^{-l}, x_{3,2} + 2^{-l}] = [0.125, 0.375]$. Each knot $x_{l,i}$ in grid G is therefore associated with a hat function $\phi_{l,i}(x)$ that is centered on it.

With these basis functions, we can conveniently express the piecewise linear interpolation f_l (blue line in Fig. 1) by the nodal basis on grid G of level l.

$$f(x) \approx f_l(x) = \sum_{i \in l} \alpha_{l,i} \phi_{l,i}(x) \tag{2}$$

The function value of f is approximated by f_l the sum of basis functions with coefficients $\alpha_{l,i} = f(x_{l,i})$. We can formalize the resulting approximation error as the distance f and f_l according to some norm $\|\cdot\|$. Equation (2) is referred to as the nodal representation of the piecewise linear interpolant f_l.

The basis of hat functions implicitly induces an approximation space of piecewise linear functions,

$$\mathcal{V}_l = \text{span}\left\{\phi_{l,i}(x)|i \in I_l \text{ and } x \in \Omega\right\}, \qquad \text{where } l < \infty, \mathcal{V}_l \subset \mathcal{V}. \tag{3}$$

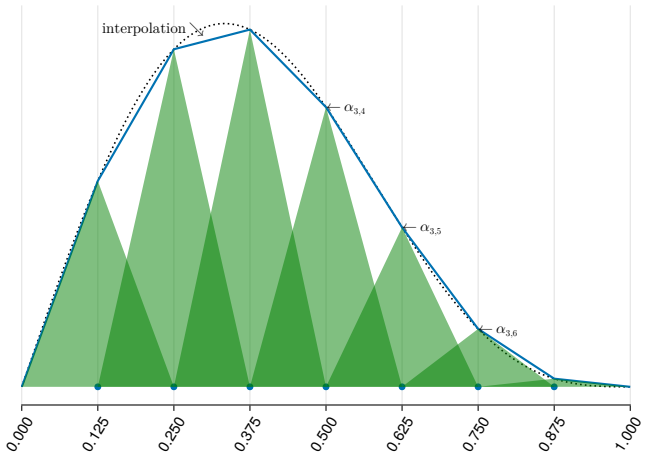

Figure 1: Nodal basis functions

Approximating $f \in \mathcal{V}$ with its piecewise linear interpolant $f_l \in \mathcal{V}_l$ incurs approximation error. For any given $l < \infty$, the approximation space $\mathcal{V}_l$ is finite-dimensional, where the original function space $\mathcal{V}$ is infinite-dimensional. Crucially, associating the vector of function values $\alpha_{l,i} = f(x_{l,i})$ with piecewise basis hat functions $\phi_{l,i}(x)$'s enables us to characterize the distance between f and f_l in the underlying space $\mathcal{V}$, which formalized the approximation accuracy of grid G.

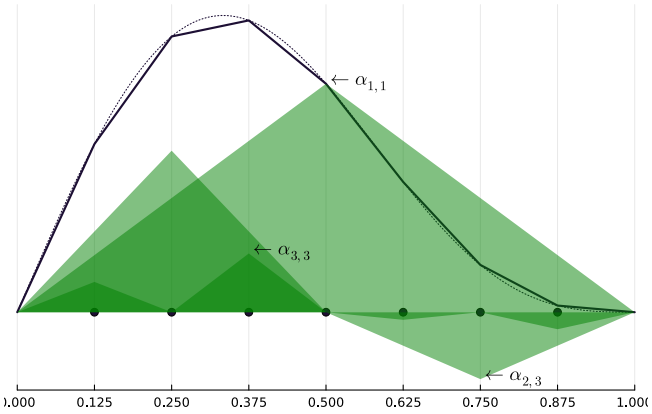

Figure 2: Hierarchical basis functions

Despite of its convenience of describing linear interpolations, the nodal basis is not able to characterize the marginal contribution of a grid point to the overall accuracy of function approximation. In order to develop an efficient procedure to systematically add grid points with large contributions and remove others, we have to consider the hierarchical basis. It restructures the basis hat functions by hierarchical orders and combines them to span the same approximation space of the nodal basis.

Definition 1. ***(Hierarchical Basis / Hierarchical Spaces)*** *The hierarchical basis of level l comprises the set of piecewise linear hat functions*

$$\left\{\phi_{k,i}(x) | k \leq l,\ i \in I_l^H \ \textit{and}\ x \in \Omega\right\} \tag{4}$$

where $I_l^H = \{i \in N | 1 \leq i \leq 2^l - 1$ *and* i *odd}* *is the hierarchical index set. The hierarchical space of level* l *is defined as*

$$\mathcal{W}_l = \text{span}\left\{\phi_{l,i}(x) |\ i \in I_l^H \text{ and } x \in \Omega\right\}. \tag{5}$$

Figure 2 illustrates the construction of the hierarchical basis. At level $l = 1$, the hierarchical space $\mathcal{W}_1$ is the span of the level 1 hat function $\phi_{1,1}(x)$. At level $l = 2$, basis functions active in hierarchical space $\mathcal{W}_2$ are those with odd indices, i.e., $\phi_{2,1}(x)$ and $\phi_{2,3}(x)$. At level $l = 3$, the hierarchical space $\mathcal{W}_3$ is spanned by the basis functions of odd indices, namely $\{\phi_{3,i}(x)\}$ for $i \in I_3^H = \{1, 3, 5, 7\}$. The approximation space of the hierarchical basis of level $l = 3$ is spanned by the hierarchical basis functions of all lower levels by taking the union of the hierarchical spaces $\mathcal{W}_k$ for $k = 1, 2, 3$:

$$\mathcal{V}_l = \oplus_{k \leq l} \mathcal{W}_k. \tag{6}$$

Notice that the nodal and hierarchical bases span the same function space $\mathcal{V}_l$. Any function is the span of the nodal basis is also in the span of the hierarchical basis. Further, both sets of basis functions induce the same approximation space of piecewise linear interpolants.

$$f_l(x) = \underbrace{\sum_{i \in I_l} \alpha_{l,i} \phi_{l,i}(x)}_{\text{nodal representation}} = \underbrace{\sum_{k \leq l} \sum_{i \in I_k^H} \alpha_{k,i}^H \phi_{k,i}(x)}_{\text{hierarchical representation}} \tag{7}$$

In comparison to the nodal spaces, the hierarchical spaces $\mathcal{W}_l$ are disjoint, so the hierarchical basis functions are invariant to the underlying grid structure. For example, if we start with a grid G of level $l = 3$ and remove one of the nodes associated with the highest level 3, the remaining hierarchical basis functions would not be affected. This guarantees stability of the procedure of adding and removing grid points.

Hierarchical surpluses, $\alpha_{k,i}^H$'s, directly measure the contribution of basis functions to the local function approximation. $\alpha_{k,i}^H$ tells us how much a grid point $x_{l,i}$ contributes to the accuracy residually, relative to the interpolant in the absence of this grid point. Adaptive grid refinement relies on the hierarchical surpluses as measure of residual local approximation error. Let us illustrate the grid adaptation procedure by Figure 3.

Panels (a) to (d) of Figure 3 present four successive steps of grid adaptation. In each panel, ϵ^{keep} and ϵ^{add} are the adaptation thresholds plotted as horizontal lines. Panel (a) depicts the hierarchical basis function of level 1, $\phi_{1,1}(x)$, centered on the grid point $x_{1,1} = 0.5$, whose hierarchical surplus $\alpha_{1,1}^H$ is $f(0.5)$. Since it is larger than the threshold ϵ^{add}, we adapt the grid around knot $x_{1,1}$ by adding its children $x_{2,1} = 0.25$ and $x_{2,3} = 0.75$ of level 2 in panel (b). Their associated surpluses $\alpha_{2,1}^H$ and $\alpha_{2,3}^H$ are obtained by taking the difference between the function values $f(x_{2,i})$ (the dashed blue line) and the interpolant $\phi_{1,1}(x_{2,i})$ (the solid blue) in panel (a). In panel (b), we can see that the hierarchical basis functions of level 2 are added and both $\alpha_{2,1}^H$ and $\alpha_{2,3}^H$ are larger than ϵ^{add} in absolute value. So we keep adding their children knots in panel (c) and obtain smaller surpluses as levels get higher. In particular, $\alpha_{3,1}^H$ and $\alpha_{3,3}^H$ are larger than ϵ^{keep}, but smaller than ϵ^{add}. We therefore keep these two knots without further refining. The hierarchical surpluses $\alpha_{3,5}^H$ and $\alpha_{3,7}^H$, on the other hand, are smaller the ϵ^{keep} in absolute

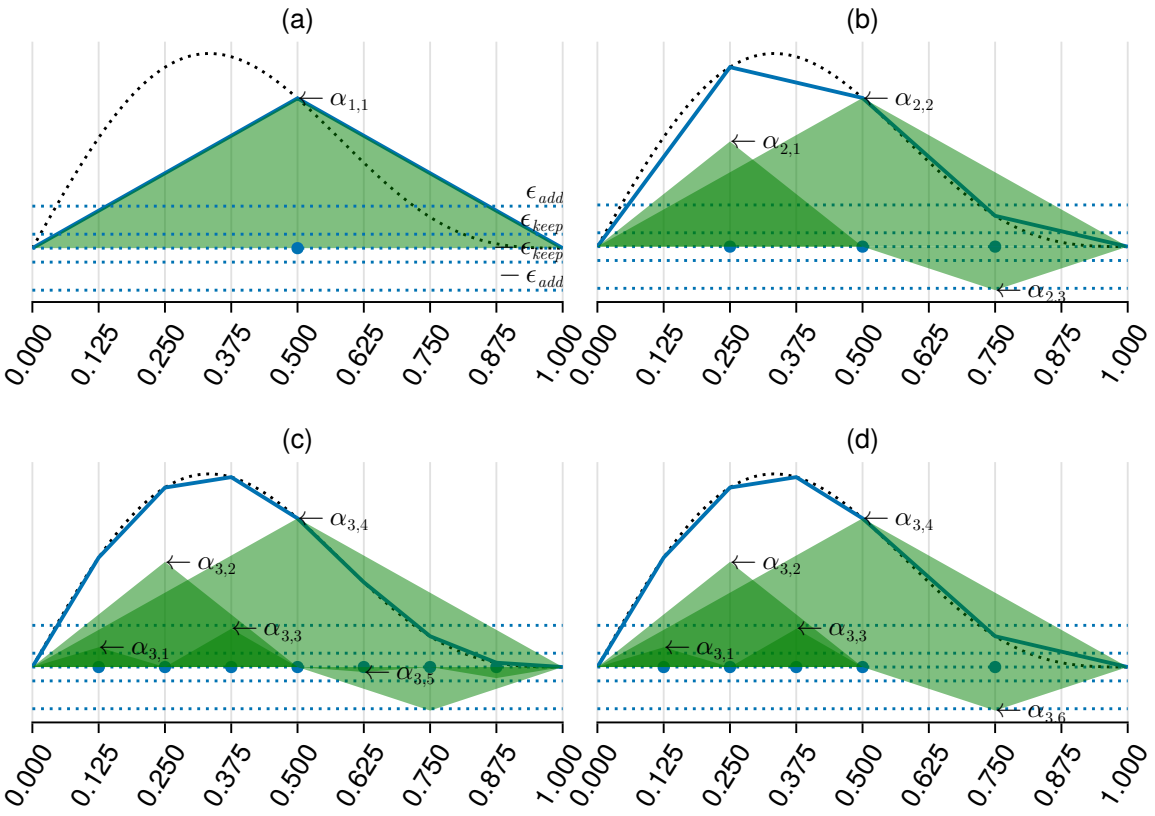

Figure 3: Local grid adaptation

value, suggesting that the associated knots can be dropped. We end up with panel (d) and the procedure converges.

By leveraging the hierarchical basis transformations, the HJB equation can be solved consistently on adaptive sparse grids using FDM. I skip formal derivation of the sparse FDM and refer to Schaab and T.Zhang (2022) for the interested.

3 Distribution Approximation and Cell Structure

The above approximation and grid adaptation procedure is widely used in solving the stochastic dynamic programming problem, but is not suited to approximating distributions. Although interpolation approximation is used as the standard method (Achdou et al., 2021) to compute the distribution on the regular grid, it has some limitations in the context of sparse grid. A core problem of the representation of high-dimensional functions on sparse grids is that they do not maintain the range of function values. The interpolation of a strictly positive function, e.g. probability density functions, can lead to negative function values of the interpolant.

Being incapable of preserving mass and ensuring positive density restricts the interpolation approximation and FDM from solving the KF equation. So, we need to configure a different grid adaptation scheme for computing the distribution. And I show that the FVM scheme and the FDM scheme are nested in a common approximation space. Such that value/policy approximations can be inherited by the FVM cell scheme as inputs for computing the KF

equation.

3.1 Intuition

To get some intuition of connections between cells in the FVM scheme, consider the simplest wealth dynamics of a household

$$dx = s(x)dt \tag{8}$$

where $s(x)$ is the saving function of household. The diagram of the behavior is given in Figure 4. Those cells with boundaries $x_1, ..., x_{n+1}$ are discretization of the domain of wealth x, each of which corresponds to an approximation of a certain distribution bar. Denote the mass of households in a given cell, $[x_i, x_{i+1}]$, by G_i:

$$G_i = \int_{x_i}^{x_{i+1}} g(x)dx \tag{9}$$

where $g(x)$ is the density function. Assume that $s > 0$ for the cell under consideration. Over the time step of Δt, households will save $s\Delta t$ and end with $x_{prev} + s\Delta t$ amount of wealth. Households will then move from the i^{th} cell to the $(i+1)^{th}$ cell (red arrows in the figure) if and only if they were in the range of $[x_{i+1} - s\Delta t, x_{i+1}]$ (shaded green in the figure). Letting g_i denote the density of the i^{th} cell (height of the cell in figure), a total of $g_i \cdot s\Delta t$ of households will move from the i^{th} to the $(i+1)^{th}$ cell. $g_i \approx \frac{G_i}{x_{i+1}-x_i}$ under the assumption of uniform distribution within the cell. The dynamics ends up with

$$G_{i+1,t+\Delta t} = G_{i+1,t} + \frac{G_{i,t}}{x_{i+1} - x_i} s(x_{i+1})\Delta t$$
$$G_{i,t+\Delta t} = G_{i,t} - \frac{G_{i,t}}{x_{i+1} - x_i} s(x_{i+1})\Delta t.$$

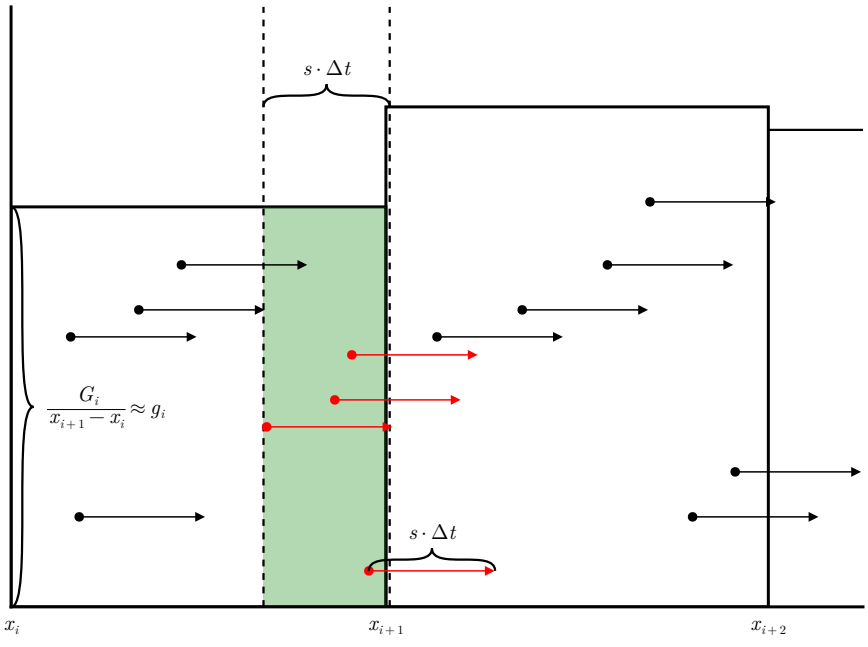

Figure 4: Diagram between histograms

As households flow from cell i to cell $i+1$, the total mass of households is naturally preserved. The positivity is also preserved, since the flow out of cell i is non-zero only if $G_{i,t}$ is positive. This is not guaranteed under the finite difference method, as it does not keep track of the volume (mass) of grid cells. For higher dimension, each cell has more than two flowing directions. FVM records and keeps the neighbor structure of the grid so that one can compute the flows between cells through edges of cells. This is the most difficult part of the implementation. Figure 5 shows two top views of cell neighbor structure with directional edges for 2D cells.

3.2 Cell Structure

From the above simplest example, we can tell a different grid usage of the FVM scheme from the interpolation scheme. In order to preserve the mass and positivity of distribution, FVM's protagonist is the cell constructed by grids rather than simply focusing on grid points. FVM does not use hierarchical surpluses for grid refinement, but relies on the hierarchical structure to inherit policy function approximations from the FDM scheme. Cell adaptation of FVM scheme relies on the histogram volume that locates on the cell and saving flows across cells (more details in Section 4).

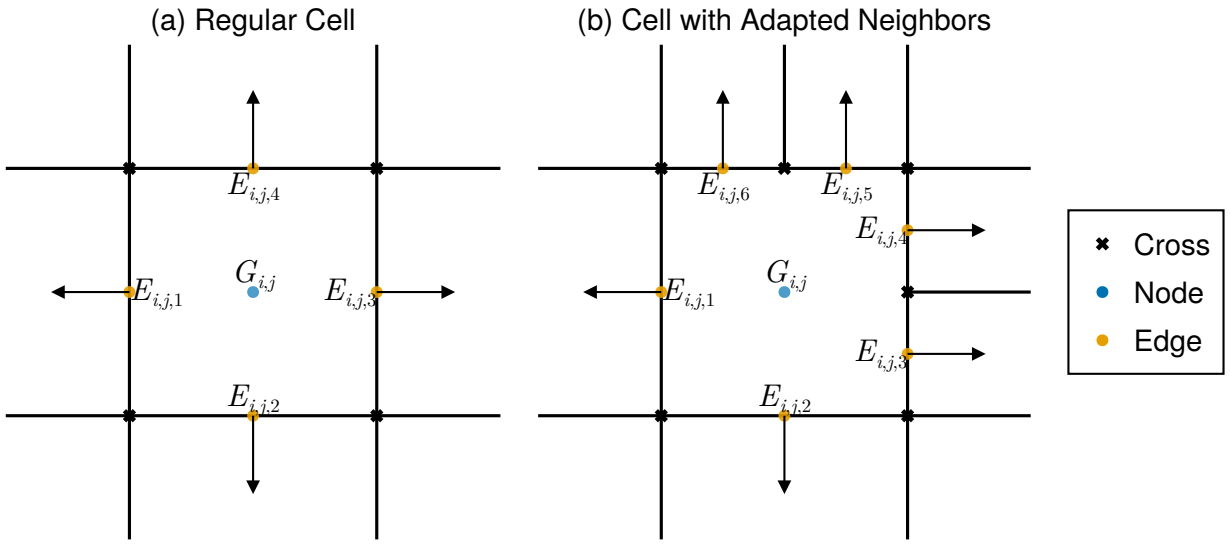

Figure 5: Neighbor structure of 2D cells

Taking one cell from a two-dimensional grid, the flows from the discretized cell are visualized in Fig. 5. There are three types of grid points in a cell. Black cross points construct each cell and are used to maintain cells' structure. Blue node points locate at the center of each cell and approximates the value of the state variable of each cell. Yellow edge points are at the center of cell edges. They approximate the flow rate (e.g. savings) across cells. Only node and edge points are used in the computation of the KF equation. Cross points are bones of cells maintaining cell structure in cell split (refinement) and update hierarchical structure.

Proposition 1. ***(Common Hierarchical Approximation Space for Two Adaptation Schemes)*** *In a MFG system (paired HJB-KF equations), the function interpolation scheme*

and the finite volume scheme can be constructed and adapted within a common hierarchical approximation space.

let $l_{\max} = \max\{l_v, l_d\}$. l_v *and* l_d *are the largest level used in approximations of the value function and the distribtion, respectively. Their adaptation schemes are nested in a common approximation space, defined as*

$$\mathcal{V}_{l_{\max}} = \oplus_{k \le l_{\max}} \mathcal{W}_k, \tag{10}$$

where

$$\mathcal{W}_k = \text{span}\left\{\phi_{k,i}(x) |\ i \in I_k^H \ \text{and}\ x \in \Omega\right\} \tag{11}$$

and $I_k^H = \{i \in N | 1 \le i \le 2^k - 1 \ \text{and}\ i \ \text{odd}\}$ *is the hierarchical index set.*

Consequently, the policy solutions of solving HJB by FDM can be easily inherited to the cell structure scheme used by FVM to solve the KF distribution. The combination of FDM and FVM solves heterogeneous agent problems and speed up the computation of high dimensional economics models. The common hierarchical structure of the grid points used by FDM and FVM are central to my approach. The hierarchical structure allows the projection of policy decisions approximated by FDM to the points used in describing the evolution of the distribution by FVM. The latter then determines the evolution of prices being taken as the input in the HJB equation. This interdependence of the HJB-KF equations promotes better grid refinement for the MFG system: Refining the grid points used by solving the KF equation provides accuracy gain for refining those used by solving the dynamic programming HJB equation, and vice verse.

4 Algorithm Illustration in a Simple Huggett Economy

FVM combined with FDM allows us to compute aggregate dynamics (captured by KF) consistent with individual behaviors (by HJB) on the non-uniform grid. In this section, I will show how to use this approach to solve a simple Huggett economy.

$$\rho v_1(x) = \max_c u(c) + v_1'(x)(z_1 + rx - c) + \lambda_1(v_2(x) - v_1(x)) \tag{12}$$

$$\rho v_2(x) = \max_c u(c) + v_2'(x)(z_2 + rx - c) + \lambda_2(v_1(x) - v_2(x)) \tag{13}$$

$$\frac{dg_1}{dt} = -\frac{d}{dx}[s_1(x)g_1(x)] - \lambda_1 g_1(x) + \lambda_2 g_2(x) \tag{14}$$

$$\frac{dg_2}{dt} = -\frac{d}{dx}[s_2(x)g_2(x)] - \lambda_2 g_2(x) + \lambda_1 g_1(x) \tag{15}$$

$$1 = \int_{\underline{x}}^{\infty} g_1(x)dx + \int_{\underline{x}}^{\infty} g_2(x)dx \tag{16}$$

$$0 = \int_{\underline{x}}^{\infty} x \cdot g_1(x)dx + \int_{\underline{x}}^{\infty} x \cdot g_2(x)dx =: S(r) \tag{17}$$

HJB equations can be solved using FDM by Schaab and T.Zhang (2022)'s method. Check out their very detailed explanation. Here I will focus on the KF equation solved by FVM and then show how to integrate it to FDM.

4.1 Upwind Scheme of Finite Volume Method

FVM discretizes $g_j(x)$ with the mass in each discretization cell. Given $\{x_i\}_{i\in 1,2,...,n+1}$ as the cell boundaries, the distribution is approximated by the vector

$$\overrightarrow{G} = [G_{1,j}\ G_{2,j}\ ...\ G_{n,j}\ G_{1,-j}\ G_{2,-j}\ ...\ G_{n,-j}]'$$

where $G_{i,j}$ is the mass in the cell $[x_i, x_{i+1}]$ given by

$$G_{i,j} = \int_{x_i}^{x_{i+1}} g_j(x)dx \tag{18}$$

Dynamics of the $\overrightarrow{G}$ is then

$$\begin{aligned}
\frac{dG_{i,j}}{dt} &= \int_{x_i}^{x_{i+1}} \frac{dg}{dt}dx \\
&= \int_{x_i}^{x_{i+1}} \left[-\frac{d}{dx}(s_j(x)\cdot g_j(x)) - \lambda_j g_j + \lambda_{-j} g_{-j}\right] dx \\
&= -(s_j(x)\cdot g_j(x))|_{x_i}^{x_{i+1}} - \lambda_j \int_{x_i}^{x_{i+1}} g_j(x)dx + \lambda_{-j}\int_{x_i}^{x_{i+1}} g_{-j}(x)dx \\
&= -s_j(x_{i+1})\cdot g_j(x_{i+1}) + s_j(x_i)\cdot g_j(x_i) - \lambda_j G_{i,j} + \lambda_{-j} G_{i,-j}
\end{aligned} \tag{19}$$

To finish the discretization, $s_j \cdot g_j$ and $\frac{dg_j}{dx}$ need to be approximated in terms of $G_{i,j}$'s, x_i's. The most natural one would be

$$g_j(x) \approx \frac{G_{i,j}}{x_{i+1} - x_i}, \qquad \forall x \in [x_i, x_{i+1}]$$

As the size of the cell decreases, the approximation error will decrease.

Following the upwind scheme of choosing the $g_j(x)$ from the direction of the flow, we put above approximation together and the final discretization equation is

$$\begin{aligned}
\frac{dG_{i,j}}{dt} =& -\frac{s_j(x_{i+1})^-}{x_{i+2} - x_{i+1}} G_{i+1,j} - \frac{s_j(x_{i+1})^+}{x_{i+1} - x_i} G_{i,j} \\
&+ \frac{s_j(x_i)^-}{x_{i+1} - x_i} G_{i,j} - \frac{s_j(x_i)^+}{x_i - x_{i-1}} G_{i-1,j} \\
&- \lambda_j G_{i,j} + \lambda_{-j} G_{i,-j}
\end{aligned} \tag{20}$$

For any number x, the notation x^+ means the positive part of x, i.e. $x^+ = \max\{x, 0\}$ and analogously $x^- = \min\{x, 0\}$. The expression is linear in $\overrightarrow{G}$ and can be written in matrix notation as

$$\frac{d\overrightarrow{G}}{dt} = \boldsymbol{A}\overrightarrow{G} \tag{21}$$

where $\boldsymbol{A}$ is the matrix representation of eq. (20)

$$\boldsymbol{A} = \begin{bmatrix} y_{1,1} & z_{1,1} & 0 & \dots & 0 & \lambda_2 & 0 & 0 & \dots & 0 \\ x_{2,1} & y_{2,1} & z_{2,1} & 0 & \dots & 0 & \lambda_2 & 0 & 0 & \dots \\ 0 & x_{3,1} & y_{3,1} & z_{3,1} & 0 & \dots & 0 & \lambda_2 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & x_{I,1} & y_{I,1} & 0 & 0 & 0 & 0 & \lambda_2 \\ \lambda_1 & 0 & 0 & 0 & 0 & y_{1,2} & z_{1,2} & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 & 0 & x_{2,2} & y_{2,2} & z_{2,2} & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 & 0 & x_{3,2} & y_{3,2} & z_{3,2} & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & \dots & 0 & \lambda_2 & 0 & \dots & 0 & x_{I,2} & y_{I,2} \end{bmatrix} \tag{22}$$

$$x_{i,j} = \frac{s_j(x_i)^+}{x_i - x_{i-1}}, \quad y_{i,j} = -\frac{s_j(x_{i+1})^+}{x_{i+1} - x_i} + \frac{s_j(x_i)^-}{x_{i+1} - x_i} - \lambda_j, \quad z_{i,j} = -\frac{s_j(x_{i+1})^-}{x_{i+2} - x_{i+1}}$$

The linearity of Equation (21) allows one to leverage the numerical linear algebra. The steady state distribution is obtained by solving $\boldsymbol{A}\overrightarrow{G} = 0$, which has a very fast solution method. If a program constructs the matrix $\boldsymbol{A}$, one can compute the dynamics of the distribution through simple matrix operations. Working with matrix operations vastly simplifies the implementation compared to problem-specific interpolations especially for lifecycle/OLG models (see Section 5.2).

4.2 Connections between Two Schemes of Grid Points

Since FVM has an extra task—preserving mass and positivity, it maintain a cell structure during grid refinement. Given Proposition 1, we can simply project policy functions solved by FDM to the FVM cell grid points and solve the KF equation accordingly. The projection is very efficient and is a key step to ensure the mapping between individual behaviors and aggregation. This mapping is guaranteed by the consistent hierarchical approximation structure of two schemes. Although there is a possibility that we can extend the grid points used by FDM to preserve the density value range (Pflüger and Franzelin, 2018), the size of the extension set, however, differs among applications and could be almost as large as, no longer sparse at all, a full grid. In contrast, my approach, although employs two adaptation schemes, has little efficiency lost, while retains the advantage of both grid structure.

Figure 6 presents the saving functions and their projections to different grid knots. The first sub-figure is the scatter solution of saving functions from the HJB equation. The second and third one are projections of the solution of HJB to cell midpoints and edge midpoints used in computing the KF equation by the FVM scheme. The successful projection can be visualized by the superposition. It shows the successful mapping of policies between grid schemes within a common hierarchical approximation space. For the simple Huggett model I consider, cell midpoints and edge midpoints are degenerated on the same x axis. For higher dimension models, e.g. the 2D case in Figure 5, edges are used to locate the flow direction between cells. Edge midpoints are knots to approximate the function values within certain edge, capturing

the flow rate at the edge. The rate of increase of the mass of a given cell is determined by the direction and flow rate at the edge boundary multiplied by the size of the edge.

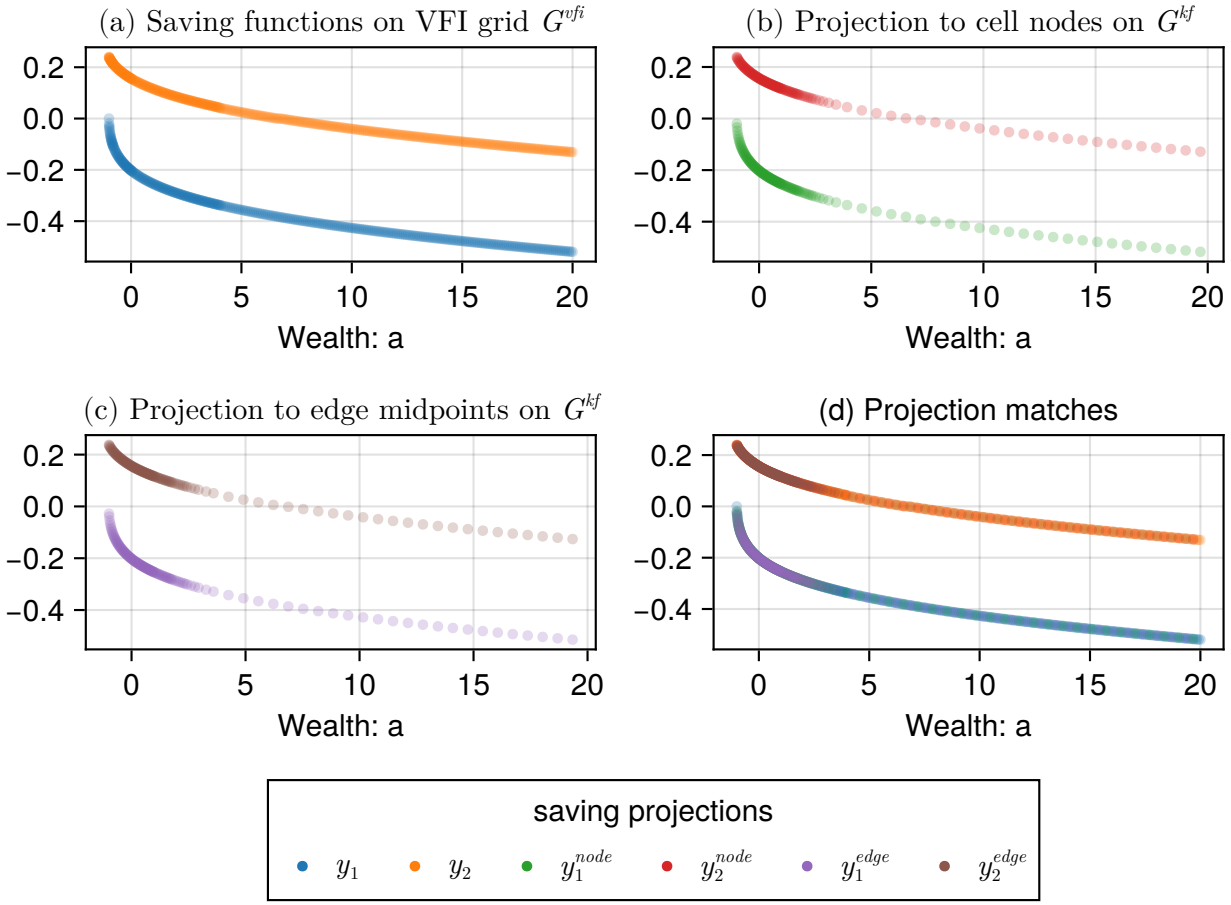

Figure 6: Saving policy functions of the Huggett model

4.3 Adaptive Refinement Criteria and Algorithm of General Equilibrium Iteration

Leveraging the combination of FDM and FVM, I present a KF augmented general equilibrium iteration algorithm on the adaptive sparse grid space. The algorithm is facilitated by an outer fixed point, in which the algorithm automatically adapts the FDM grid to increase (decrease) resolution where the numerical approximation error in the value function remains large (small), while refine the FVM grid to split cells adaptively based on an error metric of the uniformity assumption of distribution within the cell. My proposed algorithm to solve general equilibrium heterogeneous agent problems is summarized in Algorithm 1.

For the FDM procedure, reasonable knots are added according to the hierarchical surpluses of the value function (see Section 2). Figure 6 presents that high resolution is added to the non-linear region of saving policy curves, where households are near their borrowing limits and are likely to be constrained. The strategy of FVM refinement is to leverage the cell structure and to split cells where distribution mass *and* saving flow (drift) are high. The refinement

metric is given by

$$metric \approx G_{i,j} \cdot \text{drift}\left(\frac{x_i + x_{i+1}}{2}\right) \tag{23}$$

This metric is intuitive. If the drift is small, the uniformity assumption is accurate. otherwise, If the mass of the cell is small, the expected gain from splitting the cell is small. Thus, the cell should be split only when both the fraction of the cell and the drift are large.

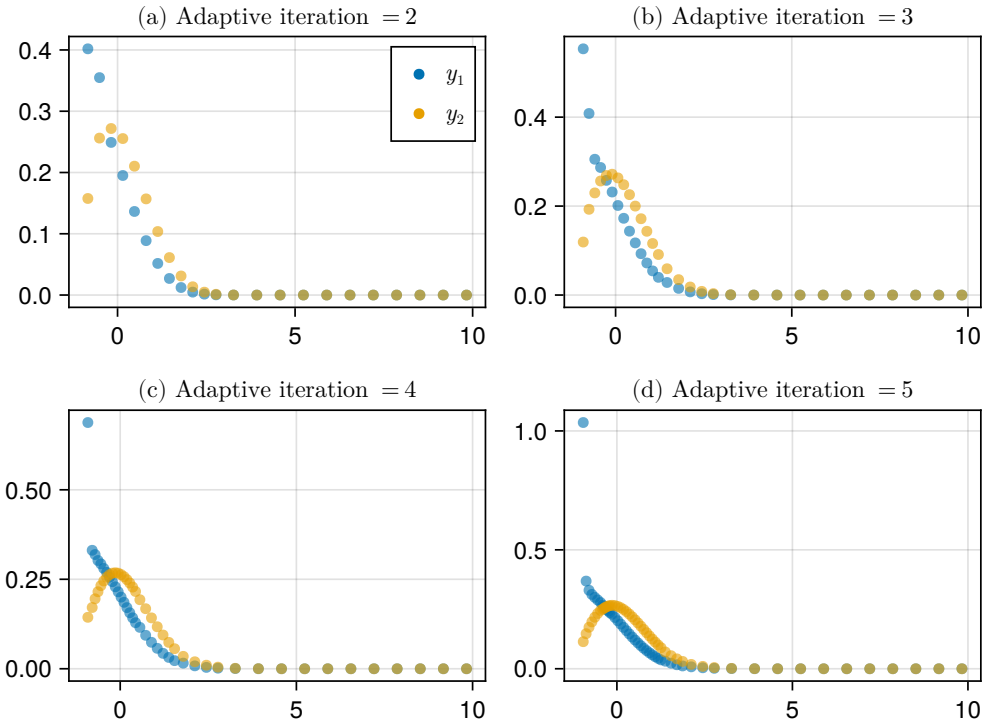

Figure 7: Wealth Distributions of the Huggett model

Figure 7 shows the solution of distributions in the Huggett model and its adaptation procedure. Grid cells are split in the steep region of the distribution. These refined cells lie in roughly the same region where saving functions are adapted frequently, implying that a large amount of distribution mass consists of constrained households. In Section 5.1, we will see a different situation where the majority of households do not locate at the constrained region. My approach is flexible enough to handle it.

Algorithm 1: VFI and KF Iteration on FDM-FVM Sparse Grids

Choose level indices $\boldsymbol{l}^{\mathrm{VFI}}$, $\boldsymbol{l}^{\mathrm{KF}}$ for the HJB and the KF equations respectively and construct associated regular sparse grid G_0^{VFI}, G_0^{KF}

Construct hierarchization operator $\boldsymbol{H}^0$ and projection matrix $\boldsymbol{B}\boldsymbol{H}^0$

Construct sparse finite difference FD operators on G_0^{VFI} and sparse finite volume FV operator on G_0^{KF}

Initialize guess for interest rate r

for $i \geq 0$ **do**

    ; `/*the most outer loop is for adaptation*/`

    Initialize guess for the value function $\boldsymbol{V}^0$ on G_0^{VFI}

    for $k \geq 0$ **do**

        ; `/*this loop is for general equilibrium*/`

        for $n \geq 0$ **do**

            Compute the transition matrix of the HJB equation $\boldsymbol{A}^{\mathrm{D}(k,n)}$ using sparse FD operators

            Solve linear system for $\boldsymbol{V}^{k,n+1}$

            Continue if $|\boldsymbol{V}^{k,n+1} - \boldsymbol{V}^{k,n}| > \epsilon^{\mathrm{VFI}}$

        end

        Project the policy function on G_i^{VFI} to G_i^{KF}

        Compute the transition matrix of the KF equation $\boldsymbol{A}^{\mathrm{V}(k)}$ using sparse FV operators

        Solve linear system for distribution approximation $\boldsymbol{g}^{k+1}$

        Compute the aggregate asset using projections of policies on G_i^{KF} and $\boldsymbol{g}^{k+1}$

        Update interest rate r and continue if the asset market does not clear

    end

    Compute hierarchical surplus $\boldsymbol{\alpha}^{i,H} = \boldsymbol{H}^i \boldsymbol{V}^i$

    Compute adaptation metrics $\boldsymbol{metric} \approx \boldsymbol{g}^i \cdot \mathrm{drift}$

    If FDM adaptation criterion met, adapt grid and update to G_{i+1}^{VFI}

    If FVM adaptation criterion met, split cells and update to G_{i+1}^{KF}

    Reconstruct $\boldsymbol{H}^{i+1}$ and projection matrix $\boldsymbol{B}\boldsymbol{H}^{i+1}$

    Reconstruct sparse FD operators on G_{i+1}^{VFI} and the FV operator on G_{i+1}^{KF}

end

4.4 Combined Sparse Grid

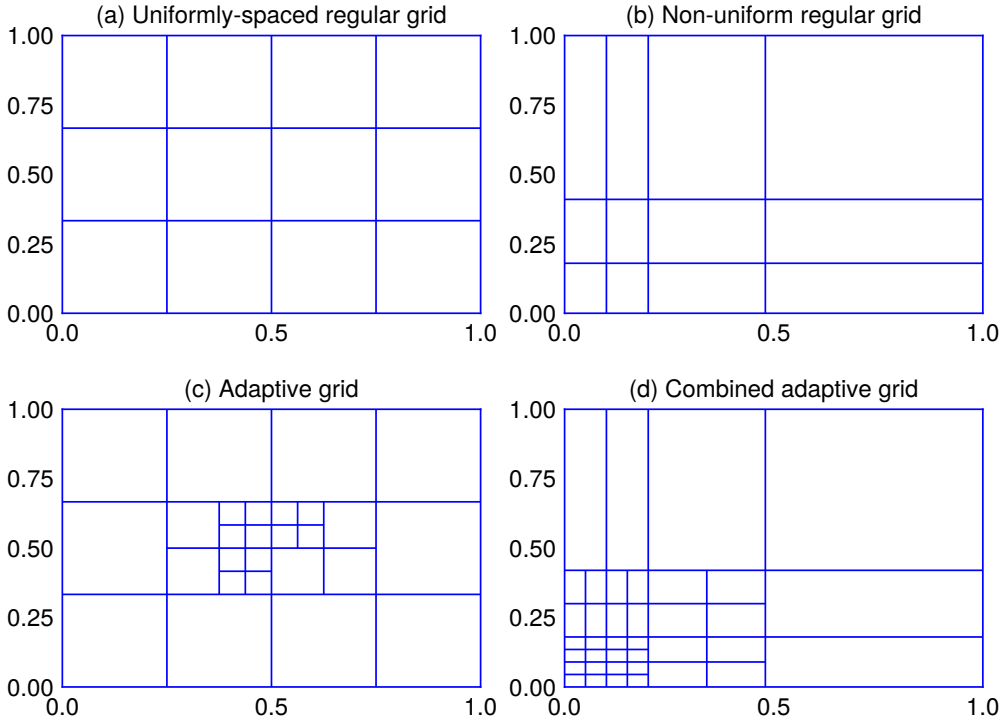

Figure 8: Different types of grids

Figure 8 presents four typical types of 2D grids used by both FDM and FVM. We have so far discussed how to transform uniformly-spaced regular grid (Fig. 8(a)) to the adaptive sparse grid (Fig. 8(c)). The non-uniform regular grid (Fig. 8(b)) is widely used in the quantitative heterogeneous agent literature. When the range of asset holdings is so large, say, $0 - 2000$, we often use log-spaced or log-square-spaced grids. This works well in practice, since empirically a large amount of households hold wealth close to the zero point, while a few wealthy households lie in the tail region of the wealth distribution.

My approach works on non-uniform regular grid, too. I call it the combined adaptive grid (Fig. 8(d)). When facing large range of asset values, one can simply start with log-spaced or log-square-spaced grids, and continue refining grids with my method. This works best if one has prior knowledge of how the asset distributes across the range, such that we start with an already sparse grid and refine grids faster.

5 Use Cases and Applications

In this section, I demonstrate the power of adaptive sparse grids methods with two heterogeneous-agent applications in macroeconomics: a Krusell-Smith economy and an overlapping generation model.

5.1 Krusell Smith Model

I start with a variant of Krusell et al. (1998) model. There is a continuum of households with preferences over consumption, given by

$$E_0 \int_0^\infty e^{-\rho t} u(c_t) dt, \tag{24}$$

where ρ is a common discount rate and c_t the rate of consumption. Households face both uninsurable idiosyncratic risk in the form of unemployment spells and aggregate risk. Insurance markets are incomplete but households can trade capital, which they rent to firms. A household's budget constraint is given by

$$\dot{k} = (r_t^k - \delta)k_t + (1 - \tau)w_t z_t - c_t \tag{25}$$

subject to the short-sale constraint $k_t \leq 0$, where $r_t^k - \delta$ is the rental rate of capital net of depreciation, τ is labor income tax. I assume that z_t follows a two-sate Poisson process.

A representative and perfectly competitive firm uses capital and labor to produce the final consumption good according to the aggregate production function

$$Y_t = e^{Z_t} K_t^\alpha N_t^{1-\alpha}, \tag{26}$$

where Z_t is the aggregate TFP, and K_t and N_t denote the aggregate caption stock and labor. α denotes the income state of capital. Under perfect competition, factor prices are simply given by $r_t^k = \alpha \frac{Y_t}{K_t}$ and $w_t = (1 - \alpha)\frac{Y_t}{N_t}$.

On the household side, I calibrate $\rho = 0.05$ and assume isoelastic preferences with $u(c) = \frac{1}{1-\gamma} c^{1-\gamma}$ and coefficient of relative risk aversion $\gamma = 2$. Households face uninsurable earnings risk encoded in the two-state Markov process $z_t \in \{z^E, z^U\}$. I model z_t using a Poisson process with arrival rates $\lambda(z)$. For simplicity, I set $z^E = 1.2$ and $z^U = 0.8$, as well as $\lambda(z^E) = \lambda(z^U) = 1/3$. For aggregate TFP risk, I set $\theta = 0.05$ and $\sigma = 0.007$. On the production side, I calibrate a capital share of $\alpha = 0.33$ and set the rate capital depreciation to $\delta = 0.05$.

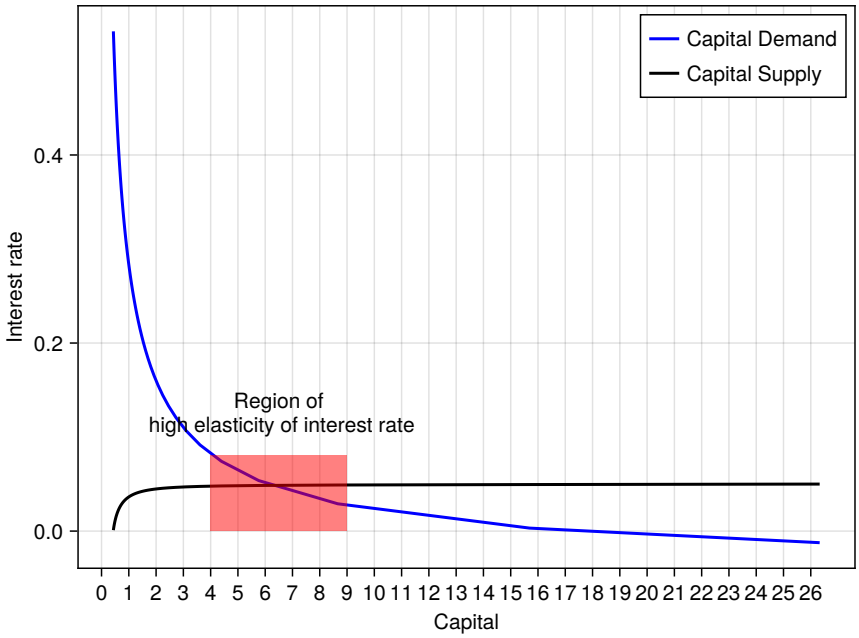

Figure 9: Demand and Supply Curves

Let us first look at the steady state with a stationary distribution. Figure 9 presents the capital market clearing using demand and supply curves. The capital demand curve is derived from the representative firm side, while the capital supply is the aggregates of heterogeneous households' cumulative savings. We can see that the supply curve is quite steep around where equilibrium lies around, i.e. interest rate elasticity near the region of intercept (clearing) is high. The aggregate asset supply rises sharply where the interest rate is close to the equilibrium. This can cause trouble in that we need good approximations around where it matters. However, merely refining the value function does not give useful insights of this region (the red region of Fig. 9), since the aggregation of capital supply relies more on the capital distribution. The adaptation process of the value function cannot target directly on it.

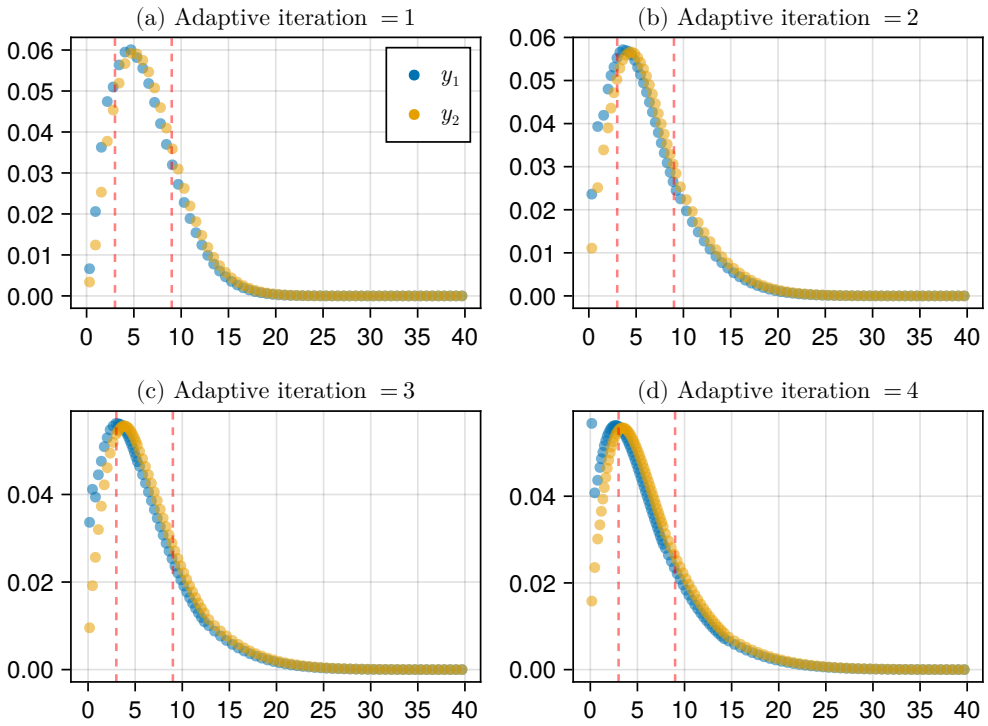

Figure 10: Adaptation of Krusell-Smith Steady State

The distribution adaptation, however, does capture this key region by computing the metric $s_j(k) \cdot g_j(k)$. This metric is intuitive. When the saving drift and the mass of the cell are large, the agents who lie in this region are important for capital aggregation and therefore to this economy. This insight cannot be obtained by solely adapting the value function. Figure 10 presents the general equilibrium adaptation. From Fig. 10(a)-(d), it is clear that the resolution of distributions is increasing during grid refinement in the region of high elasticity of interest rate (the interval between the two red dash lines). This guarantees the accuracy we need for approximating the steady state.

Figure 11 presents the numerical solution of the Krusell-Smith model. I initialize the economy at the stationary equilibrium without uncertainty at time $t = 0$ and simulate the economy

under a sequence of aggregate TFP shocks. The blue line corresponds to the time series of aggregate capital, denoted K_t^{sim}. The yellow line, on the other hand, plots an unconditional and out-of-sample time-0 forecast by households, denoted K_t^{lom}. I compute the Den Haan (2010) metric of the solution, which is defined as $\epsilon_K^{DH} = 100 \times \max_t \log\{K_t^{lom} - K_t^{sim}\}$. We can interpret ϵ_K^{DH} as the largest percentage error in an unconditional forecast of capital given the realized draw of shocks $\{Z_t\}$. If ϵ_K^{DH} is small, households' beliefs are consistent with the true law of motion of the economy, which is a required condition for a rational expectations equilibrium. With almost the same size of grid points, my approach achieves a lower ϵ_K^{DH} compared to the standard sparse grid method.

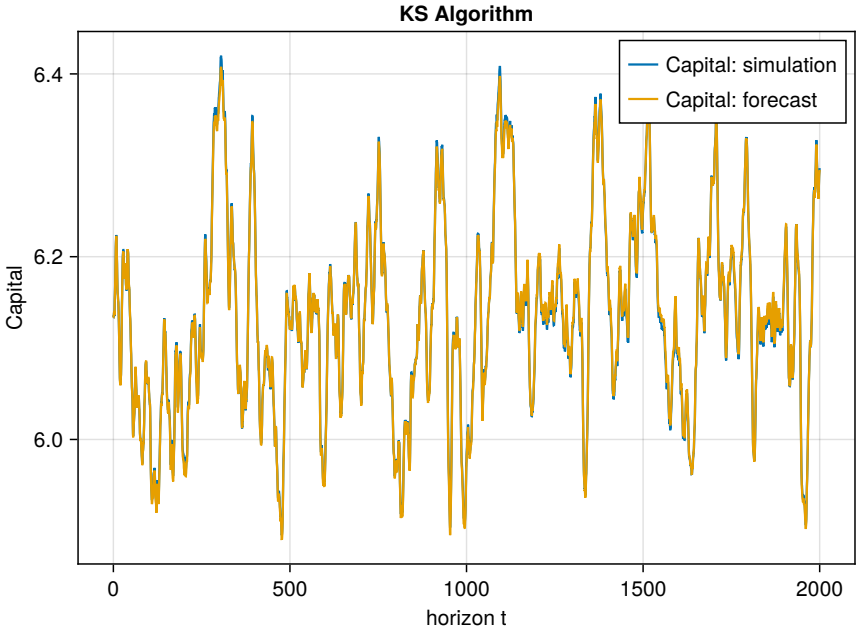

Figure 11: Results of Krusell Smith Simulations

5.2 Life-Cycle and Overlapping Generations

This section explains how to solve a finite-horizon life-cycle model in general equilibrium (that forms overlapping generations). Such models (e.g. Huggett (1996)) are typically solved by explicitly iterating over time or age dimension, which involves solving partial differential equations for every discretized time point. This often becomes the most computationally intensive step of the algorithm.

With the continuous time setup, we can treat age as an exogenous state variable in the grid just as the income risk. Unlike the standard approach, we do not need to explicitly iterate over the N points used to discretize the age dimension. This forces a uniform grid of complexity $J \times N$. The benefit is instead discretizing all n dimensions jointly on a sparse grid of including age as another dimension. Adaptive grid refinement then efficiently places grid points in the age dimension where policy functions feature most non-linearity and distribution functions feature largest densities.

Although it does not take advantage of the special structure of deterministic aging (i.e., it is easy to do backward and forward iterations from the end and beginning of life, respectively),

modeling age in partial differential equations allows us to use implicit and semi-implicit schemes to speed up the iteration. In practice, the number of iterations required for VFI convergence under a semi-implicit scheme is small. For the model I show below, VFI convergence requires 7 iterations in $n-1$ dimensions excluding age and only 10 when adding age as an nth dimension. The standard approach instead requires explicitly iterating from the boundary N times, where N is the number of points used to discretize the age dimension.

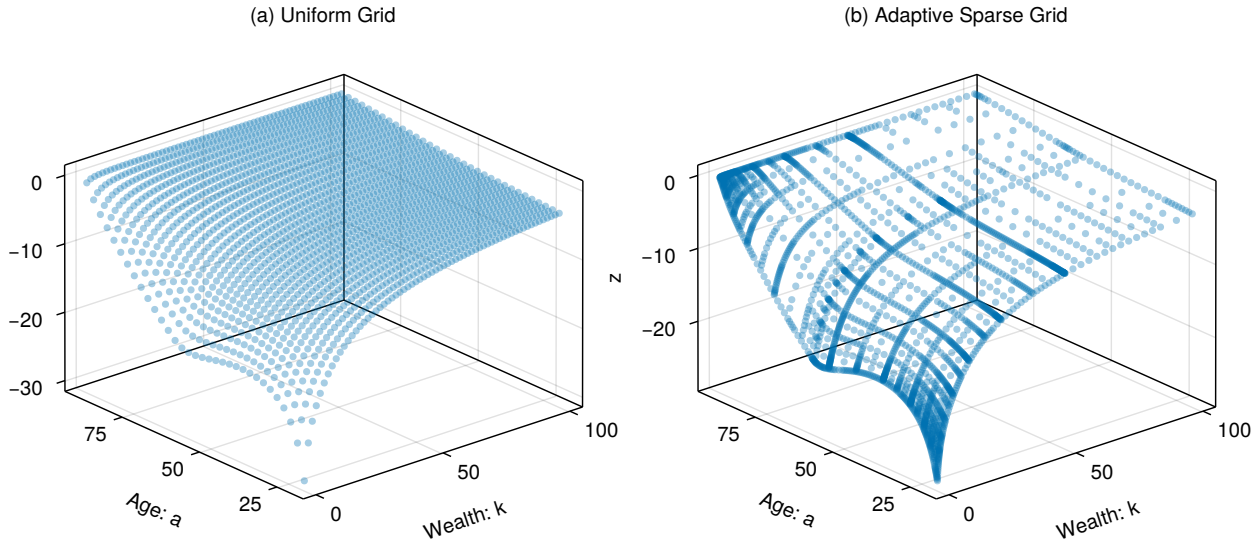

Figure 12: Adaptive Sparse Grid Representation of the Value Function

I illustrate the power of adaptive sparse grids for OLG problems in the context of a simple one-asset life-cycle model. It is similar to the Aiyagari (1994) variant presented in Section 5.1, except a life cycle. I denote the age of a household by $a_t \in [\underline{a}, \bar{a}]$, explicitly distinguished from t. While t indexes the flow of time in the economy, age a_t denotes the age of a particular household at calendar time t. I interpret $\underline{a}$ as the start of working life and $\bar{a}$ as the time of death. Households entering the economy at time 0 with $a_0 = \underline{a}$ face finite horizon problem

$$V_0 = E_0 \int_0^{\bar{a}-\underline{a}} e^{-(\rho+\tau(a))t} u(c_t) dt \tag{27}$$

subject to the capital accumulation eq. (25) and the same labor productivity defined in Section 5.1. $\tau(a)$ is an age-dependent death rate. The law of motion for the new state variable age is trivially given by

$$\dot{a}_t = 1. \tag{28}$$

The household consumption and saving policy functions now depend on both calendar time t and age a separately

$$c_t = c_t(a, k, z)$$
$$s_t = s_t(a, k, z).$$

The household's lifetime value now solves the HJB equation

$$\begin{aligned}\rho V_1(a,k) =& \max_c u(c_1(a,k)) + \partial_k V_1(a,k)(wz_1 + rk - c)\partial_a V_1(a,k) \\ &+ \lambda_1(V_2(a,k) - V_1(a,k)) \\ \rho V_2(a,k) =& \max_c u(c_2(a,k)) + \partial_k V_2(a,k)(wz_2 + rk - c)\partial_a V_2(a,k) \\ &+ \lambda_2(V_1(a,k) - V_2(a,k))\end{aligned}$$

implying a first-order condition for consumption given by $u'(c_j(a,k)) = \partial_k V_j(a,k)$. The induced Fokker-Plank equation is

$$\frac{\partial}{\partial t} g_1(a,k) = -\frac{\partial}{\partial k}[s_1(a,k)g_1(a,k)] - \frac{\partial}{\partial a} g_1(a,k) - \lambda_1 g_1(a,k) + \lambda_2 g_2(a,k) \tag{29}$$

$$\frac{\partial}{\partial t} g_2(a,k) = -\frac{\partial}{\partial k}[s_2(a,k)g_2(a,k)] - \frac{\partial}{\partial a} g_2(a,k) - \lambda_2 g_2(a,k) + \lambda_1 g_1(a,k) \tag{30}$$

I set $\underline{a} = 25$ as the start of working life and $\bar{a} = 80$ as the time of death. The age-dependent death probabilities $\tau(a)$ are set according to those used in Huggett (1996). The two-state earnings risk Markov chain is the same as in Section 5.1.

In Figure 12, I plot the numerical solution of the stationary household value function for the low earnings type, $V_1(a,k)$. Panel (a) plots the solution on a uniform grid, while panel (b) exhibits that on an adaptive sparse grid using roughly the same number of grid points. It is obvious from the comparison how the grid refinement procedure removes grid points from regions where the value function is approximately linear, while adding them to regions featuring substantial non-linearity. These regions include the most young age period, the retirement and the near-death region. It is evident from Fig. 12(a) that a uniform age grid leads to substantially approximation error in those important regions.

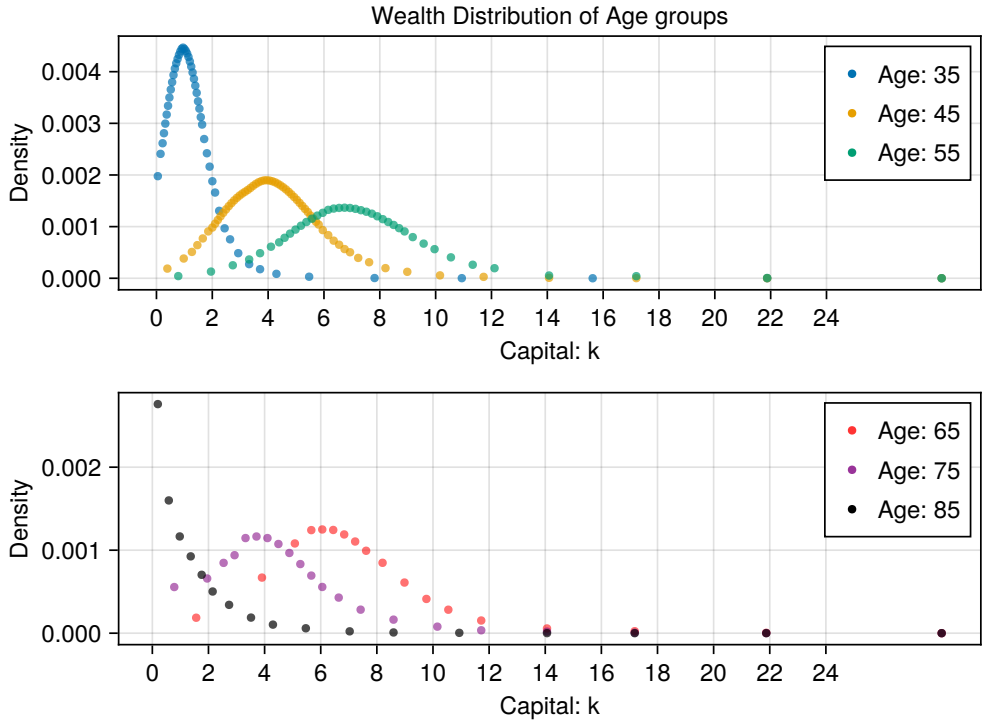

Figure 13: Adaptive Sparse Grid Representation of Wealth Distributions

Figure 13 shows how the adaptation procedure works for the wealth distribution. As people in different parts of their life cycle have different wealth levels, the wealth refinements need to differ based on their age, and the FVM method tracks the distribution well over the lifetime. For the lifecycle problem, the refinement metric that performed best is

$$\text{metric} = (r \cdot k_i + w \cdot z_j - c_j^*(a, k_i)) \cdot G_{ij} \cdot (150 - a) \cdot (k_{i+1} - k_i)$$

The notable difference compared to the previous section is the $(150 - a)$ term. We weigh more on approximation of the young for two reasons. One is that the wealth distribution of young is more concentrated and the density is steeper. Another is that the approximations on young affect the values of older households. It is clear from the figure that throughout the lifecycle people save for retirement when they are young, and gradually dissave upon retirement during aging.

6 Conclusions

In this paper, I propose a new approach to solving heterogeneous agent models in continuous time leveraging adaptive sparse grids. I develop a FDM-FVM method and a general equilibrium iteration algorithm that are robust across a broad class of grids. My algorithm automatically adapts grids and adds local resolutions in regions of the state space where both the value function and the distribution approximation errors remains large. I demonstrate the power of my approach in applications feature high-dimensional state spaces, occasionally binding constraints, lifecycle and overlapping generations.

References

Achdou, Y., J. Han, J.-M. Lasry, P.-L. Lions, and B. Moll (2021, 04). Income and wealth distribution in macroeconomics: A continuous-time approach. *The Review of Economic Studies 89*(1), 45–86.

Ahn, S. (2017). Sparse grid methods for continuous-time heterogeneous agent models. Working Paper 10/2017. hdl:11250/2604936.

Ahn, S. (2019). Computing the distribution: Adaptive finite volume methods for economic models with heterogeneous agents. Working Paper 10/2019. hdl:11250/2604936.

Aiyagari, S. R. (1994). Uninsured idiosyncratic risk and aggregate saving. *The Quarterly Journal of Economics 109*(3), 659–684.

Bilal, A. (2023). Solving heterogeneous agent models with the master equation. Working paper, National Bureau of Economic Research.

Cardaliaguet, P., F. Delarue, J.-M. Lasry, and P.-L. Lions (2019). *The Master Equation and the Convergence Problem in Mean Field Games: (AMS-201)*, Volume 2. Princeton University Press.

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. Computational Suite of Models with Heterogeneous Agents: Incomplete Markets and Aggregate Uncertainty.

Huggett, M. (1996, December). Wealth distribution in life-cycle economies. *Journal of Monetary Economics 38*(3), 469–494.

Krusell, P., A. A. Smith, and Jr. (1998, October). Income and Wealth Heterogeneity in the Macroeconomy. *Journal of Political Economy 106*(5), 867–896.

Pflüger, D. and F. Franzelin (2018). Limiting ranges of function values of sparse grid surrogates. Working paper.

Ruttscheidt, S. (2018). Adaptive sparse grids for solving continuous time heterogeneous agent models. Working paper.

Schaab, A. and A. T.Zhang (2022). Dynamic programming in continuous time with adaptive sparse grids. Working paper.

Young, E. R. (2010, Jan). Solving the incomplete markets model with aggregate uncertainty using the krusell–smith algorithm and non-stochastic simulations. *Journal of Economic Dynamics and Control 34*(1), 36–41.